ROMA TRE
UNIVERSITÀ DEGLI STUDI

SCUOLA DOTTORALE EDEMOM

Sezione Elettronica: dalle Nanostrutture ai Sistemi

XXIII ciclo

Rivelatori in diamante per *imaging* di sorgenti UV e X

Marco Girolami

Docente-guida: ***Prof. Gennaro Conte***

Coordinatore: ***Prof. Giuseppe Schirripa Spagnolo***

A.A 2010/2011

A Valerio

Indice

Introduzione **11**

Capitolo I - Radiazioni UV e X: applicazioni, sorgenti e rivelatori

1.1 I raggi UV e X nello spettro elettromagnetico 13

1.2 Principali applicazioni delle radiazioni UV 14

1.3 Principali applicazioni dei raggi X 16

1.4 Sorgenti UV 17
- 1.4.1 Lampade a Xenon-Mercurio 17
- 1.4.2 Laser a eccimeri 18
- 1.4.3 Lampade a eccimeri 19

1.5 Sorgenti X 20
- 1.5.1 Tubo di Coolidge 20
- 1.5.2 Sorgenti radioattive 22
- 1.5.3 Luce di sincrotrone 23
- 1.5.4 Sorgenti plasma-laser 25

1.6 Rivelatori di radiazioni UV 25
- 1.6.1 Rivelatori foto emissivi 26
- 1.6.2 Rivelatori al silicio "UV-*enhanced*" 26
- 1.6.3 Il diamante come alternativa al silicio "UV-*enhanced*" 28

1.7 Rivelatori di raggi X 29
- 1.7.1 Rivelatori a gas 29
- 1.7.2 Rivelatori a scintillazione 30
- 1.7.3 Rivelatori a semiconduttore 31

Capitolo II - Il diamante: proprietà, sintesi e fotoconducibilità

2.1 Struttura del carbonio 35

2.2 Proprietà fisiche ed elettroniche del diamante 36

2.3 Tecniche di crescita del diamante 37

2.4 Morfologia e struttura del diamante policristallino 39
- 2.4.1 Dimensione dei grani cristallini 40
- 2.4.2 Rapporto tra fase diamante e fase grafitica 42
- 2.4.3 Direzione cristallografica del film 43

2.5 Drogaggio del diamante CVD 44

2.6 Generazione e ricombinazione di carica nei fotorivelatori....45
2.6.1 Modello di Shockley-Hall-Read (SHR)....47
2.6.2 Presenza di un livello trappola....48

2.7 Assorbimento di radiazioni ottiche e fotoconducibilità....49

2.8 Assorbimento di raggi X....51

Capitolo III - Realizzazione e caratterizzazione di rivelatori a singolo pixel

3.1 Introduzione....53

3.2 Struttura del rivelatore a singolo pixel....53

3.3 Fabbricazione dei rivelatori....55
3.3.1 Pulizia superficiale dei substrati....55
3.3.2 Metallizzazione....56
3.3.3 Fotolitografia....57
3.3.4 Attacco chimico....59

3.4 Studio dell'ohmicità dei contatti....60
3.4.1 Elettrometro Keithley 6517....61
3.4.2 Risultati sperimentali....63

3.5 Studio della fotoconducibilità spettrale....64
3.5.1 Apparato di misura....64
3.5.1.1 Amplificatore *lock-in*....67
3.5.1.2 Controllo e acquisizione dati....68
3.5.2 Risultati sperimentali....69

3.6 Misure di fotoconducibilità UV impulsata....70
3.6.1 Laser a eccimeri Neweks PSX100....71
3.6.2 Ottica e *beam splitter*....72
3.6.3 *Bias tee* Picosecond 5575A....73
3.6.4 Oscilloscopio digitale LeCroy Wavepro 960....74
3.6.5 Fotodiodo Schottky per il *trigger* esterno....75
3.6.6 Risultati sperimentali....75

3.7 Misure di fotocorrente X....77
3.7.1 Sorgente di raggi X e sottosistema di filtraggio....77
3.7.2 Risultati sperimentali....79

3.8 Misure di spettroscopia a raggi X....83
3.8.1 Sorgente Amptek *Cool-X*....84
3.8.2 ORTEC 556 *High Voltage Power Supply*....86
3.8.3 Preamplificatore di carica ORTEC 125....86
3.8.4 AMPTEK PX4 *Digital Pulse Processor* (analizzatore multicanale)....88
3.8.5 Impulsatore GW INSTEK GFG-3015....91
3.8.6 Rete di attenuazione....91
3.8.7 Risultati sperimentali....92
3.8.7.1 Calibrazione dell'analizzatore multicanale....92
3.8.7.2 *Pulse Height Analysis* (PHA)....94

Capitolo IV - Progettazione e realizzazione di un sistema elettronico multicanale per l'acquisizione e il condizionamento di segnali di corrente

4.1 Introduzione 97
4.1.1 Sistemi multicanale per misure di corrente 98
4.1.2 Scheda di valutazione ECSIC2 99
4.1.3 Studio di fattibilità del sistema multicanale: ECSIC8 100

4.2 Schema a blocchi del sistema ECSIC32 102

4.3 Convertitore analogico/digitale DDC112 104
4.3.1 Ciclo base d'integrazione 106
4.3.2 Tensione di riferimento 109
4.3.3 Risposta in frequenza 110
4.3.4 Funzionamento in modalità "*test*" 110
4.3.5 Modalità di operazione "continua" e "non-continua" 111
4.3.5.1 Esempi di diagrammi di temporizzazione 113
4.3.5.2 Cambiamento di modalità di operazione 115
4.3.6 Utilizzo di più DDC112 in cascata 116
4.3.7 Trasferimento dei dati 116
4.3.8 Sequenza di *power-up* 118

4.4 Microcontrollore LPC2103 119
4.4.1 Architettura del microcontrollore LPC2103 120
4.4.2 Processore ARM7TDMI-S 122
4.4.2.1 *Pipeline* 122
4.4.2.2 Registri 122
4.4.2.3 Modalità privilegiate 125
4.4.2.4 Set di istruzioni ARM 126
4.4.2.4.1 *Branching* 128
4.4.2.4.2 Elaborazione dati 128
4.4.2.4.3 Copia di registri 129
4.4.2.4.4 Istruzioni di scambio (*swap*) 130
4.4.2.4.5 Modifica dei registri di stato 130
4.4.2.4.6 *Software interrupt* 131
4.4.2.4.7 Unità MAC 131
4.4.2.5 Set di istruzioni *Thumb* 131
4.4.3 Periferiche di sistema 133
4.4.3.1 Struttura dei *bus* 133
4.4.3.2 Mappa di memoria 134
4.4.3.3 Programmazione dei registri 135
4.4.3.4 Modulo di accelerazione di memoria 135
4.4.3.5 Programmazione della memoria *flash* 137
4.4.3.6 Circuito ad aggancio di fase 137
4.4.3.7 Gestione del consumo di potenza 139
4.4.3.8 Sistema di gestione degli *interrupt* IRQ e FIQ 139
4.4.3.8.1 Blocco di connessione *pin* 139
4.4.3.8.2 *Pin* riservati ad *interrupt* esterni 140
4.4.3.8.3 Controllore per *interrupt* vettorizzati (VIC) 140
4.4.3.8.4 *Interrupt* FIQ 141
4.4.3.8.5 *Interrupt* IRQ vettorizzati 141
4.4.3.8.6 *Interrupt* IRQ non vettorizzati 142
4.4.3.8.7 Simulazione di un *interrupt* 143
4.4.3.8.8 *Interrupt* annidati 143
4.4.4 Periferiche di servizio (*user*) 144
4.4.4.1 *Pin* I/O *general purpose* (GPIO) 144

4.4.4.2 *Timer general purpose* 144
4.4.4.3 Modulatore PWM 146
4.4.4.4 *Real Time Clock* (RTC) 146
4.4.4.5 *Watchdog* 147
4.4.4.6 UART 148
4.4.4.7 Interfaccia I^2C 151
4.4.4.8 Interfaccia SPI 152
4.4.4.9 Convertitore analogico/digitale (ADC) 153

4.5 Progetto delle CPLD per la gestione del banco di DDC112 154
4.5.1 Generazione del segnale CONV 155
4.5.2 Generazione di segnali necessari al prelievo dei dati (*data retrieval*) 158
4.5.3 Gestione della modalità "*test*" 161
4.5.4 Gestione della sequenza di *power-up* 164
4.5.5 Soluzione dei problemi relativi al cambiamento di modalità 165
4.5.6 Verifica del funzionamento della CPLD per la gestione del banco di DDC112 167

4.6 Banco di memoria 172
4.6.1 Gestione delle fasi di scrittura e lettura 176
4.6.1.1 Fase di scrittura 177
4.6.1.2 Fase di lettura 177
4.6.2 Inizializzazione e *test* del banco di memoria 178
4.6.2.1 Periodo di *reset* (i) 179
4.6.2.2 Periodo di scrittura (ii) 181
4.6.2.3 Periodo di lettura (iii) 183
4.6.3 Verifica di funzionamento della CPLD per la gestione del banco di memoria 184
4.6.3.1 Simulazione della fase di scrittura 184
4.6.3.2 Simulazione della fase di lettura 185
4.6.3.3 Simulazione delle fasi di inizializzazione e *test* 185
4.6.4 Codici ABEL dei blocchi progettati 187

4.7 Interfacciamento per tastiera a matrice 189
4.7.1 Interfacciamento analogico 190
4.7.1.1 Codice di gestione della tastiera a matrice 191
4.7.2 Interfacciamento digitale 193
4.7.2.1 Metodo a scansione 193
4.7.2.2 Metodo con scambio del verso di lettura 195
4.7.3 Schema d'interfacciamento tra tastiera e μC LPC2103 196

4.8. Interfacciamento con *display* alfanumerici a cristalli liquidi 197
4.8.1 Il *controller* HD44870 198
4.8.2 Schema d'interfacciamento tra *display* LCD e μC LPC2103 202
4.8.3 *Software* di gestione per LCD con *controller* HD44780 202
4.8.3.1 Funzione '*InitLCD*' di inizializzazione 206
4.8.3.2 Funzione '*WriteByte*' 208
4.8.3.3 Funzione '*PutCommand*' 209
4.8.3.4 Funzione '*WriteChar*' 210
4.8.3.5 Funzione '*WriteString*' 210
4.8.3.6 Funzione '*Write_2digitsval*' 211
4.8.3.7 Funzione '*Write_ndigitsval*' 211
4.8.3.8 Programma di *test* per l'LCD 212

4.9 Interfacciamento con il PC 213
4.9.1 L'interfaccia seriale RS-232 213
4.9.1.1 Caratteristiche di un segnale RS-232 213
4.9.1.2 Parametri elettrici della RS-232 e compatibilità TTL 214
4.9.1.3 Connettori RS-232 215

4.9.2 Generalità sui dispositivi UART 216

4.10 Descrizione del *layout* 217

4.11 Procedure di *test* e collaudo dell'elettronica 225

Capitolo V - Risultati sperimentali per rivelatori multipixel

5.1 Rivelatori multipixel 1D e 2D 229

5.2 Caratterizzazione preliminare a 8 canali 232
5.2.1 Sensori di posizione 232
5.2.2 Emulazione di profili UV 234

5.3 *Imaging* di sorgenti UV di bassa intensità 235

5.4 *Imaging* di laser UV a eccimeri 241

5.5 *Imaging* di sorgenti X 244

Conclusioni 247

Bibliografia 249

Partecipazione a conferenze nazionali e internazionali 253

Pubblicazioni 255

Introduzione

L'attività di ricerca presentata in questa tesi di dottorato è focalizzata sulla progettazione, realizzazione e verifica sperimentale di un sistema completo per la diagnostica di radiazioni UV e X, con particolare riferimento all'analisi della forma e dell'intensità del fascio. In questi ultimi anni, nel campo della ricerca sui fotorivelatori, lo studio di profilometri per radiazioni UV ha rappresentato motivo di approfondimento e sviluppo, grazie all'impiego sempre più consistente di sorgenti come laser e lampade ad eccimeri in numerose applicazioni come la litografia di circuiti integrati a semiconduttore in scala d'integrazione VLSI-ULSI, la realizzazione di dispositivi elettromeccanici miniaturizzati (MEMS), la chirurgia oculistica refrattiva per la correzione di difetti come la miopia, per arrivare infine all'UV *imaging*, tecnica che prevede l'analisi della radiazione UV riflessa dalla superficie dell'oggetto sotto esame. Di natura diversa, ma non di minore importanza, sono le applicazioni per le quali è richiesto un monitoraggio costante e in tempo reale delle radiazioni X: radioterapia, microscopia e spettroscopia a raggi X, diagnostica a raggi X (*X-ray imaging*) in campo medico, cristallografia, astrofisica, sicurezza, radiografia industriale.
I profilometri attualmente esistenti in commercio, sia per i fasci UV che per la radiazioni X, sono per la maggior parte basati su rivelatori in silicio di varia natura (fotodiodi in silicio *UV-enhanced*, matrici CCD, etc.): la profonda conoscenza del materiale e l'avanzata (e ben collaudata) tecnologia hanno permesso lo sviluppo e il successo commerciale e industriale di dispositivi in silicio, nonostante le sue limitazioni intrinseche. Per quel che riguarda le radiazioni UV, ad esempio, è penalizzante per il silicio il fatto di non essere trasparente alle regioni VIS e IR dello spettro: questo ha reso necessario in molti casi l'impiego, nei profilometri in silicio, di costosi sistemi ottici per la conversione della lunghezza d'onda. Nell'ambito dell'analisi di sorgenti X, il maggiore punto debole del silicio, in particolar modo per sorgenti di elevata intensità, è senza dubbio la non elevata resistenza al danno da radiazione, il che necessita la presenza di attenuatori e/o di scintillatori, con un'inevitabile perdita di risoluzione spaziale ed energetica.
Per i motivi sopraelencati, è d'indubbio interesse scientifico lo studio di materiali alternativi al silicio per la realizzazione di rivelatori per profilometria UV e X. I rivelatori presentati in questo lavoro di tesi sono stati realizzati in diamante, un materiale che per molti aspetti può essere considerato ideale per questa tipologia di applicazioni: la sua elevata *gap* (5.47 eV) lo rende trasparente alle radiazioni VIS-IR, minimizzando la corrente di buio e rendendo non necessaria la presenza di filtri antiriflesso o convertitori di lunghezza d'onda; l'alta mobilità e velocità di saturazione dei portatori di carica rendono il diamante in grado di avere una veloce fotorisposta (pochi ns), consentendo di realizzare rivelatori capaci di monitorare in tempo reale sorgenti UV impulsate come un laser a eccimeri; infine, l'alta energia di coesione e soprattutto l'elevata conducibilità termica permettono di minimizzare il danno da radiazione, e questo permette di fare a meno dell'utilizzo di attenuatori, sia per il monitoraggio di sorgenti laser di potenza che per l'analisi di radiazioni X di elevata intensità. In

sostanza, a differenza dei comuni profilometri in silicio, nel prototipo in diamante realizzato nell'ambito di quest'attività di ricerca non ci sono elementi aggiuntivi (attenuatori, filtri, etc.) fra la sezione d'uscita della sorgente e il piano del rivelatore, minimizzando dunque l'introduzione di componenti che possano alterare la natura del fascio, o che possano in ogni caso rendere difficoltoso il recupero delle informazioni su forma e intensità.
Una lunga parte dell'attività di ricerca è stata inoltre dedicata alla progettazione di un'elettronica completa a componenti discreti per il condizionamento e l'elaborazione dei segnali in corrente provenienti dai rivelatori: si tratta di un'elettronica multicanale, adatta dunque ad essere utilizzata come elettronica di *front-end* e acquisizione per rivelatori multipixel. Visto l'ingombro, e soprattutto gli alti costi, dei sistemi elettronici multicanale disponibili in commercio come strumentazione da laboratorio, l'obiettivo principale è stato quello di rendere la testa di misura e l'elettronica di elaborazione il più possibile compatte e dal costo limitato, pur mantenendo un livello di prestazioni compatibile con le esigenze della profilometria UV e X (*front-end* veloce, basso rumore, e un elevato *throughput* dei dati per il monitoraggio in tempo reale).

La tesi si articola in 5 capitoli, dei quali qui di seguito viene data una breve descrizione:

- Nel *capitolo I* viene fatta una panoramica delle sorgenti e dei rivelatori per radiazioni UV e X di maggior interesse in ambito scientifico e industriale, mettendo in luce soprattutto gli elementi di innovazione introdotti dall'utilizzo di materiali alternativi come il diamante.
- Il *capitolo II* è interamente dedicato al diamante policristallino, con particolare riferimento ai meccanismi di generazione e raccolta delle cariche nei fotorivelatori in diamante, nonché ai fenomeni di assorbimento delle radiazioni ottiche e dei raggi X.
- Nel *capitolo III* viene presentata l'attività di caratterizzazione preliminare dei rivelatori: prima della realizzazione di strutture multipixel, si è resa necessaria un'analisi del rivelatore a singolo pixel, in termini di caratterizzazione del materiale, studio dei contatti metallici, valutazione della fotorisposta a radiazioni UV e X di varia natura.
- Nel *capitolo IV* viene dettagliatamente descritto il sistema elettronico multicanale per l'acquisizione e il condizionamento dei segnali in corrente provenienti dal rivelatore: particolare cura è stata dedicata nell'esposizione delle varie fasi del progetto, dalla descrizione dello schema a blocchi alle fasi finali di *test* e collaudo.
- Il *capitolo V* è infine dedicato alla presentazione e discussione dei risultati sperimentali ottenuti con i rivelatori multipixel in tre diversi campi d'applicazione: *imaging* di sorgenti UV continue di debole intensità, *imaging* di laser UV a eccimeri, e *imaging* di fasci prodotti da tubi a raggi X.

Capitolo I

Radiazioni UV e X: applicazioni, sorgenti e rivelatori

1.1 I raggi UV e X nello spettro elettromagnetico

La radiazione ultravioletta (UV) comprende per definizione le lunghezze d'onda comprese fra 10 nm e 400 nm, che corrispondono ad energie fotoniche comprese fra 3.1 eV e 124 eV. A causa di questo ampio *range* di energia, le proprietà della radiazione UV e la sua interazione con la materia cambiano sensibilmente lungo lo spettro: per questo motivo la regione UV è stata suddivisa in UV-A (400 ÷ 320 nm), UV-B (320 ÷ 280 nm) e UV-C (280 ÷ 10 nm). La suddivisione si riferisce in particolar modo alla diversa attività biologica della radiazione UV [1]: i raggi UV-A penetrano l'atmosfera senza un sostanziale assorbimento ed hanno un scarsa attività biologica (riferibile soprattutto ad un invecchiamento prematuro della pelle umana); i raggi UV-B sono parzialmente assorbiti dall'atmosfera, e interagiscono sensibilmente con i tessuti, fino a causare diversi tumori della pelle; i raggi UV-C non raggiungono la superficie terrestre a causa dell'assorbimento da parte dell'ozono contenuto nella parte superiore dell'atmosfera, ma sono altamente pericolosi per gli organismi viventi, in quanto fortemente assorbiti dalle proteine e dagli acidi nucleici. Un'ulteriore divisione della regione UV si riferisce alle lunghezze d'onda inferiori ai 200 nm, meglio note come VUV (*Vacuum* UV) a causa del forte assorbimento da parte dell'ossigeno.

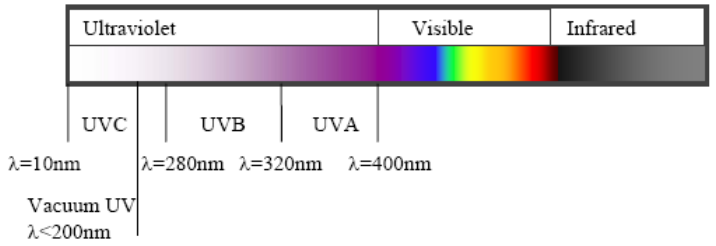


Fig. 1.1 *Suddivisione della regione ultravioletta*

I raggi X sono radiazioni elettromagnetiche con lunghezza d'onda compresa fra 1 nm e 10^{-4} nm (energia nell'intervallo 1.24 keV ÷ 12.4 MeV). La lunghezza d'onda dei raggi X è in genere maggiore di quella dei raggi γ, tuttavia la distinzione fra i due tipi di raggi

non è data tanto dal valore della lunghezza d'onda quanto dalla loro origine: in particolare, i raggi X sono dovuti a fenomeni che avvengono al di fuori dei nuclei atomici, mentre i raggi γ sono di origine nucleare.

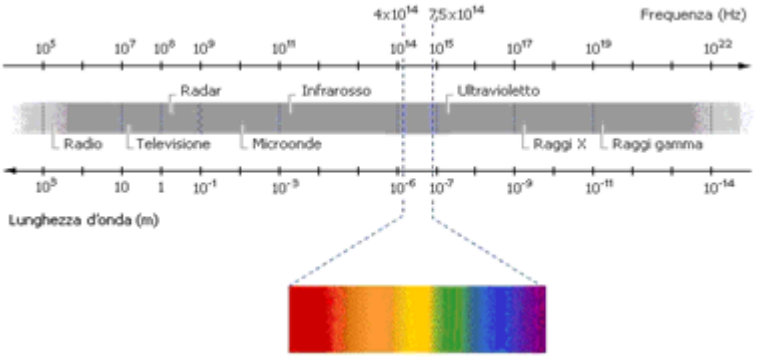


Fig. 1.2 *Lo spettro elettromagnetico*

1.2 Principali applicazioni delle radiazioni UV

L'applicazione più importante delle radiazioni UV è sicuramente nell'industria microelettronica, e più precisamente nella litografia di circuiti integrati VLSI-ULSI. La minima *feature size* che si può ottenere in un processo fotolitografico, com'è noto, è proporzionale alla lunghezza d'onda utilizzata per l'esposizione del resist: questo, unito alla continua miniaturizzazione dei dispositivi (è ancora valida la legge di Moore, e secondo le previsioni lo sarà ancora per molti anni), spinge all'uso di radiazioni di lunghezza d'onda sempre più corta. Il laser a eccimeri KrF* (248 nm), grazie ad una buona efficienza di emissione, sostituì già alla fine degli anni '90 le lampade a mercurio; il passaggio successivo fu poi l'adozione del laser a eccimeri ArF* (193 nm), che rimane tuttora il laser più utilizzato per la produzione di dispositivi elettronici ULSI allo stato dell'arte [2].

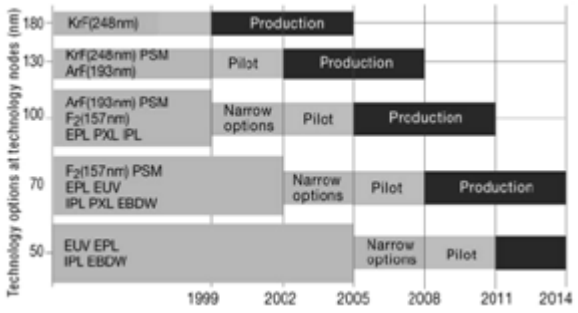


Fig. 1.3 *Roadmap internazionale sulle sorgenti da utilizzare per la fotolitografia dei circuiti integrati*

Naturalmente, in questo contesto è di cruciale importanza studiare e realizzare rivelatori per il lontano UV che siano in grado di effettuare una diagnostica *in situ* e in tempo reale del fascio UV utilizzato: per il momento si utilizzano rivelatori a semiconduttore in silicio (vedi par. 1.6.2), soprattutto per i bassi costi e la tecnologia consolidata, fattori che compensano la scarsa adattabilità del silicio a queste applicazioni. Tuttavia, andando a lunghezze d'onda ancora minori (come previsto dalla *roadmap* internazionale, che prevede nei prossimi anni di passare a radiazioni UV con $\lambda < 100$ nm), saranno insormontabili le limitazioni intrinseche del silicio, e dunque c'è una forte richiesta in campo industriale di rivelatori basati su materiali alternativi.
Un'altra applicazione di grande importanza delle radiazioni UV (in particolar modo del laser a eccimeri ArF*) sta nella chirurgia oculistica. Il laser ad eccimeri (vedi par. 1.4.2) emette una radiazione UV ad alta intensità e viene utilizzato per trattare la cornea [3]: guidato e controllato in modo opportuno, il laser asporta tessuto in quantità di pochi μm per impulso, vaporizzandolo istantaneamente. Il laser può così modificare il profilo della cornea, ad esempio appiattendone la parte centrale in modo preciso e regolare per la correzione della la miopia, oppure agendo in maniera diversa per correggere altri difetti come l'astigmatismo o l'ipermetropia. Il laser ad eccimeri ArF* può essere utilizzato in due modi: sulla superficie anteriore della cornea (in tal caso la procedura si chiama PRK o *fotoablazione corneale di superficie*) oppure all'interno della cornea, dopo aver eseguito una microscopica incisione semicircolare che consente di sollevare un sottile strato di tessuto (in tal caso la procedura si chiama LASIK o *cheratomileusi con laser ad eccimeri*). Se il difetto è di grado medio ed elevato il trattamento PRK non è sufficiente a correggerlo adeguatamente e occorre eseguire l'intervento all'interno della cornea (LASIK). L'operazione LASIK consta di due parti (vedi fig. 1.4): la prima consiste nel tagliare, con uno strumento chiamato *microcheratomo*, un sottile strato semicircolare di tessuto superficiale (per un diametro di circa 9 mm e ad una profondità di circa il 25% dello spessore totale della cornea), che viene poi sollevato e aperto; la seconda parte consiste nell'eseguire, nella parte interna della cornea così esposta, il trattamento laser ad eccimeri. La parte di cornea sollevata viene poi riposizionata senza necessità di dare punti di sutura.

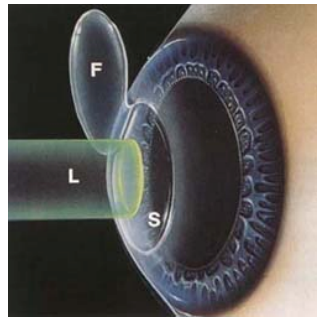


Fig. 1.4 *LASIK: fasi dell'intervento di chirurgia oculistica refrattiva*

Un'ultima importante applicazione delle radiazioni UV da menzionare è l'UV *imaging*, tecnica che prevede l'analisi della radiazione UV riflessa dalla superficie dell'oggetto sotto esame [4]. Una delle applicazioni più comuni dell'UV *imaging* è l'analisi della

presenza di graffi su una superficie: grazie alla corta lunghezza d'onda, le radiazioni UV tendono ad interagire più efficacemente con le caratteristiche superficiali di un oggetto, se paragonate alle radiazioni VIS-IR. In questo modo, ad esempio, graffi che non risultano visibili in un'immagine raccolta con luce visibile, possono esserlo in un'immagine UV: per questo motivo, sistemi automatici di UV *imaging* per l'analisi di graffi e altre imperfezioni sono molto utilizzati nella caratterizzazione di superfici ottiche come lenti o finestre, oppure nell'industria microelettronica per trovare difetti submicrometrici nei *wafer* di silicio prima di iniziare la scrittura dei circuiti integrati. In sostanza, le applicazioni dell'UV *imaging* possono essere molteplici: immagini biomediche, analisi di contaminanti superficiali, sviluppo di OLED (*Organic* LED), *imaging* cellulare, sviluppo di dispositivi a microfluido, *imaging* minerale, analisi della superficie di opere d'arte, identificazione di tinte su tessuti, sicurezza domestica, individuazione di contraffazioni in ambito forense, etc.
Naturalmente, in tutte le applicazioni UV menzionate in questo paragrafo, è di cruciale importanza assicurare l'omogeneità della radiazione UV su tutta la superficie d'interesse, ed è dunque particolarmente importante avere a disposizione sistemi in grado di monitorare efficacemente la forma e l'intensità del fascio: lo sviluppo di un sistema di questo tipo è proprio uno degli obiettivi di quest'attività di ricerca.

1.3 Principali applicazioni dei raggi X

E' noto che l'applicazione di gran lunga più importante dei raggi X si ha in campo medico, come strumento diagnostico (*imaging* a raggi X) dalle molteplici varianti: si va dalle semplici radiografie a tecniche sempre più sofisticate di analisi, fra le quali è doveroso menzionare la TAC (Tomografia Assiale Computerizzata) e la fluoroscopia. In questo tipo di tecniche, che si basano tutte sul parziale assorbimento dei raggi X da parte di ossa e tessuti molli, si utilizzano necessariamente raggi X con energie superiori a 20 keV (*hard X-rays*), in quanto i raggi X di bassa energia (*soft X-rays*) vengono totalmente assorbiti dal corpo umano, e questo aumenterebbe inevitabilmente la dose assorbita. Un altro importante utilizzo dei raggi X in campo medico consiste nella radioterapia, tecnica per il trattamento di tumori basata sul bombardamento delle cellule tumorali, che fa uso di raggi X di energia più elevata rispetto alle tecniche diagnostiche.
Altre applicazioni dei raggi X degne di nota sono:

- Cristallografia a raggi X: il *pattern* prodotto dalla diffrazione subita da un raggio X attraversando un reticolo cristallino viene registrato e analizzato per scoprire la natura del reticolo.
- Astronomia a raggi X: è una branca dell'astronomia che si occupa di rivelare e analizzare l'emissione di raggi X dai corpi celesti.
- Analisi microscopica a raggi X: è una tecnica che fa uso di raggi X "molli" (con energie inferiori ai 12 keV) per produrre immagini dettagliate di oggetti molto piccoli.

- Fluorescenza a raggi X: è una tecnica in cui i raggi X sono generati all'interno dell'oggetto sotto esame, e poi rivelati per poter ottenere informazioni composizionali.
- XPS (*X-ray Photoelectron Spectroscopy*): è una tecnica di microanalisi chimica che sfrutta l'effetto fotoelettrico, ovvero l'emissione di elettroni dall'oggetto in esame in seguito all'interazione con un raggio X.
- Radiografia industriale: in questa tecnica si fa utilizzo di tecniche radiodiagnostiche per ispezionare componenti industriali, in particolar modo nelle zone di saldatura.
- Sicurezza: è obbligatorio l'utilizzo della diagnostica a raggi X negli aeroporti, per l'ispezione dei bagagli prima dell'imbarco, e anche in molte zone di confine per il controllo di mezzi di trasporto merci.

Per ognuna delle applicazioni sopra indicate esiste una vasta e collaudata gamma di rivelatori, che vanno dai rivelatori a gas a quelli a scintillazione, fino ai rivelatori a semiconduttore: verranno presentati e discussi in dettaglio nel par. 1.7.

1.4 Sorgenti di radiazione UV

Essendo numerose le applicazioni industriali e scientifiche delle radiazioni UV, altrettanto nutrito è il panorama di sorgenti UV a disposizione. In questo paragrafo ne verranno descritte alcune:

- Lampade a Xenon-Mercurio, utilizzate in genere insieme ad un monocromatore, per ottenere luce incoerente UV a bassa intensità.
- Laser a eccimeri: sono sicuramente le sorgenti UV dalle applicazioni più interessanti, che vanno dalla chirurgia oculistica rifrattiva alla litografia di circuiti ULSI e dispositivi MEMS con *feature sizes* submicrometriche.
- Lampade a eccimeri, introdotte recentemente in laboratorio per la stimolazione di alcuni processi chimici, e utilizzate in ambito industriale nella fabbricazione di *display* LCD e OLED.

1.4.1 Lampade a Xenon-Mercurio

Le lampade a Xenon-Mercurio (Xe-Hg) utilizzate per la generazione di luce UV sono lampade a scarica ad arco corto e uscita continua, che sfruttano una miscela di gas Xe ionizzato nel quale è dispersa un piccola quantità di Hg: la luce prodotta è di colore bianco-blu, con presenza di righe molto intense nell'UV.
Le lampade Xe-Hg sono costituite da un tubo di quarzo alle estremità del quale sono posizionati due elettrodi di tungsteno, drogato con torio per aumentare l'emissione termoionica di elettroni: il tubo viene dapprima svuotato, e poi riempito con la miscela di Xe-Hg. Per ottenere la massima efficienza, il gas è mantenuto all'interno del tubo ad una pressione molto alta (fino a 30 atmosfere).

La tensione d'innesco della scarica ad arco è nell'intervallo 25 ÷ 50 kV: una volta innescato l'arco con questo impulso di alta tensione, si ha generazione di luce soprattutto nella zona di plasma vicino agli elettrodi. Il volume di generazione della luce appare come due coni che si intersecano, con una diminuzione esponenziale dell'intensità luminosa dagli elettrodi verso il centro del tubo.
La piccola dimensione dell'arco permette di focalizzare la luce emessa dalla lampada con una buona precisione: per questo motivo le lampade Xe-Hg di piccole dimensioni (fino a 10 W) vengono usate in ottica, ad esempio per l'illuminazione di microscopi o altri strumenti diagnostici. Ad ogni modo, gli utilizzi principali delle lampade Xe-Hg sono in ambito medico (terapia UV, sterilizzazione di oggetti) e chimico (generazione di ozono).

1.4.2 Laser a eccimeri

Il laser a eccimeri è un tipo di laser UV comunemente usato in chirurgia oculistica e nella tecnologia dei semiconduttori. Utilizza tipicamente una miscela costituita da un gas nobile (argon, kripton, xenon) e un gas reattivo (come gli alogeni fluoro e cloro): sotto appropriate condizioni di stimolazione elettrica, viene creata una pseudo-molecola, l'eccimero appunto (che sta per "dimero eccitato"), che può esistere solo in uno stato eccitato, dal quale decade emettendo fotoni UV.
La luce UV emessa da un laser a eccimeri è ben assorbita da tessuti biologici e composti organici. Piuttosto che bruciare o tagliare un materiale, il laser a eccimeri si limita a rompere i legami molecolari superficiali, disintegrando le superfici in modo controllato (ablazione laser): quindi un laser a eccimeri può rimuovere strati molto sottili di materiale senza riscaldarlo o cambiare le caratteristiche della zona al di sotto della parte rimossa, che rimane intatta. Per queste proprietà, i laser a eccimeri sono utilizzati in campo litografico (ULSI e MEMS) e in operazioni di chirurgia oculistica (tecnica LASIK).

Excimer	**Wavelength**	**Relative Power [mW]**
Ar_2*	126 nm	
Kr_2*	146 nm	
F_2*	157 nm	10
Xe_2*	172 & 175 nm	
ArF*	193 nm	60
KrF*	248 nm	100
XeBr*	282 nm	
XeCl*	308 nm	50
XeF*	351 nm	45
KrCl*	222 nm	25
Cl_2*	259 nm	
N_2	337 nm	5

Tab. 1.1 *Principali tipologie di laser a eccimeri*

Sostanzialmente, in un laser a eccimeri, l'azione laser avviene perché l'eccimero è caratterizzato da uno stato eccitato legante (o associativo) e da uno stato fondamentale repulsivo (o dissociativo): questo è dovuto al fatto che i gas nobili sono altamente inerti, e solitamente non formano composti. Tuttavia in uno stato eccitato, indotto da una scarica elettrica o da fasci elettronici ad alta energia, i gas nobili possono formare temporaneamente composti, o fra loro o con alogeni (fluoro, cloro, etc.). Il composto eccitato cede poi la sua energia in eccesso dando luogo ad emissione di fotoni (spontanea o stimolata), e torna al suo stato fondamentale repulsivo, nel quale molto rapidamente (ordine dei ps) si dissocia nei due atomi di partenza: questo determina l'inversione di popolazione fra i due livelli necessaria per l'azione laser.
La lunghezza d'onda di un laser a eccimeri dipende dalle molecole usate, ma è solitamente nell'UV (vedi tabella 1.1).

1.4.3 Lampade a eccimeri

L'utilizzo più importante delle lampade ad eccimeri in ambito industriale è la pulizia a secco delle superfici nella fabbricazione di dispositivi a cristalli liquidi (LCD) e di OELD (*Organic ElectroLuminescence Displays*): in queste due applicazioni hanno ormai sostituito definitivamente le lampade Xe-Hg, che determinavano un deleterio surriscaldamento del materiale sottoposto a *dry cleaning*. Il principio alla base del funzionamento di una lampada a eccimeri è lo stesso di un laser a eccimeri, e per la sua spiegazione si rimanda al par 1.4.2.
Un esempio di lampada a eccimeri commerciale [5] è mostrata in fig. 1.5:

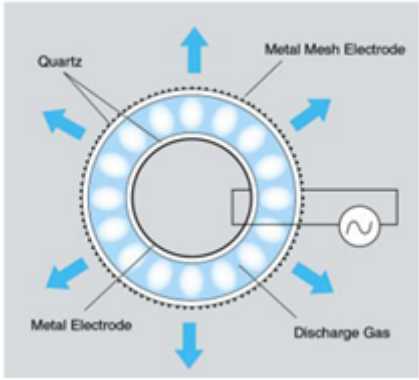


Fig. 1.5 *Struttura di una lampada ad eccimeri*

La lampada è costituita da due tubi di quarzo concentrici: il primo elettrodo è sulla superficie interna del tubo interno, mentre il secondo elettrodo è sulla superficie esterna del tubo esterno. Ad entrambi gli elettrodi è fornita potenza elettrica ad alta frequenza (fino a pochi MHz): nello spazio compreso fra i due tubi viene generata la scarica, e la radiazione prodotta dal diseccitamento dell'eccimero viene emessa in tutte le direzioni. Ovviamente, l'elettrodo esterno deve essere costituito da una griglia per far passare la luce UV prodotta all'interno della lampada.

Le lampade a eccimeri più commercializzate sono quelle a dimero di gas nobile: Ar_2* (126 nm), Kr_2* (146 nm) e Xe_2* (172 nm). Per lunghezze d'onda maggiori, esistono lampade a eccimeri a KrCl* (222 nm) e XeCl* (308 nm).

1.5 Sorgenti di raggi X

Le sorgenti di raggi X di maggior interesse in ambito commerciale e di laboratorio sono:

- Tubi ad alto vuoto, detti di *Coolidge*: sono le sorgenti più utilizzate in cristallografia.
- Sorgenti radioattive: i raggi X sono prodotti dal decadimento di isotopi naturali o artificiali; questo tipo di sorgenti è utilizzato soprattutto per la calibrazione di rivelatori per spettroscopia.
- Luce di sincrotrone: i raggi X sono prodotti da elettroni relativistici in moto lungo un'orbita di forma opportuna.
- Sorgenti plasma-laser: i raggi X sono prodotti attraverso un plasma che viene generato dal bombardamento di bersagli con fasci laser ad alta energia; è una tecnica che permette di produrre efficacemente radiazioni di bassa energia (*soft X-rays*).

1.5.1 Tubo di *Coolidge*

La più comune sorgente di raggi X è il tubo a vuoto (o di *Coolidge*): si compone di un filamento di tungsteno (che viene portato ad alta temperatura mediante una corrente elettrica, così da emettere elettroni per effetto termoionico), un catodo ed il *target* metallico (anodo); il tutto è contenuto in un tubo in cui è stato creato un vuoto spinto.

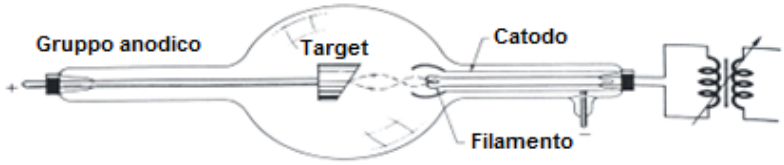


Fig. 1.6 *Schema di un tubo di Coolidge*

Si hanno poi due alimentazioni distinte ed esterne al tubo: una serve per dare tensione al filamento, mentre l'altra ha il compito di fissare la differenza di potenziale tra catodo ed anodo. Il catodo avvolge il filamento, ed è posto ad un'elevata tensione negativa, comportandosi in questo modo come una "lente elettronica": gli elettroni emessi dal filamento vengono focalizzati sul *target*, dove arrivano fortemente accelerati dalla differenza di potenziale. Nel tubo di *Coolidge* la pressione deve dunque essere molto

bassa (10^{-3} Pa), al fine di massimizzare il cammino libero medio degli elettroni: un numero sensibile di collisioni rappresenterebbe infatti un fattore di disturbo alla controllabilità del fenomeno.
In seguito all'impatto elettronico sul *target*, si ha la produzione di raggi X caratterizzati da uno spettro di emissione composito, dato dalla sovrapposizione di uno spettro continuo (*radiazione di frenamento* o *Bremsstrahlung*) e di righe spettrali caratteristiche del materiale del *target* (*spettro caratteristico*), come si può vedere in fig. 1.7.

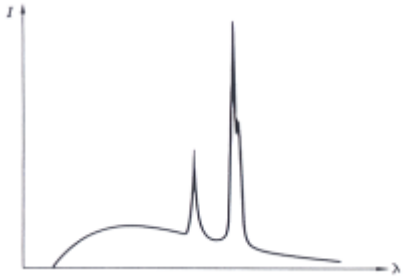


Fig. 1.7 *Generico spettro d'emissione di un tubo di Coolidge*

Il parametro di controllo più importante dell'intero processo è la temperatura del filamento, che dipende quasi esclusivamente dalla corrente che vi scorre: ad una determinata temperatura, il numero massimo di elettroni emessi è fisso, pertanto la tensione applicata al tubo incrementa solamente la loro velocità (e quindi l'energia cinetica) senza influenzare la corrente elettrica. E' questo il grande vantaggio del tubo di *Coolidge*: l'unico parametro di controllo, una volta raggiunta la zona di saturazione della corrente, è la temperatura del filamento. L'efficienza di generazione dei raggi X, definita come il rapporto tra l'energia della radiazione X emessa e quella degli elettroni assorbiti, segue la legge empirica:

$$\eta = 1.4 \cdot 10^{-9} ZV \qquad [I.1]$$

dove Z rappresenta il numero atomico del materiale che costituisce il *target* e V la tensione applicata al tubo. In condizioni di lavoro normali, tale efficienza è pari a pochi punti percentuali, il che vuol dire che la quasi totalità dell'energia posseduta dagli elettroni proiettile è convertita in calore, con la conseguenza che il bersaglio potrebbe riscaldarsi troppo e degradarsi nel tempo: per questo nei moderni tubi di *Coolidge* il *target* viene raffreddato.
Importante è anche la regolazione dello *spot focale* (o *macchia focale*), che rappresenta la porzione di superficie del *target* che viene investita dal fascio elettronico. In radiografia le sue dimensioni sono essenziali per la risoluzione delle immagini a raggi X: infatti, quanto minori sono le dimensioni dello *spot* focale tanto maggiore è la quantità di dettagli che si riescono a discriminare (e si ottiene dunque un'immagine più risolta). Allo scopo di ridurre il più possibile le dimensioni dello *spot focale*, il *target* è

montato su di un piano inclinato rispetto alla direzione degli elettroni incidenti (fig. 1.8): tutto va come se la sorgente fosse equipaggiata con un filamento di dimensioni minori rispetto a quelle reali. Le dimensioni dello *spot* focale dipendono anche dall'intensità della corrente utilizzata: basse correnti di tubo (100 mA) consentono di utilizzare filamenti piccoli e dunque di ridurre le dimensioni dello *spot* focale, ma quando si devono necessariamente utilizzare correnti di tubo più elevate (200 mA o maggiori), ad esempio per ridurre i tempi di esposizione, gli *spot* focali saranno inevitabilmente più ampi.

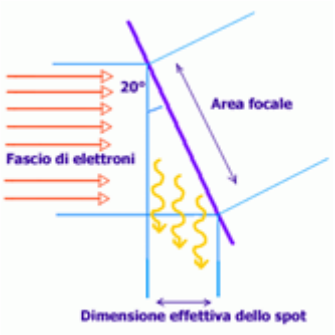


Fig. 1.8 *Target inclinato per ridurre lo spot focale*

1.5.2 Sorgenti radioattive

Le sorgenti radioattive sono utilizzate in diffrattometria per la calibrazione dei rivelatori di raggi X, in quanto presentano il vantaggio di essere molto stabili nel tempo e con una velocità di decadimento ben nota. Un pregio di questo tipo di sorgenti è la purezza dello spettro di emissione, che risulta essere praticamente monocromatico.

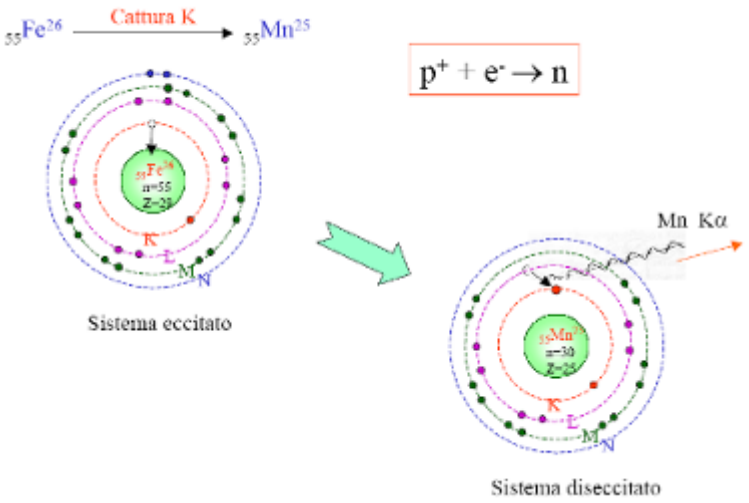


Fig. 1.9 *Schema di generazione X da una sorgente radioattiva di* ^{55}Fe

Per la generazione di raggi X vengono utilizzati isotopi radioattivi che decadono per *cattura elettronica* (fig. 1.9); in questo tipo di decadimento, un nucleo di numero atomico *Z* cattura un elettrone dei livelli energetici più bassi, trasformandosi in un nucleo di numero atomico *Z-1*: l'avvenuta trasformazione nucleare si manifesta per mezzo dei raggi X emessi nel successivo riassestamento elettronico, e dunque la radiazione emessa è caratteristica del nucleo generato.
Una sorgente radioattiva compie un certo numero di disintegrazioni al secondo ed in ognuna di queste il radionuclide si trasforma in un altro emettendo radiazioni (non solo X, ma anche *alfa*, *beta*, *gamma* a seconda del tipo di decadimento). L'unità di misura dell'attività di una sorgente radioattiva è il *Becquerel* (Bq), che si riferisce ad una disintegrazione al secondo, ma ancora utilizzato è il vecchio Curie (Ci), che indicava $3.7 \cdot 10^{10}$ disintegrazioni al secondo. C'è da precisare che l'attività non indica il numero di particelle emesse al secondo: ogni disintegrazione può comportare anche meno di una o più di una emissione al secondo (dipende dallo schema di decadimento). L'attività delle sorgenti radioattive diminuisce nel tempo con una legge esponenziale: per comodità si usa l'*emivita* o *tempo di dimezzamento fisico* $T_{1/2}$, che è il tempo necessario per ridurre della metà sia l'attività che il numero di radionuclidi che devono ancora decadere, ed è caratteristico unicamente del tipo di sorgente o radionuclide.
Nella tabella 1.2 sono mostrati i principali isotopi radioattivi, la specie generata dal decadimento e l'energia della radiazione prodotta.

Nuclide radioattivo	**Elemento**	**$K\alpha_1$ (keV)**
^{37}Ar	Cl	2.622
^{51}Cr	V	4.952
^{55}Fe	Mn	5.898
^{71}Ge	Ga	9.251
^{103}Pd	Rh	20.214
^{109}Cd	Ag	22.16
^{125}I	Te	27.47
^{131}Cs	Xe	29.80

Tab. 1.2 *Principali isotopi radioattivi che emettono raggi X*

1.5.3 Luce di sincrotrone

La *luce di sincrotrone* (o *radiazione di sincrotrone*) è una radiazione elettromagnetica generata da elettroni e positroni ad alta energia che si muovono su traiettorie circolari, in un campo magnetico relativamente debole, a velocità prossime alla velocità della luce. Tanto più elevata è la velocità della particella, tanto minore è la lunghezza d'onda della radiazione emessa. La luce di sincrotrone è così chiamata perché viene solitamente prodotta per mezzo di un sincrotrone, ovvero un acceleratore circolare di particelle.
Una particella carica, avente energia *E* e massa *m*, che si muove su di un'orbita circolare di raggio *R* ad una velocità costante *v*, irraggia su un angolo solido di 4π una potenza elettromagnetica *P* pari a:

$$P = \frac{2q^2 c\left(\frac{\nu}{c}\right)^4 \left(\frac{E}{mc^2}\right)^4}{3R^2} \qquad [I.2]$$

Sostituendo nella [I.2] i valori delle costanti si ottiene:

$$P = \frac{0.0885 E^4 I}{R} \qquad [I.3]$$

dove E è espressa in GeV, I è la corrente (in mA) dovuta dagli elettroni che ruotano lungo l'orbita di raggio R (espresso in m). La radiazione elettromagnetica fuoriesce formando un fascio tangente all'orbita con un angolo di apertura $\psi \approx mc^2/E$ (dell'ordine di decimi di mrad per energie dell'ordine dei GeV) e viene convogliata in apposite guide. Naturalmente, l'orbita della particella può essere mantenuta solamente se l'energia persa sotto forma elettromagnetica è costantemente bilanciata, solitamente tramite l'applicazione di un campo magnetico (fig. 1.10).
La distribuzione spettrale della radiazione di sincrotrone si estende dall'infrarosso alla regione dei raggi X. Un parametro importante che caratterizza la distribuzione spettrale è la lunghezza d'onda critica λ_c, che divide lo spettro in due parti di eguale potenza (cioè la potenza irraggiata per $\lambda > \lambda_c$ è uguale a quella irraggiata per $\lambda < \lambda_c$) :

$$\lambda_c = \frac{4}{3}\pi R\left(\frac{E}{mc^2}\right)^3 \qquad [I.4]$$

La radiazione prodotta dalla luce di sincrotrone possiede caratteristiche uniche rispetto a ciò che si può ottenere con altri tipi di sorgente: un'elevata intensità, un intervallo spettrale continuo molto ampio, dimensioni trasversali del fascio limitate e una struttura temporale ad impulsi molto regolari.

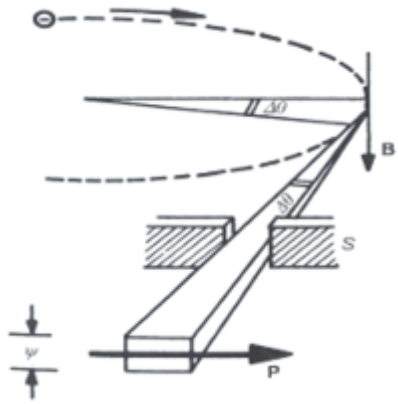


Fig. 1.10 *Radiazione di sincrotrone*

1.5.4 Sorgenti plasma-laser

La lunghezza d'onda del picco di emissione di un corpo nero è data dalla *legge di Wien*:

$$\lambda_{peak} = \frac{A}{T} \quad \text{con} \quad A = 2898\ [\mu m \cdot K] \qquad [I.5]$$

Dalla [I.5] si può dedurre che la radiazione viene emessa nella regione dei raggi X per temperature comprese tra 10^6 K e 10^7 K: a queste temperature la materia è nello stato di plasma *caldo*, consistente in un gas completamente ionizzato e in equilibrio termico (ioni ed elettroni di eguale energia). In base a queste osservazioni teoriche si sono sviluppate le sorgenti di raggi X plasma-laser, nelle quali un fascio laser a infrarossi ad alta intensità (superiore a 10^{12} W/mm^2) viene inviato su un *target* tipicamente metallico. Il laser è caratterizzato da una sequenza di due impulsi: il primo impulso vaporizza la superficie del *target* creando un plasma debolmente ionizzato, mentre il secondo riscalda il plasma portandolo ad un a temperatura estremamente elevata e producendo la ionizzazione desiderata. Il laser pompa energia negli ioni prodotti, portandoli ad uno stato eccitato fino a raggiungere l'inversione di popolazione: il diseccitamento degli ioni ad alta energia è accompagnato dall'emissione di fotoni X. (fig. 1.11a)
Lo spettro di emissione (fig. 1.11b) contiene gruppi di righe situate poco al di sotto dell'energia di assorbimento caratteristica del livello K del materiale, e l'intensità cresce con l'aumento del numero atomico della specie che costituisce il *target*.

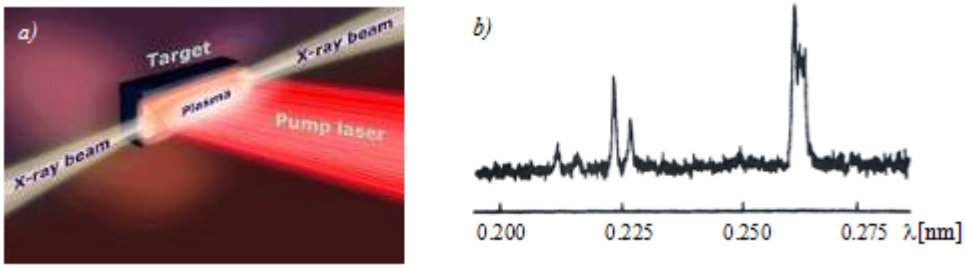


Fig. 1.11: *Emissione di raggi X da un plasma realizzato per bombardamento laser di un target di titanio: a) schema di base e b) spettro di emissione*

1.6 Rivelatori di radiazioni UV

I rivelatori per radiazioni UV attualmente più utilizzati in ambito industriale e di laboratorio si possono suddividere in due grandi categorie:

- Rivelatori fotoemissivi: sono i primi ad essere stati introdotti, in quanto sono tubi a vuoto, e dunque non richiedono l'utilizzo di semiconduttori.

- Rivelatori in silicio "*UV-enhanced*": sono rivelatori in silicio nei quali, con particolari accorgimenti tecnologici, si riesce ad ottenere una buona efficienza di risposta anche nel *range* UV. Sono utilizzati soprattutto per la profilometria (analisi della forma e dell'intensità) di fasci UV, come ad esempio i laser a eccimeri.

1.6.1 Rivelatori fotoemissivi

I fotoni UV sono sufficientemente energetici da stimolare l'emissione di elettroni da un materiale: questo effetto è efficacemente usato per la realizzazione di rivelatori fotoemissivi, costituiti da un tubo a vuoto nel quale una tensione molto elevata viene applicata fra due elettrodi. Un fotone UV, colpendo l'elettrodo polarizzato negativamente, causa l'emissione di un elettrone, che viene accelerato verso l'elettrodo positivo: in questo modo si può misurare esternamente una corrente. Essendoci il vuoto fra i due elettrodi, la corrente di buio è virtualmente nulla, e dunque i rivelatori fotoemissivi possono raggiungere elevate sensibilità: inoltre, scegliendo adeguatamente il materiale del fotocatodo, è possibile realizzare rivelatori sensibili alla sola radiazione UV.
Gli svantaggi principali dei rivelatori fotoemissivi sono: 1) l'alta tensione di polarizzazione necessaria per un corretto funzionamento dei dispositivi, cosa che li rende molto fragili; 2) il notevole ingombro, che non permette la realizzazione di rivelatori a pixel per la cattura di un'immagine della radiazione UV.

1.6.2 Rivelatori in silicio "*UV-enhanced*"

La diffusione e lo sviluppo della tecnologia del silicio rende conveniente l'utilizzo di questo materiale per la maggior parte delle applicazioni: questo non significa solo quelle applicazioni per le quali è naturalmente adatto, ma anche quelle in cui in teoria sarebbero più adatti materiali alternativi.
Nel caso della rivelazione di radiazioni ottiche, la *gap* del silicio (1.12 eV) lo rende ideale per la realizzazione di fotodiodi adatti alla rivelazione di radiazioni nell'intervallo VIS-IR, in particolare di lunghezze d'onda minori di 1.1 μm (valore che corrisponde alla *gap*). A lunghezze d'onda inferiori a 350 nm, però, il coefficiente di assorbimento del silicio è molto elevato, e la radiazione è completamente assorbita nei primi 100 nm di spessore a partire dalla superficie esposta: questo vuol dire che i portatori sono generati al di fuori della regione di svuotamento (la regione attiva del fotodiodo), dove l'elevato campo elettrico assicura un'alta efficienza di raccolta; inoltre, la superficie è sempre caratterizzata da un'alta densità di difetti, che riducono il tempo di vita medio dei portatori fotogenerati. Per questi motivi, in pratica, i fotodiodi in silicio hanno normalmente un limite inferiore nella fotorisposta intorno ai 320 nm.
Lo sviluppo di rivelatori UV è ancora oggi diretto verso l'ottimizzazione di rivelatori già esistenti in silicio, piuttosto che verso l'utilizzo di nuovi materiali: realizzando una giunzione estremamente superficiale e usando strati antiriflesso, è possibile ottenere rivelatori con efficienza quantica superiore al 60% fino a 200 nm. Nella fig. 1.12 sono

messe a confronto le risposte spettrali di due fotodiodi in silicio: uno standard (Hamamatsu S1227-BR) e uno "*UV-enhanced*" (Hamamatsu S1227-BQ) [6].

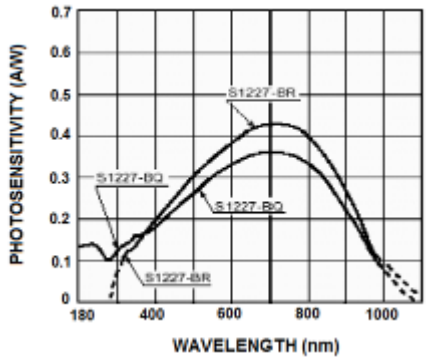


Fig. 1.12 *Confronto fra un fotodiodo standard e uno "UV-enhanced"*

Un discorso analogo è valido per i rivelatori a pixel in silicio utilizzati per la profilometria UV: in questo caso, i rivelatori commerciali migliori sono realizzati a matrice CCD [7], e quando vengono utilizzati nel lontano UV presentano gli stessi problemi dei fotodiodi in silicio a singolo pixel. Le uniche differenze sono che le matrici CCD hanno bisogno di un materiale di altissima qualità per funzionare correttamente, e inoltre le prestazioni sono molto più influenzate dai difetti superficiali rispetto ai fotodiodi; infine, se non si interviene con uno strato antiriflesso, la struttura CCD ha un'efficienza quantica praticamente nulla al di sotto dei 400 nm (vedi fig. 1.13), e anche dopo la creazione dello strato antiriflesso l'efficienza quantica non supera mai il 10%.

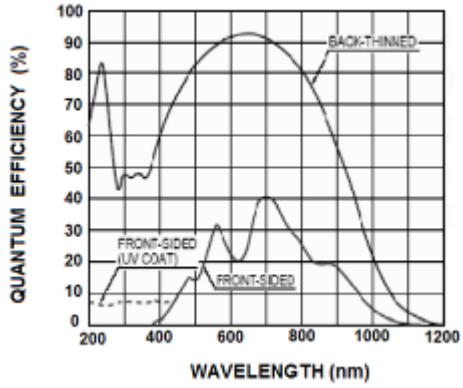


Fig. 1.13 *Efficienza quantica di matrici CCD in silicio*

La soluzione a problemi di questo tipo è stata quella di illuminare la matrice CCD dal retro: ovviamente, per assicurare l'assorbimento di fotoni all'interno della zona attiva, il substrato di silicio dev'essere inevitabilmente molto ridotto in spessore (10 ÷ 20 μm). In

questo modo si riescono ad ottenere prestazioni molto buone, con efficienza quantica sempre superiore al 50% nel *range* 200 ÷ 300 nm (vedi fig. 1.13).
La tecnologia utilizzata per arrivare a prestazioni di questo tipo incrementa i costi di produzione: tuttavia, i risultati sono buoni abbastanza da ritenere non economicamente favorevole investire nello sviluppo di materiali alternativi.

1.6.3 Il diamante come alternativa al silicio "*UV-enhanced*"

Lo studio di nuovi semiconduttori come materiali attivi per un dispositivo è giustificato solo se permette di superare alcune limitazioni relative all'utilizzo del silicio. Analizzando le proprietà dei rivelatori in silicio nelle applicazioni che comprendono l'analisi di radiazioni UV, si possono individuare due problemi.
Il primo problema è che, come si può vedere nelle figure 1.12 e 1.13, i fotorivelatori "*UV-enhanced*", sebbene abbiano una buona risposta nel lontano UV, mantengono comunque un'ampia risposta spettrale nel visibile e nel vicino IR, con efficienze quantiche che superano facilmente l'80%: questo può essere un fattore limitante in tutte quelle applicazioni dove la radiazione UV da rivelare è sovrapposta ad un forte *background* di radiazioni nel visibile e nel vicino IR, come ad esempio in tutte le applicazioni "*solar blind*". Com'è stato mostrato in precedenza, questo problema viene risolto interponendo un filtro che blocca tutte le radiazioni indesiderate: la presenza di un filtro, come di un qualsiasi altro oggetto fra sorgente e rivelatore (ad esempio un attenuatore nel caso di sorgenti ad elevata intensità) può però rappresentare a sua volta un problema, specialmente nell'analisi della forma e dell'intensità di un fascio UV, in quanto può introdurre distorsioni.
Il secondo limite dei rivelatori in silicio "*UV-enhanced*" si incontra nell'analisi di sorgenti UV impulsate, come i laser a eccimeri utilizzati in fotolitografia. Sorgenti di questo tipo producono impulsi della durata di pochi ns: a causa dell'elevato coefficiente di assorbimento del silicio alle lunghezze d'onda dei laser d'interesse (ArF* a 193 nm, F_2* a 157 nm), la massima densità di energia che può essere inviata sul rivelatore prima di indurvi un danneggiamento da radiazione è piuttosto bassa: inoltre la vita operativa di matrici CCD in silicio "*UV-enhanced*", in caso di utilizzo con laser a eccimeri, è limitata mediamente a circa 10^5 impulsi [8]. Il problema è ancora più grave se si pensa che questa è una tipica applicazione industriale, in cui dunque è di cruciale importanza l'affidabilità del rivelatore. Per evitare un grave danneggiamento del rivelatore, è assolutamente necessario l'utilizzo di attenuatori e/o fare utilizzo di un cristallo fluorescente (ad esempio vetro drogato con ioni di cerio Ce^{3+}) in grado di convertire la radiazione UV in luce visibile [9]: in ogni caso, si ha comunque una perdita di risoluzione nell'analisi spaziale, soprattutto in caso di fasci laser con un piccolo *spot-size*. Tutto questo porta alla conclusione che l'attenuazione nei profilometri in silicio per laser a eccimeri è un problema serio, e risolvibile soltanto scendendo a compromessi in termini di prestazioni e, questione da non sottovalutare, costi.
Viste le limitazioni del silicio nella rivelazione di radiazioni UV, e individuate le applicazioni in cui queste limitazioni si fanno più sentire, è doveroso valutare la possibilità di trovare materiali alternativi. Per la diagnostica di fasci laser nel lontano UV (come i laser a eccimeri) e per tutte le applicazioni *solar blind* in cui sono impiegate sorgenti UV ad elevata intensità, il diamante può essere visto come la scelta migliore: la

sua *gap* (5.5 eV) ben si adatta alle radiazioni utilizzate nella fotolitografia di circuiti integrati di ultima generazione (dispositivi ULSI con *feature sizes* inferiori a 100 nm), ovvero i laser a eccimeri ArF* (energia 6.4 eV) e F_2* (energia 7.9 eV). Inoltre, la sua alta energia di coesione e l'elevatissima conducibilità termica permettono al diamante di resistere ad alte intensità di radiazione senza subire praticamente danni.
Il profilometro UV in diamante realizzato nell'ambito di quest'attività di ricerca si è dimostrato superare le limitazioni del silicio "*UV-enhanced*": le caratteristiche del diamante permettono di avere un'alta efficienza quantica nel solo intervallo d'interesse, e di evitare ogni tipo di attenuazione o conversione in lunghezza d'onda del fascio da analizzare senza avere problemi di danno da radiazione.

1.7 Rivelatori di raggi X

I rivelatori di raggi X attualmente in uso si possono suddividere in tre grandi categorie:

- Rivelatori a gas : sfruttano l'azione ionizzante dei raggi X in alcuni tipi di gas.
- Rivelatori a scintillazione: si basano sul conteggio di fotoni generati dalla luminescenza indotta dai raggi X su cristalli scintillatori.
- Rivelatori a semiconduttore: sfruttano l'aumento di conducibilità di materiali semiconduttori o semiisolanti in seguito all'irraggiamento X .

1.7.1 Rivelatori a gas

Questi dispositivi sono stati tra i primi ad essere utilizzati per la rivelazione di radiazioni ionizzanti. Sfruttano la ionizzazione prodotta dal passaggio di un fotone o di una particella carica in un gas, processo nel quale un elettrone viene rimosso da un atomo o da una molecola in modo da creare una coppia elettrone-ione: il numero medio di coppie create è dunque proporzionale all'energia depositata nel dispositivo. Per aumentare l'efficienza di generazione di coppie, è opportuno utilizzare gas facilmente ionizzabili: per questo si impiegano gas nobili (es. Xe, Ar, Kr), nei quali l'energia media necessaria alla produzione di una coppia è stimata nell'intervallo di 30 ÷ 40 eV.

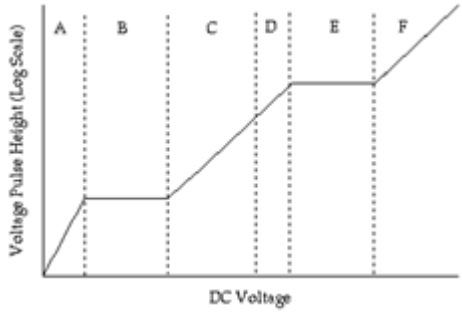


Fig. 1.14 *Curva segnale d'uscita-potenziale di un rivelatore a gas*

I rivelatori a gas sono costituiti da un tubo metallico messo a terra (che fa da catodo) riempito di gas nobile, e avente come anodo un filo isolato al centro: sotto l'azione del campo elettrico, gli elettroni vengono accelerati verso l'anodo e gli ioni verso il catodo, dando vita ad un segnale d'uscita che dipende fortemente dal potenziale applicato (fig. 1.14).
Nella regione A non tutte le cariche prodotte vengono raccolte, in quanto a causa del piccolo valore del campo elettrico il processo di ricombinazione delle varie coppie ione-elettrone è notevole. Aumentando però la differenza di potenziale applicata, il tempo a disposizione per la ricombinazione diminuisce, perché aumenta la componente della velocità delle coppie lungo la direzione del campo, e questo migliora l'efficienza di raccolta delle cariche.
Nella regione B, detta di *saturazione*, gli effetti della ricombinazione diventano trascurabili, e la carica raccolta è tutta quella prodotta: in questa regione lavorano le *camere a ionizzazione*.
Nelle regioni C e D il campo elettrico è sufficientemente intenso da far acquistare agli elettroni primari (prodotti cioè dalla radiazione X) energia cinetica sufficiente a ionizzare gli atomi del gas, dando vita in questo modo ad una moltiplicazione a valanga di ioni: la ionizzazione secondaria è strettamente dipendente da quella primaria ed è in queste regioni che lavorano i *contatori proporzionali*.
Nella regione E, detta di *Geiger-Müller* (in questa regione lavorano i contatori Geiger), la carica raccolta non è più proporzionale alla ionizzazione primaria: oltre alla ionizzazione, si hanno infatti altri fenomeni quali l'eccitazione seguita da emissione di luce visibile e ultravioletta, e questo produce un impulso costante in un certo intervallo del potenziale applicato, indipendentemente dal tipo di radiazione utilizzata.
Nella regione F non è più possibile nessun tipo di rivelazione: l'impulso in uscita non dipende più dalla radiazione incidente, poiché avviene una scarica in presenza o meno di radiazione.

1.7.2 Rivelatori a scintillazione

I rivelatori a scintillazione sono formati (fig. 1.15) da un cristallo scintillante (*scintillatore*) accoppiato otticamente ad un *tubo fotomoltiplicatore* (o a un fotodiodo a semiconduttore).

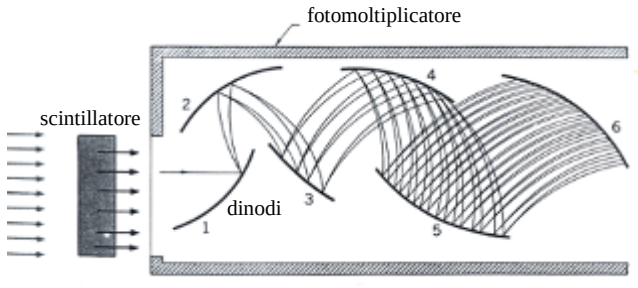


Fig. 1.15: *Schema di un rivelatore a scintillazione composto dallo scintillatore e dal fotomoltiplicatore*

Il cristallo scintillante assorbe energia dalla radiazione X in ingresso, fatta passare attraverso una finestra di berillio, e la riemette sotto forma di fotoni nel visibile (fenomeno della *luminescenza*): il materiale più utilizzato per gli scintillatori è lo *ioduro di sodio* (NaI) [10], contenente piccole quantità di tallio (1 ÷ 3%) in soluzione solida. Il cristallo è generalmente di forma cilindrica, con una delle basi rivolta verso il catodo del fotomoltiplicatore (in alternativa si può collegare il cristallo al fotomoltiplicatore attraverso una fibra ottica a riflessione interna); è sigillato contro l'umidità esterna e protetto dalla luce con un foglio d'alluminio, che serve anche da riflettore interno. La parte posteriore del contenitore del cristallo è realizzata in vetro, per permettere il passaggio della radiazione visibile; inoltre, per ridurre fenomeni di riflessione, viene interposto fra il cristallo ed il fotomoltiplicatore un fluido con indice di rifrazione paragonabile a quello del vetro. I fotoni prodotti dallo scintillatore vengono raccolti dal tubo fotomoltiplicatore: il *fotocatodo*, costituito da una sostanza fotoemissiva depositata in sottilissimo strato sulla parete interna della finestra d'ingresso del fotomoltiplicatore, converte la luce incidente in elettroni sfruttando l'effetto fotoelettrico; subito dopo il fotocatodo c'è il *moltiplicatore di elettroni*, che ha la funzione di amplificare la debole fotocorrente proveniente dal fotocatodo, in modo tale da far arrivare all'anodo una corrente misurabile. Il moltiplicatore di elettroni è costituito da una serie di elettrodi (*dinodi*) mantenuti a diverso potenziale in modo da accelerare e guidare gli elettroni verso l'anodo: si ottiene così in uscita un segnale elettrico proporzionale al numero di fotoni in ingresso al tubo, a sua volta proporzionale all'energia della radiazione X incidente sullo scintillatore.

1.7.3 Rivelatori a semiconduttore

I rivelatori per raggi X a semiconduttore sono essenzialmente delle camere a ionizzazione in cui il gas è sostituito con un mezzo solido di bassa conducibilità. Se una radiazione ionizzante attraversa questo mezzo, essa crea delle coppie elettrone-lacuna, che vengono separate dal campo elettrico e raccolte sugli elettrodi (vedi fig. 1.16).
In un rivelatore a gas occorrono circa 30 eV per creare una coppia elettrone-ione positivo, mentre in un rivelatore a semiconduttore l'energia media per la produzione di una coppia elettrone-lacuna (vedi par. 2.8) è circa $3E_{gap}$ (si va dai 3.6 eV nel silicio ai 13 eV per il diamante): quindi, a parità di altre condizioni, il segnale di uscita può essere anche 10 volte maggiore.
Altra differenza rispetto ai rivelatori a gas è che mentre in questi la mobilità degli elettroni è molto più alta di quella degli ioni positivi, nei rivelatori a semiconduttore le differenze di mobilità si attenuano: nel diamante, ad esempio, la mobilità delle lacune (1600 $cm^2 \cdot V^{-1} \cdot s^{-1}$) è di poco inferiore a quella degli elettroni (2200 $cm^2 \cdot V^{-1} \cdot s^{-1}$), e dunque i contributi di segnale sono praticamente gli stessi per entrambi i tipi di portatori.
Il semiconduttore ideale per un rivelatore di radiazione dovrebbe avere alta resistività, per far sì che il numero di coppie create per ionizzazione sia maggiore del numero di cariche libere presenti nella regione sensibile, in modo tale che l'impulso raccolto dagli elettrodi sia di ampiezza superiore al rumore di fondo; per risolvere questo problema si possono seguire due strade:

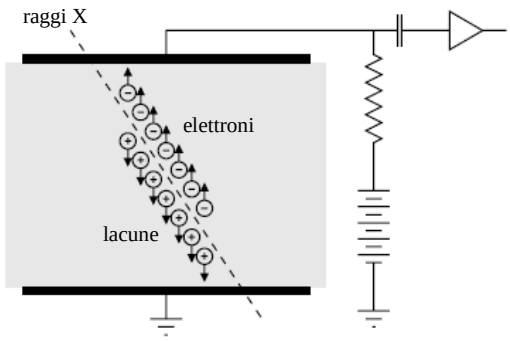


Fig. 1.16 *Schema di principio di un rivelatore a semiconduttore*

1) Sfruttare una *giunzione p-n*: nella regione di carica spaziale non vi sono cariche libere di alcun segno, per cui in tale regione aumenta localmente la resistività del semiconduttore. I rivelatori in silicio e germanio, due semiconduttori a *gap* intermedia (rispettivamente 1.12 e 0.66 eV), sono tutti a giunzione e consentono di soddisfare contemporaneamente l'esigenza di avere una buona efficienza di generazione di carica (grazie alla *gap* non troppo elevata) e l'esigenza di un basso rumore di fondo (grazie alla presenza della giunzione *p-n*). Volendo allargare la regione di svuotamento, che costituisce il volume sensibile di questo tipo di rivelatori, si può polarizzare inversamente la giunzione.

2) Utilizzare *semiconduttori ad elevata gap*, come il diamante (5.5 eV) o il solfuro di cadmio (2.42 eV): il vantaggio è ovviamente quello di non avere la necessità di creare giunzioni, visto che l'elevata *gap* conferisce di per sé un'alta resistività al materiale; al contempo, però, una *gap* elevata si traduce in una maggiore energia per la creazione di una coppia elettrone-lacuna, e questo impedisce di avere ampiezze di segnale comparabili con quelle che si ottengono con i rivelatori a giunzione.

Comunque venga realizzato, un rivelatore a semiconduttore per raggi X dovrebbe avere:

- un'alta risoluzione spaziale e un'ampia regione attiva.
- un'eccellente risoluzione temporale e l'abilità di sostenere elevati *rate* di conteggio.
- una buona risoluzione in energia, unita ad un'efficienza quantica prossima al 100% su un ampio *range* di energie.
- un'uscita stabile nel tempo e una corrente di buio trascurabile.
- un'elevata resistenza al danno di radiazione.

Inoltre dovrebbe essere semplice, resistente, facile da realizzare, economico, leggero ed avere un basso consumo di potenza.
Un rivelatore per raggi X che soddisfi tutte queste specifiche non esiste. Per questo motivo esiste una così grande varietà di rivelatori a seconda del tipo di applicazione

(raggi X *hard* o *soft*, alta o bassa intensità, etc.). Il rivelatore in diamante policristallino realizzato nell'ambito di quest'attività di ricerca è stato utilizzato sia per misurare la corrente fotoindotta da un tubo a raggi X di elevata intensità (ottenendo anche un'analisi della forma del fascio) che per rivelare il singolo fotone (ottenendo una stima dell'energia della radiazione): ha mostrato dunque una buona versatilità, ed evidenziato le potenzialità dei rivelatori in diamante in numerose applicazioni che fanno uso dei raggi X.

Capitolo II

Il diamante: proprietà, sintesi e fotoconducibilità

2.1 Struttura del carbonio

Le due forme allotropiche più conosciute del carbonio nello stato cristallizzato sono il *diamante* e la *grafite* [11]: nonostante siano composti dallo stesso materiale, essi presentano proprietà chimiche, fisiche ed elettroniche completamente differenti.
La struttura elettronica esterna del carbonio nello stato fondamentale è $2s^2 2p^2$: questo implica la possibilità di realizzare solo due legami covalenti con altrettanti atomi. Tale limitazione può essere superata attraverso l'ibridizzazione di parte degli orbitali esterni: in base a quanti di questi orbitali prendono parte all'ibridizzazione, il numero dei legami disponibili cambia, così come la struttura cristallina e le sue proprietà. Il più alto numero di legami disponibili si ottiene nel caso di un'ibridizzazione sp^3. In questo caso si ottengono quattro orbitali isoenergetici (l'orbitale *s* e i tre orbitali *p*), con la possibilità di formare quattro legami covalenti: questo è il caso del diamante, in cui ogni atomo di carbonio è legato ad altri quattro atomi. La struttura di base può essere vista come un tetraedro in cui gli atomi sono posti al centro e ai quattro vertici. Un altro approccio per rappresentare questa struttura consiste nell'uso di una forma cubica come base: in tal caso la struttura del diamante può essere vista come la composizione di due reticoli cubici facce-centrati traslati l'uno rispetto all'altro, nella direzione della diagonale, di ¼ della lunghezza della diagonale stessa.

Fig. 2.1 *Struttura reticolare del diamante (a) e della grafite (b)*

Nel caso della grafite si ha un'ibridizzazione sp^2: i legami covalenti di tipo σ disponibili in questo caso sono tre, mentre l'orbitale *p* residuo (non coinvolto nell'ibridizzazione), forma un legame di tipo π. La struttura reticolare che ne deriva è data dalla sovrapposizione di piani cristallini la cui cella di base forma un esagono. Il legame di tipo π è delocalizzato: questa delocalizzazione, che permette ad un elettrone di muoversi liberamente su un piano cristallino, ma non da piano a piano, dà luogo a una conducibilità elettrica fortemente anisotropa, che caratterizza la grafite. I piani cristallini

sono legati fra loro attraverso interazioni di Van der Waals: l'energia caratteristica relativamente bassa di quest'interazione permette ai diversi piani cristallini di scivolare uno sull'altro, conferendo alla grafite le sue note proprietà meccaniche.

2.2 Proprietà fisiche ed elettroniche del diamante

Il diamante è uno dei materiali più interessanti per un gran numero di applicazioni industriali. Fra le più importanti proprietà fisiche, è importante menzionare l'elevata durezza meccanica (~ 90 GPa), la bassa comprimibilità ($8.3 \cdot 10^{-13}$ m^2/N), il basso coefficiente di dilatazione termica ($0.8 \cdot 10^{-6}$ K) e la più alta conducibilità termica esistente in natura ($2 \cdot 10^3$ W/m·K). E' inoltre chimicamente inerte, resistente ad ambienti ostili, elettricamente isolante e trasparente alle radiazioni ottiche dal vicino UV all'infrarosso. Alcune delle applicazioni derivate da queste proprietà sono ben collaudate, come il suo utilizzo in strumenti di taglio e abrasione; altre stanno cominciando ad uscire fuori dall'ambito della ricerca imponendosi sul mercato, come l'utilizzo del diamante come dissipatore di calore nei circuiti integrati.
Da un punto di vista elettronico, il diamante è considerato un semiconduttore a larga *gap* (5.47 eV a temperatura ambiente) [12]. L'utilizzo dei semiconduttori a larga *gap* per applicazioni elettroniche permette di ottenere numerosi vantaggi, i più importanti dei quali sono:

- Disponibilità di substrati ad alta resistività, che permettono una forte riduzione delle correnti di buio e delle capacità parassite.
- Possibilità di lavorare ad alte temperature.
- Possibilità di lavorare ad alti campi elettrici, e dunque ad alta potenza.
- Interazione con le radiazioni ottiche solo nella regione UV.

Inoltre, sia la mobilità dei portatori che la loro velocità di saturazione sono, nella maggior parte dei casi, più elevate rispetto ad altri semiconduttori, e questa è una caratteristica molto importante per la velocità di un dispositivo.
Nella tabella 2.1 sono riassunte le principali proprietà del diamante, messe a confronto con quelle dei semiconduttori più utilizzati in ambito elettronico e optoelettronico.
Da questa tabella emerge come il diamante, rispetto al Si e ad altri semiconduttori utilizzati nella fotorivelazione, presenti numerosi vantaggi, fra i quali vanno menzionati:

- Elevato campo di *breakdown* e grande energia di coesione (legami C – C di tipo σ) che assicurano grande resistenza al danno da radiazione.
- Maggiore *gap* (5.5 eV a 300 K), con possibilità dunque di resistere a temperature operative più alte e assicurare bassissime correnti di buio (il materiale è inoltre un filtro naturale nei confronti delle radiazioni ottiche).
- Una risposta più veloce dovuta ad una maggiore velocità di saturazione dei portatori (doppia rispetto a quella del Si) e ad una maggiore mobilità dei portatori.

- La costante dielettrica (5.7) è meno della metà di quella del Si (11.9), e questo garantisce una capacità più bassa, dunque un rumore intrinsecamente più basso nell'analisi del segnale.
- La bassa reattività implica una consistente refrattarietà alla corrosione chimica.
- Il numero atomico del carbonio è prossimo a quello dell'acqua: il diamante è dunque ideale per applicazioni dosimetriche, in quanto ha un comportamento tessuto - equivalente.

	Diamond	β-SiC	Si	GaAs
Energy bandgap [eV]	5.48	3.0	1.12	1.43
Lattice constant [nm]	0.3567	0.4358	0.5430	0.5650
Density [$g\ cm^{-3}$]	3.515	3.216	2.328	5.316
Melting point [$^{\circ}C$]	4000	2540	1420	1238
Saturation velocity [$10^7\ cm\ s^{-1}$]	2.7	2.5	1.0	1.0
Electrons mobility [$cm^2\ V^{-1}\ s^{-1}$]	2200	400	1500	8500
Holes mobility [$cm^2\ V^{-1}\ s^{-1}$]	1600	50	600	400
Breakdown electric field [$10^7\ V/cm$]	10	4	0.3	6
Relative dielectric constant	5.5	9.7	11.8	12.5
Resistivity [$\Omega\ cm$]	10^{13}	10^{10}	10^3	10^8
Thermal conductivity [$W\ cm^{-1}\ K^{-1}$]	20	5	1.5	0.46

Tab. 2.1 *Proprietà del diamante e dei principali semiconduttori per fotorivelazione*

2.3 Tecniche di crescita del diamante

L'utilizzo del diamante in campi diversi dalla gioielleria era relegato, fino a pochi anni fa, ad applicazioni industriali come strumenti per taglio e abrasione: questo è dovuto al fatto che queste applicazioni sono le uniche in cui si possono efficacemente usare piccoli (e relativamente poco costosi) diamanti naturali. Ogni altra applicazione che richiedesse cristalli di maggiori dimensioni, naturali o sintetici che fossero, era dunque preclusa per questione soprattutto di costi.
Ciò che rende difficoltosa e costosa la produzione di diamante sintetico non è da trovarsi nella rarità della materia prima, essendo il carbonio uno dei più abbondanti elementi sulla Terra, ma piuttosto nelle condizioni necessarie per la formazione del diamante. La fig. 2.2 mostra il diagramma di stato del carbonio [13], dal quale si evince che in condizioni standard (20 °C e 1 atm) il carbonio cristallizza in forma di grafite. Per ottenere diamante, è necessario creare condizioni di pressione e temperatura molto elevate (dell'ordine di 10^{10} Pa e migliaia di K rispettivamente): sono appunto queste le condizioni che si verificano nella profondità della crosta terrestre, dove si forma il diamante naturale. E' necessario notare che il diamante, anche se posto in condizioni ambientali di pressione e temperatura, non si trasforma in grafite, perché l'energia di attivazione del processo di trasformazione è molto più elevata dell'energia termica: il diamante è dunque una forma metastabile di carbonio.

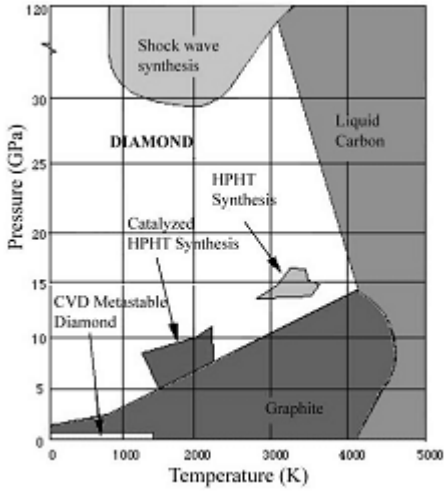


Fig. 2.2 *Diagramma di stato del carbonio e tecniche di sintesi del diamante*

Il primo processo sviluppato per crescere diamante sintetico è stato la tecnica HPHT (*High Pressure High Temperature*), il cui obiettivo è appunto quello di riprodurre le condizioni di formazione del diamante naturale. Tuttavia, gli elevati costi di produzione rendono la tecnica HPHT conveniente solo per la realizzazione di polvere di diamante usata in applicazioni industriali.

L'industria elettronica non ha mai fatto largo uso della tecnica HPHT, non solo per ragioni di costi, ma anche per l'intrinseca incompatibilità con la tecnologia planare dei semiconduttori: per questo motivo negli ultimi anni sono stati compiuti numerosi sforzi per adattare la deposizione chimica da fase vapore (CVD), ben collaudata nell'industria del silicio, alla crescita di film in diamante. La tecnica CVD è basata su una reazione chimica che avviene sulla superficie del substrato [14], e che causa la deposizione di un film di materiale su di esso. Fondamentale è la sorgente dell'energia necessaria per la dissociazione delle molecole dei gas precursori, processo in seguito al quale gli atomi sono liberi di formare legami sulla superficie di crescita: l'energia può essere termica, e a tale scopo si utilizza un filamento incandescente (HFCVD), oppure può essere fornita attraverso un plasma a microonde (MWCVD). Le due tecniche condividono la maggior parte dei parametri di processo, come la temperatura del substrato (1000 ÷ 1400 K) e il gas precursore utilizzato, che tipicamente è CH_4 (1 ÷ 2% vol.) diluito in idrogeno.

I film ottenuti con le tecniche CVD sono policristallini, e la loro qualità e morfologia dipendono fortemente dalle condizioni di crescita, in particolare dal *rate* di crescita, parametro che può essere regolato variando la percentuale di metano nel gas precursore: ad ogni modo, una qualità sufficiente può essere ottenuta solo usando una percentuale molto bassa di CH_4, che conduce a velocità di crescita intorno a 1 μm/h.

Una prima valutazione della qualità del film cresciuto può essere fatta valutando il rapporto esistente fra legami sp^3 (diamante) e legami sp^2 (grafite), oppure valutando la composizione del film (ad esempio il contenuto di azoto o idrogeno). Essendo la grafite, e non il diamante, la forma termodinamicamente stabile del carbonio nelle condizioni di crescita, la possibilità di formare il diamante è assicurata dalla presenza di atomi di

idrogeno formati dall'attivazione del gas precursore. Si pensa che questi atomi siano responsabili di diversi fenomeni, fra i quali:

- La sottrazione di idrogeno al gas precursore, con la conseguente produzione di radicali liberi di carbonio estremamente reattivi.
- La terminazione della superficie del film in crescita, evitando in tal modo che gli atomi superficiali di carbonio si leghino fra loro in modo diverso da quello previsto per la struttura cristallina del diamante.
- L'attacco chimico sia del diamante che della grafite: nelle tipiche condizioni di crescita, però, la grafite viene rimossa più velocemente, e si pensa che proprio questo sia alla base della deposizione preferenziale di diamante piuttosto che di grafite.

Il substrato maggiormente utilizzato per la crescita di film di diamante è il silicio monocristallino. La scelta è dovuta non solo al basso costo del materiale, ma anche ad alcune caratteristiche che soddisfano i requisiti delle tecniche di crescita:

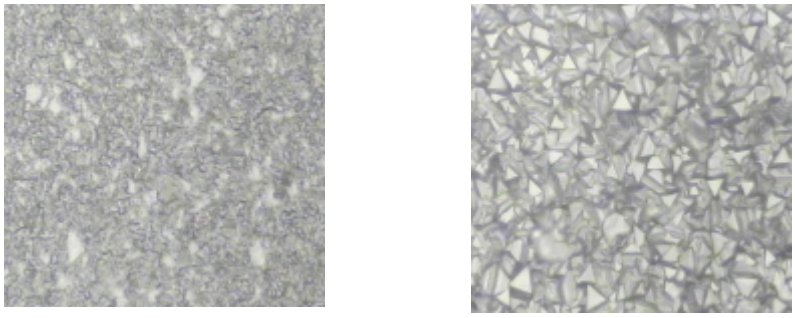
- Punto di fusione più elevato della temperatura di crescita (1000 ÷ 1400 K).
- Possibilità di formare uno strato di carburo all'interfaccia per favorire il processo di nucleazione e migliorare l'adesività del film in crescita.
- Coefficiente di dilatazione termica molto vicino a quello del diamante.

Negli ultimi anni, sono stati compiuti numerosi studi per stabilire quale substrato fosse il migliore per la crescita di film di diamante: l'obiettivo finale è trovare un materiale che abbia un basso *mismatch* reticolare con il diamante, consentendo dunque di avere a disposizione substrati di grandi dimensioni (e relativamente economici) per la crescita di diamante monocristallino. Al momento, i risultati migliori si sono ottenuti utilizzando substrati di diamante HPHT: substrati di questo tipo, sebbene contenenti elevate densità di difetti e impurità, costituiscono una scelta perfetta per ottenere film di diamante omoepitassiali. Proprio su film di diamante omoepitassiale cresciuto su substrato HPHT sono state misurate le più alte mobilità dei portatori di carica, rispettivamente 4500 e 3800 $cm^2/V \cdot s$ per elettroni e lacune: questi valori sono più alti persino di quelli generalmente riportati per un diamante naturale di tipo IIa (elevato grado di purezza).

2.4 Morfologia e struttura del diamante policristallino

Il diamante CVD si presenta come una struttura policristallina i cui grani, delle dimensioni medie di 70 ÷ 100 nm, hanno una forma colonnare (crescita preferenziale nella direzione <110>). I grani sono tra loro separati dai caratteristici *bordi di grano*, la cui composizione è varia e molto spesso comprende fasi grafitiche. Questa struttura colonnare trae origine dal fatto che la possibilità di un grano cristallino in formazione di espandersi nelle tre direzioni è impedita dall'avvicinamento di un altro grano cristallino: la crescita può dunque svilupparsi solo lungo l'asse *z*, con la generazione del bordo intergrano. Questo spiega il perché della forte differenza morfologica tra la superficie

d'interfaccia film-substrato (fig. 2.3a) e quella di crescita (fig. 2.3b): nella prima il materiale presenta una struttura quasi amorfa (caotica e molto difettata), dovuta anche al fatto che il substrato è spesso sottoposto ad un'abrasione pre-deposizione [15] (si rovina intenzionalmente la superficie aumentando il numero di legami insaturi a disposizione dei primi atomi di carbonio in arrivo); la seconda presenta invece una direzione cristallografica preferenziale ed una struttura cristallina più ordinata.



(a) **(b)**

Fig. 2.3 *Immagini della superficie di nucleazione (a) e della superficie di crescita (b)*

I parametri per stimare la qualità morfologico-strutturale del film sono tre: le dimensioni dei grani cristallini, il rapporto tra fase diamante e fase grafitica, e la direzione cristallografica prevalente.

2.4.1 Dimensione dei grani cristallini

La dimensione media dei domini cristallini è un parametro fondamentale, poiché rappresenta la tendenza del film ad avere una struttura poli- o mono- cristallina, con diminuzione della densità dei bordi di grano. La spettroscopia *Raman* costituisce un potente strumento d'indagine per stimare la dimensione dei grani cristallini e per evidenziare la presenza di eventuali fasi non-diamante.
Questa tecnica si basa sull'irraggiamento del materiale con una luce monocromatica di frequenza ν_0 e sulla misurazione della luce diffusa atomicamente di frequenza $\nu_0 \pm \Delta\nu$, dove ν_0 è determinato dallo *scattering* di *Rayleigh* (diffusione elastica), e $\Delta\nu$ è detto *shift Raman*.
Quando un fotone incidente sul materiale viene diffuso dagli atomi del bersaglio, interagisce con i modi di vibrazione del reticolo cristallino (fononi) scambiando energia. L'energia con cui il fotone emerge dal materiale sarà pertanto:

$$h\nu' = h\nu_0 \pm h\Delta\nu \qquad \text{[II.1]}$$

La differenza di energia tra la luce incidente e quella riemessa si traduce nella creazione o annichilazione di fononi: nel caso di segno positivo, si ha l'annichilazione di un fonone, viceversa se ne ha la creazione. Poiché le energie (quantizzate) dei fononi sono

caratteristiche del materiale in esame, lo spostamento in lunghezza d'onda del fotone scatterato fornisce informazioni sul materiale stesso.
La linea *Raman* caratteristica del diamante naturale (fig. 2.5a) corrisponde a $\Delta\nu = 1332.5 \pm 0.5\ cm^{-1}$ a temperatura ambiente, e la larghezza a mezza altezza del suo picco (FWHM) è tipicamente dell'ordine di $2.2\ cm^{-1}$.
I dati ricavabili da questa analisi sono molteplici:

- Lo studio della forma e della posizione del picco FWHM consente di valutare il grado di cristallinità del diamante.
- La larghezza a mezza altezza del picco è inversamente proporzionale al tempo di vita dei modi vibrazionali (fononi) all'interno della struttura, quindi righe più strette corrispondono a tempi di vita più lunghi e pertanto a grani più grandi.
- La presenza di bande spettrali a lunghezze d'onda maggiori della riga caratteristica del diamante evidenzia la presenza di fasi carbonio di differente natura.
- Uno scostamento dalla frequenza caratteristica del diamante naturale indica la presenza di eventuali stress.

In molti campioni si è potuto osservare che l'ampiezza della FWHM del picco *Raman* è una funzione approssimativamente lineare della velocità di deposizione del film: dunque velocità di deposizione più basse (unitamente a spessori maggiori), comportano dimensioni più grandi dei domini cristallini. Intuitivamente, quanto più bassa è la velocità di crescita tanto maggiore sarà il tempo a disposizione dell'atomo di carbonio per coordinarsi con la superficie del substrato, dare luogo al legame chimico e porsi nella configurazione di minima energia (cioè secondo una direzione cristallografica opportuna). In questo modo si ottiene una crescita colonnare. (fig. 2.4).

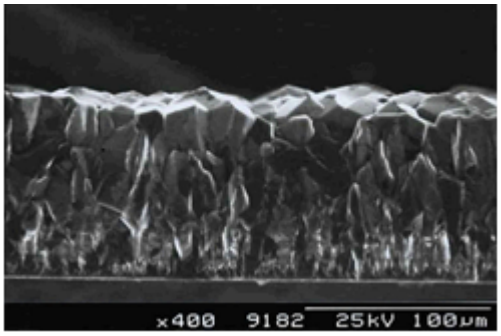


Fig. 2.4 *Crescita colonnare del diamante CVD*

Un aumento dello spessore del film (e quindi un aumento della dimensione dei grani) coincide anche con un incremento della mobilità dei portatori e del loro tempo di vita medio [16]: si è dimostrato che film molto spessi (200 ÷ 500 μm) mostrano proprietà elettroniche superiori a quelle dei diamanti puri.
Il campione utilizzato nell'ambito di questa tesi ha uno spessore di 270 μm, e in effetti (come si può vedere dalla fig. 2.5b) ha uno spettro *Raman* che si avvicina moltissimo a

quello di un diamante naturale: il picco *Raman* si presenta esattamente a 1332.5 cm^{-1}, e la FWHM del picco è di 9 cm^{-1}.

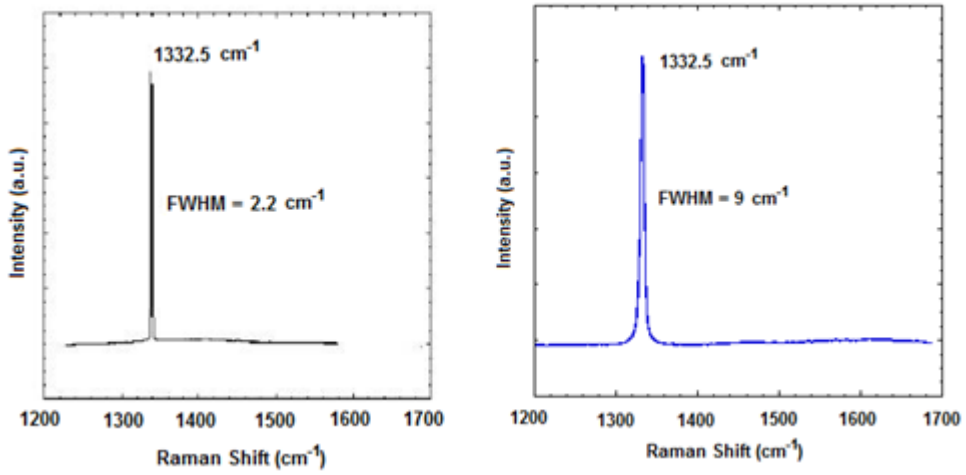


Fig. 2.5 *Spettri Raman del diamante naturale (a) e del diamante CVD utilizzato per il rivelatore (b)*

2.4.2 Rapporto tra fase diamante e fase grafitica

In generale, all'aumentare del contenuto delle fasi non-diamante nel film ci si aspetta una diminuzione della resistività (fig. 2.6), soprattutto a causa della presenza di legami sp^2 (tipici della grafite, che è praticamente un conduttore): un film di buona qualità deve dunque presentare la più bassa percentuale possibile di fase grafitica.

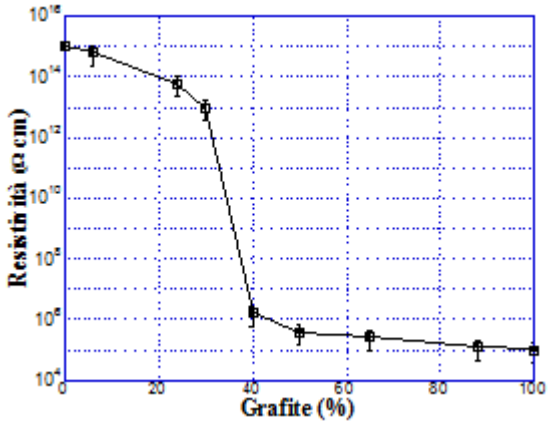
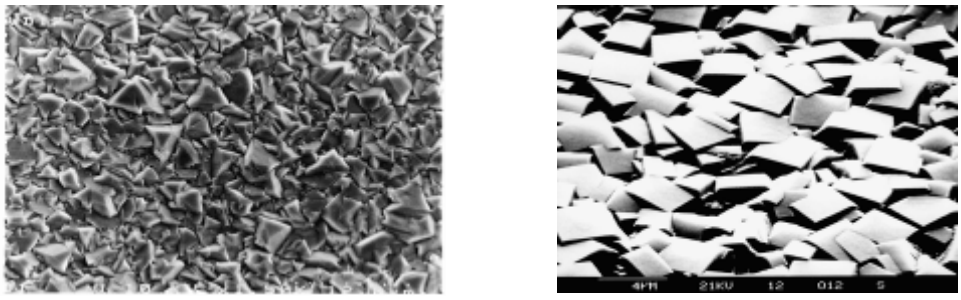


Fig. 2.6 *Andamento della resistività del diamante CVD in funzione del contenuto di grafite*

Generalmente, viene fatta l'ipotesi che il film sia costituito da tanti grani con caratteristiche isolanti (semiconduttore a larga *gap*) e che le eventuali fasi grafitiche (che introducono un sensibile incremento della conducibilità sia sulla superficie che ortogonalmente ad essa) siano concentrate sui bordi intergrano. Ad ogni modo, per mantenere le caratteristiche del campione su un buon livello di resistività, il contenuto di grafite entro il film non deve eccedere che pochi punti percentuali; in base alla spettroscopia *Raman*, sappiamo che la grafite monocristallina ha un picco attorno a 1575 cm^{-1}, mentre la grafite policristallina si colloca tra i 1350 e i 1360 cm^{-1}: integrando l'intensità del picco relativo al diamante (1332.5 cm^{-1}) e dei picchi associati alla grafite e facendone il rapporto, tenendo conto dell'efficienza di diffusione *Raman* delle due fasi carboniose (il diamante è 50 ÷100 volte meno efficiente), si può avere una valida stima della percentuale di grafite presente.

2.4.3 Direzione cristallografica del film

Terzo ed ultimo parametro è la direzione cristallografica prevalente nel film, che dipende quasi esclusivamente dalla composizione della miscela gassosa dei reagenti [17]. Quando la concentrazione di metano in idrogeno è molto bassa (minore dello 0.5%), la direzione cristallografica preferenziale è la <111>, che corrisponde alla formazione di strutture piramidali sulla superficie; all'aumentare della concentrazione di metano, fino ad un valore dell'1.5%, la direzione cristallografica preferenziale diventa la <100>, con la formazione di strutture superficiali di forma quadrata; infine per concentrazioni di metano maggiori dell'1.5%, si assiste all'innesco di un fenomeno di nucleazione secondaria molto forte, che comporta la formazione di grani sempre più piccoli ed una tendenza sempre più marcata verso la microcristallinità (fig. 2.7).



(a) **(b)**

Fig. 2.7 *Immagini SEM di superfici di film di diamante con direzione cristallografica <111> (a), e con direzione cristallografica <110> (b)*

La morfologia superficiale dipende quindi in modo critico dalla frazione di metano nella miscela gassosa: ciò significa che il completo controllo della concentrazione di CH_4 determina la qualità del film, e che piccole variazioni inattese degli altri parametri termodinamici potrebbero fortemente influenzare le condizioni di crescita. Spesso viene utilizzato uno specifico parametro per determinare la forma dei singoli cristalli, definito

come il rapporto tra il *rate* di crescita lungo la direzione <100> ed il *rate* di crescita lungo la direzione <111>:

$$\alpha = \sqrt{3}\frac{v_{100}}{v_{111}} \qquad \text{[II.2]}$$

In riferimento a un cristallo singolo, si ottiene $\alpha = 1$ per una struttura cubica, $\alpha = 3$ per una struttura ottaedrica, mentre per una struttura cubottaedrica si ha $1 < \alpha < 3$.

2.5 Drogaggio del diamante CVD

A causa della corta lunghezza di legame (0.154 nm), è molto difficile trovare atomi di grosse dimensioni in campioni di diamante, e questo crea un grosso limite per la scelta di materiali droganti.

Per ottenere un drogaggio di tipo *p* si utilizza principalmente boro, un atomo relativamente piccolo che può essere efficacemente incorporato all'interno del reticolo cristallino del diamante. Tuttavia, il livello energetico introdotto dal boro è a 0.36 eV dalla banda di valenza [18]: questo valore è circa un ordine di grandezza maggiore rispetto all'energia termica di un elettrone a temperatura ambiente, e di conseguenza solo l'1% circa dei droganti è ionizzato. Una valida alternativa al drogaggio *p* convenzionale nel diamante è rappresentata dal drogaggio superficiale, ottenuto terminando la superficie con atomi di idrogeno: questa condizione si ottiene agevolmente attraverso un trattamento in plasma di idrogeno (idrogenazione), e si può trovare ad esempio in quei film di diamante CVD che non hanno subito trattamenti di ossidazione dopo la crescita. In caso di idrogenazione, l'assorbimento di ioni H_3O^+ sulla superficie induce una conducibilità di tipo *p* limitata a pochi strati atomici, formando quel che viene definito un gas bidimensionale di lacune (2DHG). Il drogaggio p superficiale è stato efficacemente sfruttato per la realizzazione di transistor MESFET a canale superficiale, con ottime prestazioni ad alta frequenza [19] e alta potenza [20].

Più problematico è il drogaggio di tipo *n*: l'azoto, per la sua posizione nella tavola periodica, sarebbe un candidato perfetto. Inoltre, l'azoto viene incorporato abbastanza facilmente nel reticolo cristallino del diamante, e infatti rappresenta una delle impurità più presenti sia nel diamante naturale che in quello sintetico. Tuttavia, il livello energetico introdotto dall'azoto è a ben 1.7 eV dalla banda di conduzione [21], rendendo completamente inefficace il drogaggio dal punto di vista della ionizzazione del drogante a temperatura ambiente.

Incorporare altri tipi di atomi nel reticolo cristallino del diamante è un'impresa ardua, a causa della sua compattezza. Per la prima volta, nel 1997 Koizumi e il suo gruppo di ricerca [22] hanno portato a termine con successo la crescita di film di diamante drogato con fosforo: tecniche di caratterizzazione ottica come la spettroscopia a ionizzazione fototermica (PTIS), la spettroscopia IR a trasformata di Fourier (FTIR) e la catodoluminescenza (CL) hanno confermato l'attività del fosforo come donore, ma allo stesso tempo è stato dimostrato che il livello energetico introdotto dal fosforo è di 0.6 eV al di sotto del minimo della banda di conduzione, un valore ancora troppo elevato

per considerare efficace il drogaggio con fosforo. Nonostante queste difficoltà, è stato comunque possibile realizzare giunzioni *pn* e LED in diamante.

2.6 Generazione e ricombinazione di carica nei fotorivelatori

In un semiconduttore ideale non sono presenti livelli energetici consentiti per gli elettroni all'interno della *gap*: dunque per gli elettroni più energetici in banda di valenza, il livello energetico più vicino a disposizione è ad una distanza in energia pari proprio alla *gap* del semiconduttore. Un elettrone in questa situazione non è disponibile alla conduzione: infatti, per potersi muovere in seguito all'interazione con un campo elettrico, un elettrone deve avere la possibilità di acquisire una certa quantità di energia cinetica, cosa possibile solo in banda di conduzione. Pertanto, è prima di tutto necessario che l'elettrone superi la barriera costituita dalla *gap*: in tal caso viene generata una coppia elettrone-lacuna, con un elettrone in banda di conduzione e una lacuna in banda di valenza, entrambi disponibili alla conduzione. Nel caso in cui l'unica energia disponibile sia quella termica, la probabilità per la transizione di un elettrone dalla banda di valenza alla banda di conduzione è proporzionale a $exp(- E_g / kT)$.
Nel caso di un semiconduttore reale, la presenza di stati consentiti all'interno della *gap* non può essere completamente esclusa. In questo caso, la generazione e ricombinazione di portatori di carica è fortemente intensificata, infatti:

- La generazione termica aumenta in virtù del fatto che la transizione banda-banda può essere ottenuta con due transizioni successive, ciascuna corrispondente ad un'energia minore della *gap*.
- L'assorbimento di radiazione può avvenire anche per fotoni con energia minore della *gap*.
- La ricombinazione può avvenire attraverso due successive interazioni elettrone-fonone e lacuna-fonone, ed è dunque agevolata rispetto alla ricombinazione diretta, che può essere vista come un interazione a tre "corpi" (elettrone, lacuna e fonone).

Considerando un film di diamante, è presente una grande concentrazione di difetti, che introducono stati consentiti all'interno della *gap*: questi difetti sono dovuti alle particolari condizioni di crescita del film. In fig. 2.8 sono rappresentate la quattro possibili transizioni che un elettrone può compiere in presenza di uno stato di difetto interno alla *gap*; si fa l'ipotesi che la densità di difetti sia sufficientemente bassa da poter affermare che ogni difetto introduca un solo livello energetico (a energia E_t) all'interno della *gap*.
Il primo meccanismo, indicato con r_1, è la cattura elettronica: un elettrone passa dalla banda di conduzione (E_c) ad uno stato libero di difetto localizzato nella *gap* (E_t). La velocità con cui avviene il processo è direttamente proporzionale alla densità elettronica *n* in banda di conduzione, alla densità di stati liberi di difetto presenti in banda proibita e alla probabilità per uno di essi di catturare un elettrone. La densità degli stati liberi di difetto all'interno della *gap* è data, all'equilibrio, da $N_t[1 - f(E_t)]$, dove N_t è la densità degli stati di difetto e $1 - f(E_t)$ è la probabilità che uno stato di difetto sia libero (la

funzione $f(E_t)$ è la ben nota distribuzione di Fermi-Dirac). La probabilità di cattura elettronica è data dal prodotto fra la velocità termica v_{th} dell'elettrone e la sezione di cattura σ_n del difetto. Pertanto:

$$r_1 = n\{N_t[1 - f(E_t)]\} v_{th}\sigma_n \qquad [II.3]$$

Il secondo processo, indicato con r_2, è l'emissione elettronica, ed è il processo inverso della cattura elettronica. La sua velocità è data dal prodotto fra la densità di stati occupati di difetto all'interno della *gap* $N_t f(E_t)$ per la probabilità e_n della transizione:

$$r_2 = \{N_t f(E_t)\} e_n \qquad [II.4]$$

La probabilità e_n di emissione può essere calcolata eguagliando le velocità dei processi di cattura ed emissione elettronica all'equilibrio:

$$e_n = v_{th}\sigma_n n_i \exp\left(\frac{E_t - E_i}{kT}\right) \qquad [II.5]$$

dove n_i è la concentrazione intrinseca del materiale e $E_i = E_g/2$.

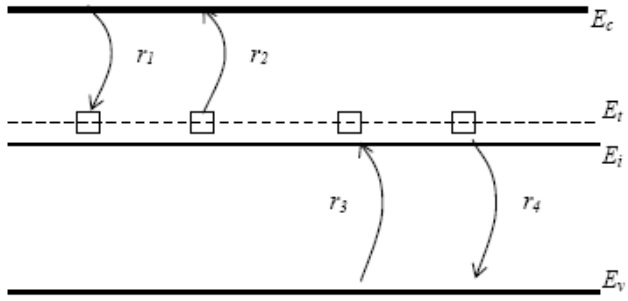


Fig. 2.8 *Possibili transizioni elettroniche in presenza di uno stato di difetto*

Ragionamenti ed espressioni del tutto analoghe possono essere compiuti per la cattura di un elettrone dalla banda di valenza (r_3), che può essere vista come l'emissione di una lacuna, e l'emissione di un elettrone dalla banda di valenza (r_4), che può essere vista come la cattura di una lacuna.
Gli stati localizzati nella *gap* possono comportarsi come *trappole* oppure come *centri di ricombinazione*: nel primo caso il portatore catturato ha un'elevata probabilità di essere nuovamente eccitato allo stato di portatore libero per effetto termico (detrappolamento attivato termicamente), mentre nel secondo caso la probabilità di ricombinazione è maggiore di quella di detrappolamento. In generale, se il livello di difetto è vicino al centro della *gap* agisce preferenzialmente come centro di ricombinazione, mentre se è vicino al limite della banda di conduzione (o di valenza) agisce soprattutto da trappola.

Nei semiconduttori a larga *gap* come il diamante, è possibile avere trappole a distanze dal limite di una banda molto superiori rispetto al quanto termico *kT* a temperatura ambiente: questo implica che l'emissione elettronica dai livelli trappola è molto lenta se attivata solo termicamente.

2.6.1 Modello di Shockley-Hall-Read (SHR)

La teoria SHR tratta il solo caso dei centri di ricombinazione, e assume che la popolazione dei centri non venga modificata in condizioni di non-equilibrio termodinamico. Il rispetto di questa condizione implica che $r_1 - r_2 = r_3 - r_4$, e pertanto il tasso di ricombinazione effettivo sarà:

$$U = R - G = r_1 - r_2 = r_3 - r_4 \qquad \text{[II.6]}$$

E' possibile eliminare la dipendenza dalla distribuzione di Fermi-Dirac $f(E_t)$ e riscrivere:

$$U = \frac{pn - n_i^2}{\tau_{n0}\left[p + n_i \exp\left(\frac{E_i - E_t}{kT}\right)\right] + \tau_{p0}\left[n + n_i \exp\left(\frac{E_t - E_i}{kT}\right)\right]} \qquad \text{[II.7]}$$

dove:

$$\tau_{n,p0} = \frac{1}{N_t v_{n,p} \sigma_{n,p}} \qquad \text{[II.8]}$$

Si fanno ora alcune ipotesi realistiche per meglio comprendere il significato fisico dei parametri che compaiono nella [II.7]. E' possibile innanzitutto considerare uguali i tempi di vita medi dei portatori all'equilibrio $\tau_{n0} = \tau_{p0} = \tau_0$, il che vuol dire che i centri di ricombinazione sono ugualmente efficaci per elettroni e lacune. Considerando poi una situazione di non-equilibrio in cui viene generato nel semiconduttore un eccesso di portatori (ad esempio per assorbimento di una radiazione), si fa l'ipotesi che la densità dei portatori in eccesso sia bassa rispetto alla concentrazione di equilibrio:

$$\begin{aligned} n' &<< n_0 + p_0 \\ p' &<< n_0 + p_0 \end{aligned} \qquad \text{[II.9]}$$

Sotto queste ipotesi il tasso di ricombinazione effettivo diventa:

$$U = -\frac{dn'}{dt} = \frac{n'(n_0 + p_0)}{\tau_0\left[n_0 + p_0 + 2n_i \cosh\left(\frac{E_t - E_i}{kT}\right)\right]} \qquad \text{[II.10]}$$

Inoltre, l'evoluzione temporale della concentrazione elettronica può essere espressa come:

$$n'(t) = n'(0)\exp\left(-\frac{t}{\tau_n}\right) \qquad \text{[II.11]}$$

dove:

$$\tau_n = \tau_0 \left[\frac{n_0 + p_0 + 2n_i \cosh\left(\frac{E_t - E_i}{kT}\right)}{n_0 + p_0} \right] \qquad \text{[II.12]}$$

Il tempo di vita medio τ_n del portatore in eccesso dipende non solo dalla densità di difetti e dalla loro sezione di cattura, ma anche dalla posizione dello stato di difetto all'interno della *gap*: se lo stato di difetto, ad esempio, corrisponde al livello di Fermi intrinseco E_i al centro della *gap*, il tempo di vita assume il suo valore minimo, ovvero il centro di ricombinazione ha la massima efficienza.

2.6.2 Presenza di un livello trappola

Si considera ora il caso di un semiconduttore con una densità non trascurabile di livelli trappola. Nella fig. 2.9 è schematicamente rappresentato il caso di una trappola per elettroni, insieme a tutte le possibili transizioni.

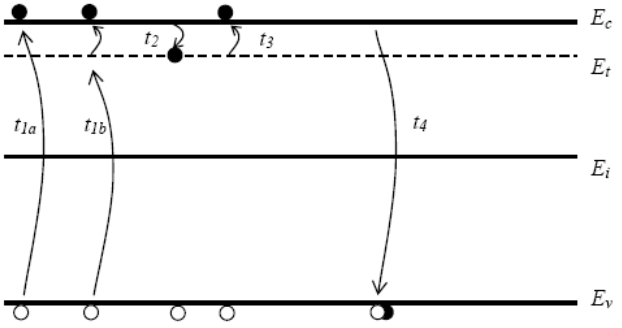


Fig. 2.9 *Possibili transizioni elettroniche in presenza di un livello trappola*

La principale differenza rispetto alla trattazione precedente è che non è permessa la transizione di un elettrone dalla banda di valenza allo stato di difetto: la generazione di una coppia elettrone-lacuna può essere pertanto ottenuta solo con una transizione diretta

banda-banda (t_{1a}), oppure sfruttando lo stato di difetto (t_{1b}). Dopo la generazione, un elettrone può essere catturato da una trappola (t_2): a questo punto, l'unica transizione possibile per tornare allo stato di equilibrio è la re-emissione dalla trappola (t_3), seguita dalla ricombinazione (diretta, come indicato in t_4, oppure attraverso un centro di ricombinazione).
Come accennato in precedenza, nei semiconduttori a larga *gap* come il diamante, la distanza in energia di una trappola dal limite di una delle due bande può essere molto grande, e il processo di re-emissione (t_3) può essere molto lento. Non appena un elettrone viene intrappolato, una lacuna è resa disponibile per la conduzione. Questo fenomeno, detto *fotoconduzione persistente*, si manifesta con un lungo transitorio nelle misure di fotoconduzione:

$$I_{ph} \propto \left[1 - \exp\left(-\frac{t}{\tau}\right)\right] \qquad \text{[II.13]}$$

dove τ è il tempo di risposta dovuto all'intrappolamento. Il tempo τ può essere espresso come:

$$\tau = \nu_{esc}^{-1} \exp\left(\frac{E_t}{kT}\right) \qquad \text{[II.14]}$$

La quantità v_{esc} è la frequenza di fuga dalla trappola (nel diamante ha un valore di circa 10^{12} s^{-1}), mentre E_t è l'energia di ionizzazione della trappola.

2.7 Assorbimento di radiazioni ottiche e fotoconducibilità

Se un fotone avente energia $h\nu$ maggiore della *gap* incide su un semiconduttore, può essere assorbito, e la sua energia può essere usata per promuovere un elettrone dalla banda di valenza alla banda di conduzione: la concentrazione di entrambi i portatori di carica risulta dunque aumentata. Il *rate* di fotogenerazione, definito come il numero di portatori fotogenerati per unità di tempo e di volume è dato da:

$$G = \frac{dn}{dt} = \frac{I\alpha(h\nu)}{h\nu} \qquad \text{[II.15]}$$

dove I è l'intensità della radiazione incidente, che decresce esponenzialmente nel semiconduttore secondo la legge di Lambert-Beer $I = I_0 \exp(-\alpha z)$, mentre $\alpha(h\nu)$ è il coefficiente di assorbimento.
Al processo di generazione si oppone il processo di ricombinazione, che riporta il semiconduttore verso una condizione di equilibrio termodinamico. Supponendo un flusso Φ di fotoni, si possono scrivere le seguenti equazioni di equilibrio:

$$\frac{dn(t)}{dt} = \alpha\Phi + G_{th} - Bnp - v_n\sigma_n N_t^n n$$
$$\frac{dp(t)}{dt} = \alpha\Phi + G_{th} - Bnp - v_p\sigma_p N_t^p p \qquad \text{[II.16]}$$

Considerando solo l'espressione per i portatori maggioritari, che nel diamante non intenzionalmente drogato sono le lacune [23], si può scrivere un'equazione semplificata:

$$\frac{d\Delta p}{dt} = \alpha\Phi - \frac{\Delta p}{\tau} \qquad \text{[II.17]}$$

L'ultimo termine rappresenta tutti i termini della [II.16] per le lacune, tranne il termine di fotogenerazione. Δp è la differenza fra le densità delle lacune nelle due condizioni di non-equilibrio e di equilibrio ($p - p_0$). La generazione termica è considerata nel termine $G_{th} = p_0/\tau$, mentre i due termini di ricombinazione sono compresi nel tempo di vita τ come $1/\tau = Bp + v_p\sigma_p N_t^p p$. Le due equazioni [II.16] tengono infatti conto di due meccanismi di ricombinazione: la ricombinazione assistita da uno stato di difetto, definita anche monomolecolare (rappresentata dai termini $v_{n,p}\sigma_{n,p}N_t^{n,p}n, p$) e la ricombinazione diretta, detta anche bimolecolare (rappresentata dal termine Bnp, dove B è la costante di ricombinazione bimolecolare). Si può dimostrare [24] che:

- In caso di bassi livelli di eccitazione luminosa (oppure alta densità di difetti) prevalgono meccanismi di ricombinazione assistita da difetti (monomolecolare), e la quantità di portatori fotogenerati è direttamente proporzionale al flusso della radiazione incidente: $n, p \propto \Phi$.
- In caso di alti livelli di eccitazione luminosa (oppure bassa densità di difetti) prevalgono meccanismi di ricombinazione banda-banda (bimolecolare), e la quantità di portatori fotogenerati ha un andamento non lineare con il flusso della radiazione incidente, più precisamente: $n, p \propto \sqrt{\Phi}$

Passiamo ora a determinare l'espressione della fotocorrente: poiché la densità di corrente è direttamente proporzionale alla densità dei portatori di carica, in seguito all'assorbimento di radiazione si potrà osservare un aumento di corrente. La corrente totale che scorre nel semiconduttore può essere espressa come:

$$J = q[(n_0 + \Delta n)\mu_n + (p_0 + \Delta p)\mu_p]F \qquad \text{[II.18]}$$

dove $\mu_{n,p}$ è la mobilità dei portatori, mentre F è il campo elettrico applicato. La componente dovuta all'assorbimento di radiazione ottica è data da:

$$J_{ph}(x) = q(\mu_n + \mu_p)\frac{\Phi(x)\alpha\tau}{h\nu}F \qquad \text{[II.19]}$$

dove è stata resa esplicita la dipendenza del flusso fotonico Φ dalla distanza *x* dalla superficie. Integrando la [II.19] su tutto lo spessore del rivelatore (da $x = 0$ a $x = d$) si ottiene la fotocorrente:

$$I_{ph}(x) = q(\mu_n + \mu_p)\Phi_0[1-\exp(-\alpha d)]\frac{z\tau}{h\nu}F \qquad \text{[II.20]}$$

dove *z* è la larghezza del contatto metallico. Infine, se si considera che una parte della radiazione incidente può essere riflessa dalla superficie del rivelatore, con coefficiente di riflessione *R*, si può ottenere l'espressione completa della fotocorrente:

$$I_{ph}(x) = (1-R)q(\mu_n + \mu_p)\Phi_0[1-\exp(-\alpha d)]\frac{z\tau}{h\nu}F \qquad \text{[II.21]}$$

Il coefficiente di riflessione per un'interfaccia piana fra l'aria e un materiale avente indice di rifrazione $n(\lambda)$ è dato dalla seguente formula:

$$R = \left[\frac{1-n(\lambda)}{1+n(\lambda)}\right]^2 \qquad \text{[II.22]}$$

Per il diamante, il cui indice di rifrazione è circa 2.9 (intorno a 200 nm) [25], circa il 23% della radiazione UV incidente si perde per riflessione all'interfaccia. La percentuale riflessa potrebbe però essere maggiore a causa della natura policristallina del diamante CVD: la superficie del campione può mostrare un'elevata rugosità, il che incrementa la riflessione. La rugosità superficiale dipende fortemente dalle condizioni di crescita, e viene generalmente ridotta attraverso tecniche di lappatura chimico-meccanica (CMP), che permettono tra l'altro di ottenere una superficie adatta all'utilizzo di processi fotolitografici per la definizione dei contatti.

2.8 Assorbimento di raggi X

A differenza delle radiazioni ottiche, in cui l'interazione con il materiale del fotorivelatore porta alla creazione di una sola coppia elettrone-lacuna per fotone incidente, nel caso di assorbimento di raggi X si assiste al fenomeno della *ionizzazione secondaria*: quando un fotone di energia notevolmente superiore alla *gap* (è il caso dei raggi X, dei raggi γ, ma anche delle particelle cariche) incide sul materiale del fotorivelatore, si ha la produzione di coppie elettrone-lacuna in eccesso, ovvero si ha un rapporto maggiore di 1 fra fotone incidente e numero di coppie create.
L'energia necessaria per creare una coppia elettrone-lacuna in eccesso per ionizzazione secondaria è indicata con E_{pair}, ed è pari nel diamante a circa 13 eV, dunque un'energia grosso modo pari a 3 volte l'energia della *gap*: si devono infatti tenere in considerazione, nel bilancio energetico, tutti i fenomeni di perdita di energia subiti dai

portatori dopo la ionizzazione secondaria. Si può dimostrare [26] che per ogni tipo di materiale semiconduttore, indipendentemente dal tipo di ionizzazione (ovvero se dovuta ad un'interazione con una radiazione elettromagnetica ad alta energia o con una particella carica) è valida la seguente espressione per E_{pair}:

$$E_{pair} = E_i + 2E_t + rE_R \qquad \text{[II.23]}$$

E_i, detta *soglia di ionizzazione*, è l'energia assorbita nella generazione di una singola coppia elettrone-lacuna, e in un modello semplificato può essere considerata uguale all'energia della *gap*, dunque $E_i = E_g$.
Il portatore fotogenerato (elettrone o lacuna che sia) è dotato di un'energia cinetica maggiore della soglia di ionizzazione E_i: parte di questa energia in eccesso viene utilizzata per generare coppie elettrone-lacuna addizionali, finché il processo di ionizzazione secondaria non si esaurisce quando il portatore fotogenerato torna al di sotto della soglia di ionizzazione E_i. Alla fine del processo di ionizzazione secondaria, l'energia cinetica di ogni portatore in eccesso sarà pari a E_t (ovviamente dovrà essere $E_t < E_i$). Per un elettrone in banda di conduzione, utilizzando un semplice modello parabolico dell'energia dell'elettrone in funzione del suo momento, si può ricavare la seguente espressione:

$$E_t = \frac{3}{5}E_i = \frac{3}{5}E_g \qquad \text{[II.24]}$$

Di conseguenza, la [II.23] si può semplificare, ottenendo:

$$E_{pair} = 2.2E_g + rE_R \qquad \text{[II.25]}$$

Infine, parte dell'energia in eccesso del portatore fotogenerato viene trasmessa al reticolo cristallino, e si manifesta attraverso la produzione di fononi ottici (*scattering* Raman, vedi par. 2.4.1): di questo fenomeno si tiene conto, nel calcolo di E_{pair}, nel termine rE_R. La quantità E_R è l'energia trasferita al reticolo per *scattering* Raman: poiché lo *shift* Raman $\Delta\nu$ corrispondente al fonone ottico nel diamante è pari a 1332.5 cm^{-1}, si può valutare E_R nel diamante come $E_R = hc\,\Delta\nu$ = 0.165 eV. Il coefficiente r è invece uguale al rapporto L_i / L_R fra il cammino libero medio dei portatori (L_i) e il cammino libero medio per lo *scattering* Raman (L_R), e rappresenta in sostanza il numero medio di fononi ottici prodotti ad ogni ionizzazione: usando il modello di Drude, il cammino libero medio dei portatori può essere stimato come $L_i \sim 500$ Å, mentre il cammino libero medio per lo *scattering* Raman è dell'ordine di $L_R \sim 50$ Å, e pertanto si ha $r \sim 10$. Usando per il diamante E_g = 5.5 eV, r = 10 ed E_R = 0.165 eV, si ottiene un valore teorico approssimato E_{pair} = 13.8 eV per la creazione di una coppia elettrone-lacuna. Risultati sperimentali hanno evidenziato dei valori per E_{pair} compresi fra 13 e 13.2 eV [27].

Capitolo III

Realizzazione e caratterizzazione di rivelatori a singolo pixel

3.1 Introduzione

Per valutare la possibilità di realizzare efficienti rivelatori multipixel 1D e 2D per la diagnostica di sorgenti UV e X, si è reso necessario uno studio di fattibilità preliminare. E'stata dunque prima di tutto realizzata una semplice struttura a singolo pixel, attraverso la quale sono state effettuate una serie di verifiche utili per una comprensione esaustiva del comportamento del rivelatore in condizioni di buio e sotto radiazione:

- Verifica della migliore configurazione di campo elettrico per la raccolta delle cariche fotogenerate (planare o *sandwich*)
- Studio delle caratteristiche di buio (tensione-corrente) per la valutazione dell'ohmicità dei contatti e della corrente di buio.
- Studio della fotoconducibilità spettrale per valutare la qualità del diamante CVD in termini di discriminazione UV-VIS, e di presenza di stati consentiti all'interno della *gap* (indice delle densità di centri di ricombinazione e livelli trappola).
- Analisi della fotoconducibilità UV impulsata per la valutazione della velocità di risposta del rivelatore a impulsi laser veloci (laser a eccimeri), e della linearità di tale risposta con l'intensità della radiazione incidente.
- Verifica di sensibilità ai raggi X attraverso misure di fotocorrente (per la verifica della linearità della risposta) e misure a discriminazione energetica (per valutare la risoluzione in energia del rivelatore e dimostrare la sua utilità anche per applicazioni spettroscopiche).

Ognuna delle operazioni sperimentali di verifica preliminare verrà illustrata nei successivi paragrafi: nel complesso, lo studio di fattibilità a singolo pixel ha dato esiti più che soddisfacenti, incoraggiando la realizzazione di rivelatori multipixel 1D e 2D.

3.2 Struttura del rivelatore a singolo pixel

Partendo da un substrato di partenza di silicio, sono stati depositati spessi strati di diamante policristallino da una miscela di metano e idrogeno, utilizzando una tecnica di deposizione chimica da fase vapore assistita da plasma (MWCVD). Il substrato di silicio è stato in seguito rimosso attraverso un attacco chimico per via umida (*wet etching*), ed

entrambe le superfici di diamante sono state lappate meccanicamente fino a rimuovere circa 100 μm di materiale: dopo quest'operazione di lappatura, la rugosità superficiale media risulta essere compresa nell'intervallo 2 ÷ 8 nm (fig. 3.1). Le operazioni descritte fino ad ora sono state eseguite dal *team* del prof. Victor Ralchenko presso l'Istituto di Fisica Generale dell'Accademia delle Scienze di Mosca.

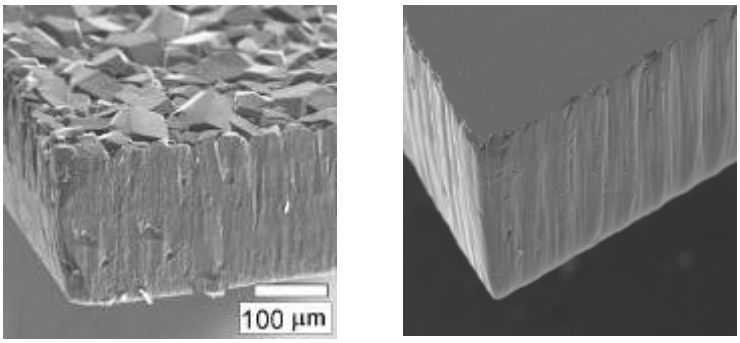


Fig. 3.1 *Il campione di diamante prima e dopo l'operazione di lappatura*

In seguito, è stato selezionato un campione di area 10 x 10 mm^2 e di spessore 270 μm, successivamente pulito attraverso un processo di attacco chimico (vedi par. 3.3.1). Per la realizzazione dei contatti sono stati depositati su entrambi i lati del campione 200 nm di argento (scelto per la natura ohmica del contatto su diamante non idrogenato, nonché per la facilità di deposizione) per evaporazione termica PVD, e successivamente su un lato è stata definita per fotolitografia una struttura interdigitata (fig. 3.2) con *fingers* di larghezza 80 μm e distanti 80 μm l'uno dall'altro: una struttura di questo tipo consente di effettuare misurazioni sia in configurazione *planare* (tensione applicata fra i due elettrodi interdigitati) che in configurazione *sandwich* (tensione applicata fra l'elettrodo posteriore e i due elettrodi interdigitati messi in cortocircuito).

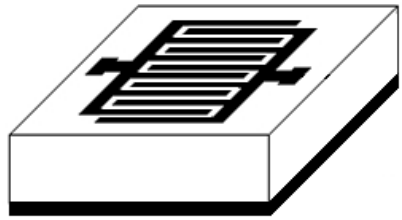


Fig. 3.2 *Schema della struttura multifinger fotolitografata sul campione*

Allo scopo di ridurre il più possibile l'effetto di sorgenti esterne di rumore (interferenze elettromagnetiche, *pick-ups* di varia origine, etc.), il rivelatore è stato inserito all'interno di una scatola metallica sulla quale sono stati praticati due fori: uno superiore per permettere l'arrivo delle radiazioni, e uno laterale per l'installazione di un connettore BNC (utilizzato per il prelevamento del segnale e la polarizzazione del dispositivo). La fig. 3.3 mostra il rivelatore completo: si può notare la struttura di sostegno del campione

e i fili di collegamento col connettore BNC, saldati sui *pads* del campione con pasta conduttiva d'argento.

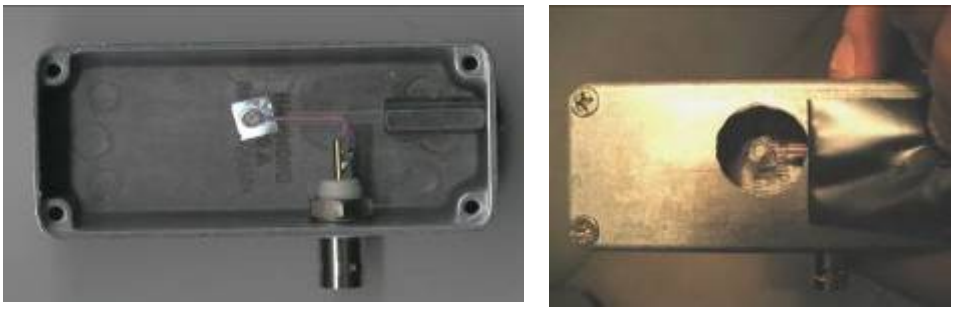


Fig. 3.3 *Struttura del rivelatore a singolo pixel*

3.3 Fabbricazione dei rivelatori

Nei successivi paragrafi verranno descritte le procedure di base per la realizzazione del rivelatore a singolo pixel. Operazioni dello stesso tipo sono state eseguite anche per i rivelatori multipixel 1D e 2D (vedi cap. V), con l' unica differenza di aver utilizzato maschere differenti per la fotolitografia dei contatti. Inoltre, per i rivelatori multipixel si è resa necessaria un'operazione finale di *wire bonding* per il collegamento finale di ogni pixel all'elettronica di *front-end*, mentre per il rivelatore a singolo pixel è stato sufficiente l'impiego di una pasta conduttiva di argento.

3.3.1 Pulizia superficiale dei substrati

Attualmente, sebbene non esista una procedura standard di pulizia superficiale dei film in diamante, si sono sviluppati diversi processi che hanno dimostrato una certa efficacia nella rimozione delle fasi non-diamante: tra questi, i trattamenti di *annealing* termico in atmosfere controllate e di attacco chimico in ambiente acido sembrano dare i migliori risultati. Nel presente lavoro, il trattamento di pulizia adottato è simile a quello proposto da Kania [28]. I passi seguiti sono:

- Bagno in acetone caldo all'interno di una cella a ultrasuoni (15 min.) per la rimozione di residui organici e microparticolati.
- Attacco chimico in mistura solfocromica (K_2CrO_7 + H_2SO_4) in ebollizione (5 min.) per la rimozione di residui metallici (relativi a metallizzazioni precedenti) e di fasi non-diamante.
- Risciacquo in soluzione RCA (H_2O_2 + NH_4OH) diluita al 30% per la rimozione di eventuali residui in seguito all'attacco chimico in mistura solfocromica.
- Risciacquo in acqua deionizzata corrente.
- Asciugatura in flusso di azoto ultrapuro.

3.3.2 Metallizzazione

Per la deposizione dei metalli necessari alla realizzazione dei contatti dei dispositivi, si può ricorrere a diverse tecniche, fra cui: l'evaporazione termica (PVD), lo *sputtering* e l'evaporazione assistita da fascio elettronico (EBL). La prima, più semplice in termini di apparato, è stata quella adottata nel presente lavoro di ricerca. Vengono qui di seguito brevemente descritti l'apparato e i tipici parametri di deposizione utilizzati.
Lo schema di massima per l'apparato PVD (Ionvac EVP 300-2S) è riportato in fig. 3.4. L'evaporazione avviene in condizioni di vuoto piuttosto elevato ($10^6 \div 10^7$ Torr), raggiunto per mezzo di un gruppo di pompaggio costituito da una pompa rotativa (pre-vuoto) e una pompa turbomolecolare: il livello di vuoto determina direttamente la qualità dell'evaporazione, in quanto con esso si evita la possibile ossidazione del metallo e qualunque forma di contaminazione proveniente dall'ambiente esterno.
Per riscaldare il crogiolo (in tungsteno in caso di evaporazione di Ag, ma sono disponibili anche crogioli in molibdeno e tantalio per l'evaporazione di altri tipi di metalli) alla temperatura di evaporazione del metallo, posto su di esso in forma solida granulare, lo si fa percorrere da un'elevata corrente: con correnti di qualche centinaio di A si raggiungono temperature fino a 1000 °C.

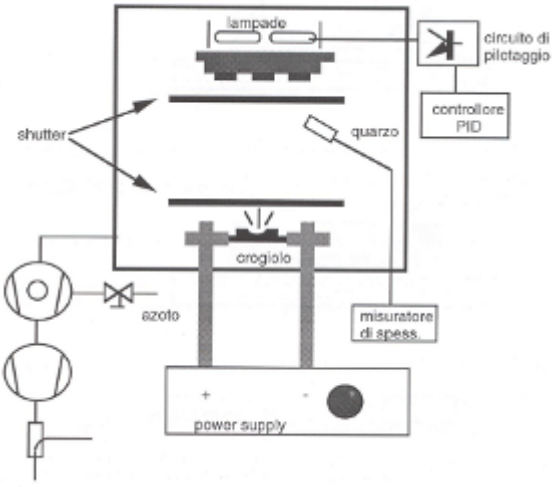


Fig. 3.4 *Schema dell'apparato utilizzato per l'evaporazione termica*

Il misuratore di spessore (Leybold Inficon, XTC 751-001-2G) permette il controllo *in situ* della quantità di metallo evaporato: esso è costituito da un oscillatore al quarzo la cui piastrina piezoelettrica è posta all'interno della camera di evaporazione. All'aumentare della quantità di deposito sulla piastrina, diminuisce la frequenza di oscillazione in seguito all'aumento di massa: per le quantità di materiale depositato utilizzate tipicamente (sempre < 1 μm), la variazione di frequenza è sempre direttamente proporzionale allo spessore del deposito stesso. Impostando la densità del metallo da evaporare, lo strumento è in grado di indicare il *rate* di evaporazione e lo spessore raggiunto; solo ogni 10 ÷ 20 evaporazioni è necessario sostituire la piastrina piezoelettrica con una nuova, oppure rimuovere da quella usata il metallo depositato.

Gli *shutter* indicati nello schema evitano che sui campioni possa finire materiale contaminante proveniente dal crogiolo o dai granuli di metallo da evaporare. Aprendo lo *shutter* più vicino al crogiolo, il sensore al quarzo del misuratore di spessore può indicare il *rate* di evaporazione del metallo; quando viene raggiunto un valore costante, si può aprire il secondo *shutter* e cominciare la deposizione sul campione. Raggiunto lo spessore voluto, determinato sempre dal misuratore a quarzo, l'evaporazione viene interrotta chiudendo lo *shutter* superiore e, con una rampa decrescente della durata di 5 minuti circa, viene ridotta a zero la corrente nel crogiolo.
L'immissione successiva di azoto attraverso la valvola di ventilazione posta sulla pompa turbomolecolare (solo dopo aver spento il gruppo di pompaggio) permette il raffreddamento dei substrati riscaldati, per irraggiamento, dal crogiolo stesso, oppure intenzionalmente attraverso un sistema di lampade alogene: spesso si ha infatti la necessità di far diffondere superficialmente il metallo evaporato, e per questo motivo si scaldano i substrati durante l'evaporazione. Come sistema riscaldatore, si è scelto un semplice sistema a lampade alogene da 450 W: un controllore PID di temperatura pilota un circuito di parzializzazione di fase che, modulando l'alimentazione delle lampade, determina il riscaldamento dei portasubstrati alla temperatura impostata. Per i rivelatori realizzati, poiché i coefficienti di diffusione nel diamante sono molto bassi per la maggior parte degli elementi, il riscaldamento del substrato garantisce soprattutto una maggiore adesione del metallo sul film.
I tipici parametri di processo adottati sono riassunti nella tabella sottostante:

Parametri	**Argento**	**Cromo**
Vuoto base [Torr]	$5 \cdot 10^{-7}$	$3 \cdot 10^{-6}$
Spessore realizzato [nm]	50 ÷ 200	100
Rate di evaporazione [Å/s]	0.5 ÷ 5	1 ÷ 3
Corrente nel crogiolo [A]	280 ÷ 300	200 ÷ 240

Tab. 3.1 *Parametri utilizzati nell'evaporazione termica*

3.3.3 Fotolitografia

Dopo la metallizzazione, con tecnica fotolitografica a contatto [29], è possibile definire il *pattern* prescelto per gli elettrodi (struttura interdigitata per il rivelatore a singolo pixel, *strip* per il rivelatore 1D, e pixel di forma quadrata per il rivelatore 2D). In questo paragrafo verranno esaminati in sintesi i fondamentali passi fotolitografici seguiti.
Il primo passo consiste nella realizzazione delle maschere sulle quali deve essere riportato il *pattern* dei contatti metallici. Sono state realizzate, in tempi successivi, tre maschere (vedi fig. 3.5): una per gli elettrodi interdigitati del rivelatore a singolo pixel, una per le *strip* del rivelatore 1D, e infine una per i pixel del rivelatore 2D. I passi seguiti per la realizzazione di una fotomaschera sono elencati qui di seguito:

- Progetto del *pattern* attraverso programmi di grafica (Corel Draw, Lasi), e realizzazione di un *file bitmap* contenente i dati relativi al *pattern*.

- Deposizione di un film di 100 nm di cromo su un supporto di quarzo per evaporazione termica (vedi par. 3.3.2).
- Deposizione per *spinning* di uno strato di fotoresist Shipley Microposit® SP1805, e *soft bake*.
- Scrittura diretta del *pattern* sullo strato di fotoresist attraverso un laser He-Cd, il cui sistema di controllo legge i dati contenuti nel *file bitmap* realizzato in precedenza.
- Sviluppo del fotoresist con Shipley Microposit® 351 Developer, e *hard bake*.
- Attacco chimico per la rimozione del cromo non protetto da fotoresist.
- Rimozione del fotoresist residuo attraverso risciacquo in acetone.
- Risciacquo in acqua deionizzata e asciugatura in flusso di azoto ultrapuro.

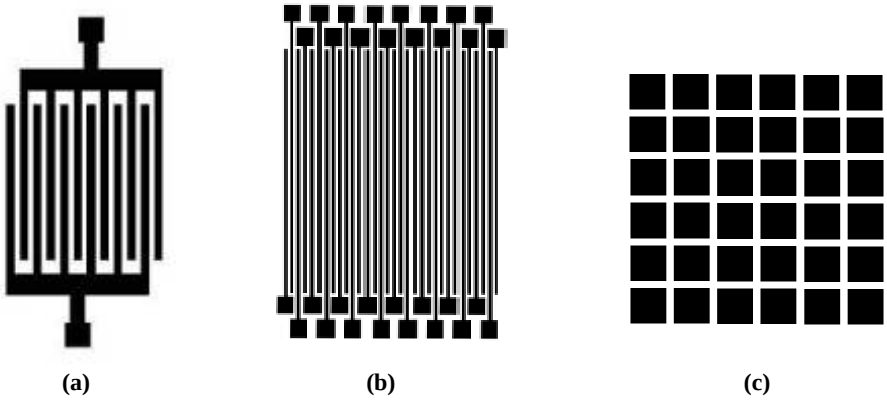


Fig. 3.5 *Pattern realizzati su fotomaschera: struttura interdigitata (a), strip detector (b), pixel detector (c)*

In seguito alla realizzazione delle maschere, si passa alla fotolitografia dei contatti sul dispositivo.
Dapprima viene depositato sulla superficie metallizzata il fotoresist (polimero fotosensibile alla radiazione UV-blu): si è utilizzato il resist positivo Shipley Microposit® SP1805, che consente di ottenere risoluzioni fino a 0.5 μm [30]. Per la deposizione del fotoresist è stato utilizzato uno *spinner* (Headway Research EG-101) che, ponendo in rapida rotazione (3000 giri/min. per 40 s) il campione su cui ne sono state versate poche gocce, permette di espandere il liquido, formando una sottile pellicola di spessore uniforme. La deposizione delle gocce di fotoresist è stata effettuata con una siringa dotata di filtri, per garantire un'ottima pulizia del deposito. Dopo la deposizione, si procede ad un *soft bake* (90 °C per 12 min.), al fine di far evaporare buona parte del solvente in cui il fotoresist è disciolto.
La riproduzione per contatto del *pattern* presente sulla maschera è stata effettuata con il *mask aligner* Karl Suss MA-6, avente risoluzione massima pari a 1 μm. In realtà, per risoluzione s'intende, in questo caso, il massimo disallineamento che si ha in due esposizioni successive: poiché per i rivelatori realizzati è stato semplicemente necessario riprodurre le strutture (elettrodi interdigitati, *strip*, pixel) della maschera, si è utilizzato il

Karl Suss MA-6 per una singola esposizione, e non sono state sfruttate pienamente le sue capacità di allineamento.
Per l'esposizione, l'apparato utilizza una lampada a mercurio da 350 W (Osram HBO 350 W), il cui spettro di emissione è fortemente assorbito dal resist Shipley SP1805. La lampada è posta su uno dei due fuochi di uno specchio ellissoidale; l'altro fuoco giace invece sul piano dove si trova un'aletta di raffreddamento. Tra lampada e ottica di dispersione del fascio luminoso è presente uno specchio a 45° per "luce fredda", nel senso che riflette con buona efficienza luce a bassa λ, mentre trasmette quasi interamente luce a λ elevata: in questo modo viene agevolmente dissipato l'elevato calore emesso dalla lampada. La dispersione uniforme della luce verso il sistema di condensatori è effettuata da una serie di lenti ad "occhio di mosca": l'intero sistema rende parallelo il fascio di luce verso il substrato da esporre.
L'apparato Karl Suss MA-6 è completamente automatizzato. Montati la maschera e il substrato sui loro relativi supporti (che garantiscono la tenuta grazie a un sistema di aspirazione), si procede all'esposizione mediante uno fra tre possibili programmi:

- HP (*vacuum contact*): tra la maschera e il substrato viene creato il vuoto. Questo consente una maggiore adesione, e dunque la più alta risoluzione, poiché il *gap* fra maschera e substrato viene reso minimo.
- ST (*standard contact*): durante l'esposizione, il vuoto nel porta substrato viene interrotto, e viene invece immesso azoto ad una certa pressione, in grado di comprimere il substrato contro la maschera.
- SC (*soft contact*): il substrato viene spinto contro la maschera attraverso una semplice pressione meccanica.

A causa della scabrosità della superficie di diamante, si è sempre utilizzato uno tra gli ultimi due programmi sopraelencati (ST o SC). Dalle prove effettuate, il tempo ottimale di esposizione è stato tra i 15 e i 20 secondi.
Dopo l'esposizione, si procede allo sviluppo del fotoresist in una soluzione 3:1 a base di soda caustica (Shipley Microposit® 351 Developer): tipicamente, in poche decine di secondi il fotoresist esposto è completamente rimosso dalla soluzione di sviluppo. L'osservazione dello sviluppo può essere effettuata con luce filtrata gialla: prima di ogni osservazione, naturalmente, si procede ad un risciacquo in acqua deionizzata corrente e a un'asciugatura in flusso di azoto ultrapuro.
Completato lo sviluppo, per rendere il fotoresist resistente ai successivi bagni per l'attacco chimico del metallo, si procede ad un *hard bake* (120 °C per 15 min.).

3.3.4 Attacco chimico

L'attacco chimico del metallo non protetto da fotoresist è stato compiuto mediante le soluzioni acide riportate nella tabella 3.2. Il tempo di attacco è chiaramente dipendente dallo spessore del metallo, ma tipicamente è stato inferiore ai 5 minuti. Tutti gli attacchi sono stati eseguiti a temperatura ambiente.
Ogni attacco chimico è seguito da un bagno in acetone per la rimozione del fotoresist residuo, seguito da un risciacquo in acqua deionizzata corrente per qualche minuto, e infine da un'asciugatura in flusso di azoto ultrapuro.

Argento	Cromo
$H_2O + HNO_3$ (2:1)	Microposit® Chrome Etch 18

Tab. 3.2 *Soluzioni di attacco chimico utilizzate per Ag e Cr*

Per verificare se tutto il metallo è stato effettivamente rimosso dalle aree non coperte dal fotoresist, è necessaria un'osservazione al microscopio ottico. Poiché il fotoresist risulta piuttosto sensibile alla luce visibile (soprattutto nei confronti delle componenti al di sotto dei 450 nm), tutte le osservazioni vanno condotte con luce gialla.
La fotolitografia delle strutture realizzate (*strip*, pixel, elettrodi interdigitati) è risultata essere piuttosto agevole, e si è ottenuta una buona definizione del *pattern*. Tuttavia, a causa della natura policristallina del diamante utilizzato, spesso è capitato che alcuni contatti poco risolti siano andati in cortocircuito: in tal caso è stato necessario rimuovere completamente il metallo e ricominciare la realizzazione dei contatti dalla procedura di pulizia superficiale. In fig. 3.6 sono riportati alcuni esempi di immagini al microscopio ottico delle strutture elettrodiche realizzate nell'ambito della presente attività.

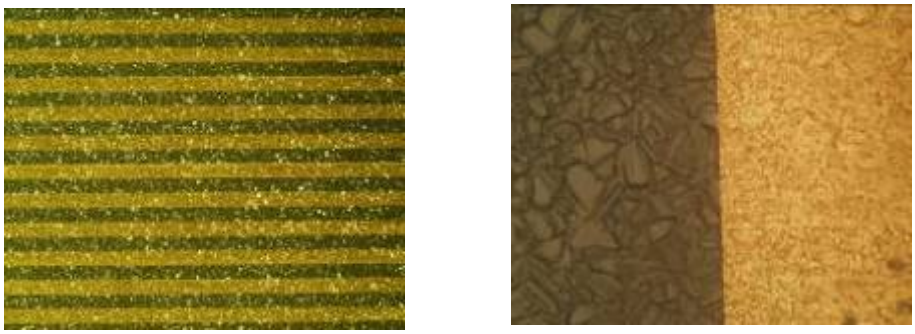


Fig. 3.6 *Immagini al microscopio ottico di struttura a strip*

3.4 Studio dell'ohmicità dei contatti

Una volta completata la fotolitografia dei contatti in argento, ne è stata studiata l'ohmicità attraverso misurazioni I-V, compiute con un elettrometro Keithley 6517A utilizzato in modalità picoamperometro.
Le misurazioni sono state effettuate sia in configurazione planare che *sandwich*: in effetti, la struttura colonnare dei film di diamante cresciuti con metodi CVD (vedi par. 2.4) fa supporre che debbano esistere differenze di trasporto nelle due geometrie. Nella configurazione planare, le linee di campo elettrico sono sostanzialmente confinate in superficie e ad essa parallele (fenomeni di trasporto superficiali), mentre in configurazione *sandwich* sono ortogonali alla superficie (fenomeni di trasporto di *bulk*).

3.4.1 Elettrometro Keithley 6517A

Il Keithley 6517A è un elettrometro programmabile ad alta sensibilità progettato per poter misurare tensione, corrente, capacità e resistenza. Nella tabella 3.3 sono riportati gli intervalli di misura per le diverse grandezze elettriche:

Grandezza elettrica	Intervallo di misura
Tensione	10 μV – 200 V
Corrente	0.1 fA – 20 mA
Resistenza	0.1 Ω - 200 GΩ
Carica	10 fC – 20 nC

Tab. 3.3 *Versatilità del Keithley 6517A*

L'alta impedenza d'ingresso (≈ 200 TΩ), unitamente alla bassa corrente di *offset* d'ingresso (5 fA a 23 °C, valore che raddoppia ogni 10 °C), consentono di effettuare misure molto accurate. Alle funzionalità classiche del multimetro digitale, il Keithley 6517A aggiunge la funzione di generatore di tensione nell'intervallo –102.35 ÷ 102.4 V, con un passo minimo di 50 mV e con una corrente massima di *output* di 2 mA. Ai fini pratici della misura, lo strumento utilizza un cavo triassiale per le funzionalità di multimetro e un cavo coassiale per applicare al rivelatore la tensione di polarizzazione.
In fig. 3.7 è illustrato un diagramma a blocchi semplificato [31]:

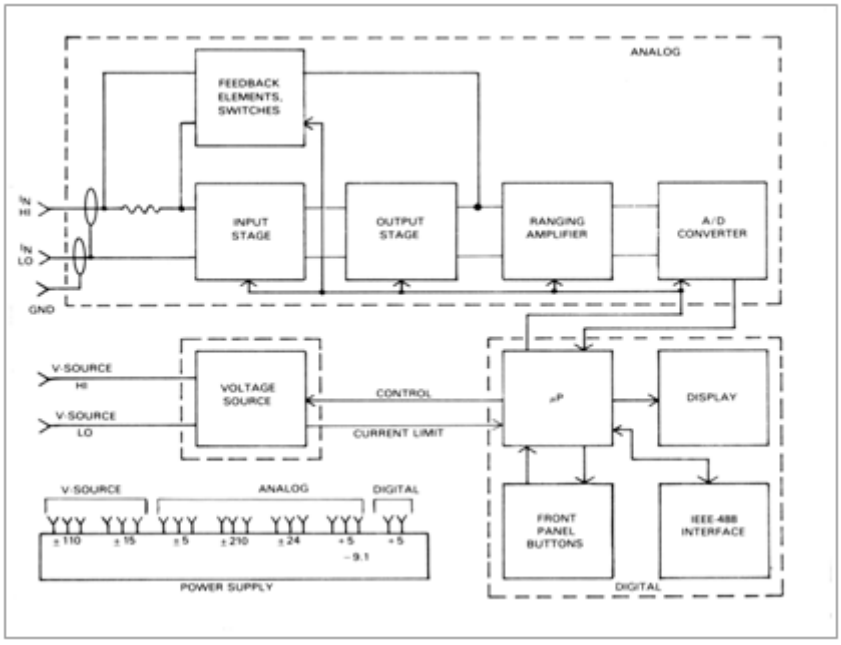


Fig. 3.7 *Schema a blocchi del Keithley 6517A*

Possiamo distinguere quattro parti essenziali: la sezione analogica, la sezione digitale, il generatore di tensione e l'alimentazione. Le prime tre sezioni sono elettricamente isolate le une dalle altre (grazie ad un sistema di alimentazioni distinte) al fine di minimizzare l'interferenza reciproca, mentre lo scambio di informazioni è assicurato da una rete di fotoaccoppiatori. Il blocco analogico consiste di uno stadio di ingresso (composto da un amplificatore FET con impedenza di ingresso di 200 TΩ e corrente di *offset* minore di 5 fA), uno stadio di uscita che fornisce un amplificazione supplementare ($V_{out} = \pm 210$ V) ed elementi di *feedback* e *switching* (per regolare il guadagno e la funzione di trasferimento dei due stadi precedenti in accordo con l'intervallo selezionato e la funzione implementata, e per la correzione dello zero). Si ha poi il *ranging amplifier*, che trasforma il segnale dello stadio di uscita in un segnale compreso tra 0 e 2 V, rendendolo compatibile con lo standard di ingresso di un convertitore A/D a 12 *bit*. Il cuore della sezione digitale è costituito dal processore (in standard CMOS 6805) che gestisce tutta la dinamica di funzionamento dello strumento. Il generatore di tensione è completamente programmabile, isolato e controllato dal microprocessore.
Nell'ambito di quest'attività di ricerca, il multimetro Keithley 6517A è stato utilizzato nella modalità picoamperometro; in questa configurazione lo strumento può lavorare in undici intervalli con differente risoluzione, come riportato nella tabella 3.4:

Corrente massima	**Risoluzione**
2 pA	100 aA
20 pA	1 fA
200 pA	10 fA
2 nA	100 fA
20 nA	1 pA
200 nA	10 pA
2 μA	100 pA
20 μA	1 nA
200 μA	10 nA
2 mA	100 nA
20 mA	1 μA

Tab. 3.4 *Prestazioni del Keithley 6517A nella modalità picoamperometro*

Nell'ambito delle operazioni di misura, si è assunto che l'incertezza sistematica fosse pari alla somma dell'incertezza intrinseca dello strumento e dell'incertezza relativa al valore della resistenza di ingresso; quest'ultima è esprimibile in forma relativa come:

$$\frac{\Delta I}{I} = \frac{1}{\frac{V}{V_B} - 1} \qquad \text{[III.1]}$$

dove V è la tensione applicata e V_B è la caduta ai capi del picoamperometro (è immediato notare come l'incertezza sia inversamente proporzionale alla tensione applicata).
Infine, per limitare resistenza e accoppiamenti capacitivi, la lunghezza dei cavi è stata il più possibile minimizzata, e l'utilizzo di fili schermati ha reso il sistema quasi immune da disturbi di carattere elettromagnetico.

3.4.2 Risultati sperimentali

In fig. 3.8 sono riportate le caratteristiche di corrente di buio del rivelatore, nelle due configurazioni planare e *sandwich*, al variare della tensione applicata: più precisamente, per poter effettuare un efficace confronto fra le due geometrie, la corrente di buio è rappresentata in funzione del campo elettrico applicato, vista la diversa spaziatura fra gli elettrodi nelle due configurazioni.

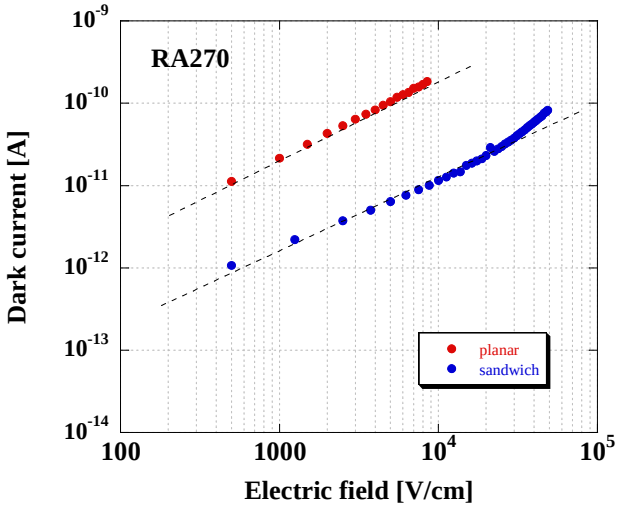


Fig. 3.8 *Caratteristiche di buio nelle due configurazioni planare e sandwich*

Si nota innanzitutto che in configurazione planare la caratteristica è lineare fino ad un campo elettrico applicato di circa $9 \cdot 10^3$ V/cm (che corrisponde ad una tensione di *bias* di circa 70 V): al di sopra di tale campo, la corrente di buio subisce un brusco aumento (arriva a qualche μA), probabilmente a causa della comparsa di cammini conduttivi superficiali dovuti alla formazione di microplasmi nelle zone di bordo-grano fra gli elettrodi interdigitati. La possibilità di sfruttare campi elettrici più intensi, per una più efficace raccolta delle cariche, è uno dei motivi che rende preferibile per i nostri scopi la configurazione *sandwich*. Come si vede dalla fig. 3.8, la configurazione *sandwich* assicura un comportamento ohmico fino a circa $3 \cdot 10^4$ V/cm (che corrisponde ad una tensione di *bias* di circa 800 V): al di sopra di tale valore di campo elettrico, si può osservare un comportamento leggermente sopralineare, probabilmente dovuto a meccanismi di conduzione in presenza di carica spaziale (SCLC, *Space Charge Limited Current*), come ampiamente riportato in letteratura [32, 33].

Inoltre si può osservare che in configurazione *sandwich*:

- Si ha una corrente di buio inferiore, rispetto al caso planare, a parità di campo elettrico.
- Le linee di forza del campo elettrico seguono la struttura colonnare del diamante CVD: questo agevola la raccolta delle cariche fotogenerate [34], che nel loro tragitto verso gli elettrodi si trovano a viaggiare in una direzione povera di difetti (e questo accresce la loro mobilità).
- Il campo elettrico è pressoché uniforme in tutto lo spessore del campione, e questo è un requisito essenziale in particolar modo per la discriminazione in energia della radiazione incidente (tassativo per i rivelatori per spettroscopia a raggi X).

3.5 Studio della fotoconducibilità spettrale

La principale indicazione della qualità di un campione di diamante è la densità degli stati di difetto in banda proibita: una tecnica efficace per questo tipo di valutazione, nonché per la determinazione sperimentale della discriminazione UV-VIS, è la fotoconducibilità spettrale, che consiste sostanzialmente nella misura della corrente generata dall'assorbimento di una radiazione monocromatica. In un semiconduttore ideale non esistono stati consentiti nella *gap*, e dunque non si ha alcun fenomeno di fotogenerazione in caso di fotoni incidenti con energia minore della *gap*; in un semiconduttore reale, invece, si trova sempre una soglia di fotoconducibilità molto più bassa in termini di energia. La posizione in energia di questa soglia è relativa alla posizione all'interno della *gap* di uno stato di difetto, mentre l'ampiezza della risposta è relativa alla densità di quel tipo di stato di difetto.

3.5.1 Apparato di misura

In fig. 3.9 è riportato lo schema complessivo del banco utilizzato per la fotoconducibilità spettrale del dispositivo. Il sistema è principalmente costituito da: sorgente luminosa allo xenon; monocromatore a reticolo motorizzato; ruota di filtri colorati; supporto portacampione; amplificatore *lock-in* con *chopper*; PC di controllo e acquisizione dati. Poiché il banco è stato utilizzato per misure nell'intervallo spettrale 200 ÷ 800 nm, tutta l'ottica è in quarzo.
Vengono qui di seguito descritti più in dettaglio i vari componenti del banco di misura.
Come sorgente luminosa capace di coprire lo spettro d'interesse, e soprattutto di avere una potenza adeguata fino a lunghezze d'onda inferiori alla *gap* del diamante ($\lambda < 225$ nm), è stata scelta una lampada a scarica a Xe-Hg (Perkin Elmer, mod. PE 300 BUV). La potenza totale assorbita è di 300 W, mentre quella luminosa, integrata sull'intero spettro, è di 30 ÷ 40 W circa. Poiché per lunghezze d'onda inferiori a 242 nm si ha dissociazione delle molecole d'ossigeno con formazione di ozono (O_3), è necessario utilizzare un semplice sistema di aspirazione e abbattimento dell'ozono (Oriel, *Ozone Eater* mod. 66087).

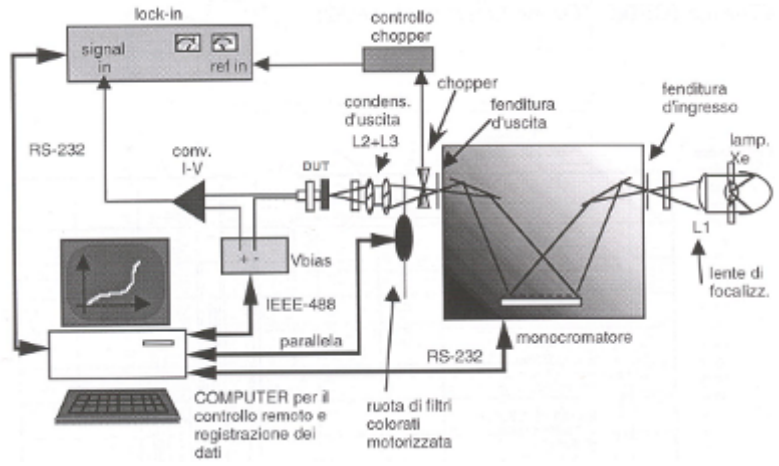


Fig. 3.9 *Schema del banco di fotoconducibilità spettrale*

Tra lampada e monocromatore è presente un sistema di focalizzazione (L1). Per ottimizzare l'ingresso di luce nel monocromatore è sempre consigliabile porre un sistema ottico (detto di *matching*) tra sorgente e fenditura d'ingresso: infatti, i fotoni che possono essere raccolti dal monocromatore sono solo quelli che entrano nella fenditura d'ingresso con l'angolo solido di accettazione del monocromatore stesso. Quindi il fascio luminoso della sorgente viene generalmente dapprima reso parallelo, e poi focalizzato sulla fenditura mediante una lente di focale uguale all'*F-number* del monocromatore. La lampada utilizzata possiede sul retro uno specchio parabolico, e dunque il fascio di luce emesso è pressoché parallelo; per focalizzarlo poi sulla fenditura d'ingresso del monocromatore si utilizza la lente biconvessa L1.
Il monocromatore, Spex mod. 1680, è schematizzato in fig. 3.10: è composto dalle fenditure d'ingresso e d'uscita, da una serie di specchi e da due reticoli di diffrazione (monocromatore doppio).

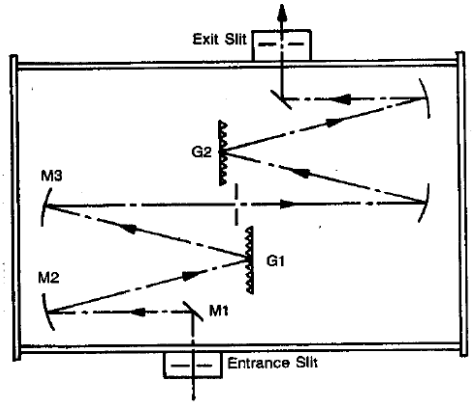


Fig. 3.10 *Schema ottico del doppio monocromatore Spex mod. 1680*

La luce focalizzata sulla fenditura d'ingresso viene inviata da uno specchio a 45° (M1) ad uno specchio collimante (M2), e da qui dispersa, attraverso il reticolo G1, sullo specchio M3: poiché il reticolo seleziona la luce in base alla lunghezza d'onda, solo una ristretta banda di lunghezze d'onda può passare attraverso la fenditura centrale e raggiungere la seconda parte del monocromatore. Qui, la luce viene ulteriormente "raffinata" in lunghezza d'onda, subendo un'ulteriore dispersione dal reticolo G2 prima di essere focalizzata sulla fenditura d'uscita. Questa semplice struttura permette di avere un'alta luminosità (l'apertura numerica è *f*/4) e una risoluzione spettrale più che adeguata alle necessità sperimentali (con una fenditura di uscita di 0.5 μm si ha una risoluzione spettrale di 0.2 nm). Il monocromatore è infine dotato di un sistema di movimentazione del reticolo: esso consiste in un motorino passo-passo che, tramite un circuito di pilotaggio, può essere interfacciato ad un PC mediante interfaccia RS-232. Un programma di controllo (vedi par. 3.5.1.2) regola, attraverso tale interfaccia, la posizione del reticolo in base alla lunghezza d'onda che si vuole far incidere sul dispositivo, impostabile da remoto.
Immediatamente all'uscita del monocromatore è posta una ruota composta da tre filtri colorati passa-alto in lunghezza d'onda, che tagliano rispettivamente a 380, 520 e 680 nm: i filtri risultano necessari poiché, qualora venga selezionata in uscita una determinata λ, sono presenti anche le armoniche ad energia maggiore ($\lambda/2$, $\lambda/3$, $\lambda/4$, etc.). E'stato dunque necessario sezionare lo spettro 200 ÷ 800 nm in quattro parti, come indicato nella tabella 3.5, dove è stato riportato il filtro utilizzato nel particolare intervallo:

Intervallo spettrale [nm]	**λ di taglio [nm]**	**Colore filtro**
200 ÷ 380	—	nessun filtro
380 ÷ 520	380	blu
520 ÷ 680	520	giallo
680 ÷ 800	680	IR (nero)

Tab. 3.5 *Filtri per il taglio delle armoniche di ordine superiore*

La ruota di filtri, movimentata da un motorino in continua, è controllata dal PC tramite la porta parallela.
Il sistema ottico in uscita (condensatore d'uscita, lenti L2 e L3) ha il compito di focalizzare il fascio uscente dal monocromatore sul campione: l'ingrandimento è stato scelto in modo da formare uno *spot* luminoso che avesse dimensioni confrontabili con l'area utile del rivelatore (circa 5 x 5 mm^2).
Per le misure del segnale di fotocorrente si è utilizzato un amplificatore a transimpedenza (EG&G mod. 5182), avente un fattore di conversione selezionabile tra 10^5 e 10^8. La tensione di polarizzazione è stata invece ottenuta utilizzando l'elettrometro Keithley 6517A (vedi par. 3.4.1) come generatore di tensione.
La luce che incide sul campione è modulata in intensità da un *chopper* meccanico, che ruota tipicamente ad una frequenza compresa nell'intervallo 1 ÷ 30 Hz. Il segnale modulato di tensione viene infine misurato con un amplificatore *lock-in* (EG&G mod. 5205), che merita una descrizione più dettagliata.

3.5.1.1 Amplificatore lock-in

Il metodo *lock-in* consente la misura di segnali di ampiezza molto bassa e banda molto stretta nell'intorno di una frequenza di riferimento. Il segnale di riferimento è generato da un sistema LED-fotodiodo montato sul *chopper*, amplificato, trasformato in livelli TTL e quindi inviato all'amplificatore *lock-in* ("Ref. In" sullo schema di fig. 3.11). In pratica l'amplificatore *lock-in* è in grado, entro una banda ristretta, di selezionare il segnale che abbia pari frequenza e differenza di fase costante rispetto al segnale di riferimento. Lo schema a blocchi del *lock-in* utilizzato è riportato in fig. 3.11, e ne viene per completezza descritto il principio di funzionamento, che si basa sulla modulazione eterodina.

Il segnale d'ingresso V_i alla frequenza f_r viene miscelato con un segnale generato da un VCO (oscillatore controllato in tensione), la cui frequenza di uscita è legata direttamente a quella f_r del segnale di riferimento. Il segnale in uscita dal *mixer*, a frequenza somma e differenza, viene successivamente elaborato da un amplificatore quasi-risonante alla frequenza intermedia f_i, in modo da rimuovere la componente ad alta frequenza. Il segnale in uscita da tale amplificatore può infine essere elaborato, e contiene l'informazione originaria del segnale d'ingresso V_i, ma a frequenza intermedia f_i. Con questo metodo si possono dunque utilizzare filtri e amplificatori che lavorino in una banda molto ristretta (intorno a f_i) e non a banda larga, riducendo notevolmente il rumore di tali circuiti.

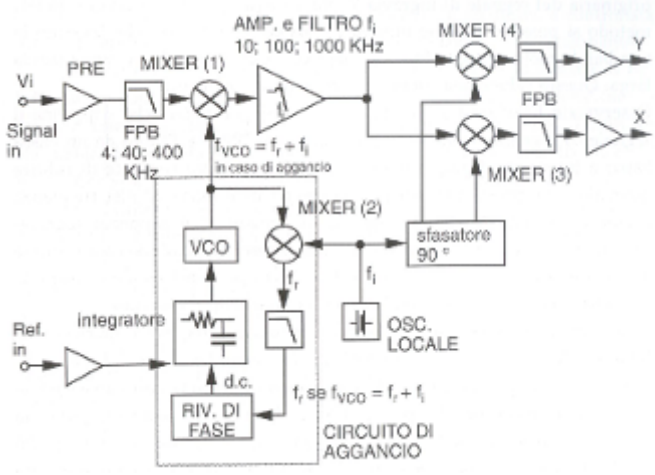


Fig. 3.11 *Schema a blocchi semplificato di un amplificatore lock-in*

Osservando lo schema a blocchi, si può notare che il segnale d'ingresso V_i viene dapprima amplificato da un preamplificatore (PRE) e filtrato da un passa-basso a frequenza di taglio selezionabile: il filtro consente di ridurre notevolmente già in ingresso le componenti di rumore in alta frequenza presenti sul segnale, e quindi di aumentare il rapporto segnale-rumore. Con un'osservazione attenta dello schema è possibile comprendere il principio di funzionamento del *lock-in*, la cui parte principale è costituita dal blocco "circuito di aggancio", che viene ora in sintesi descritto.

Si noti innanzitutto la presenza dell'oscillatore locale a frequenza intermedia f_i. Il blocco "rivelatore di fase" rivela eventuali sfasamenti tra i segnali f_{rif1} e f_{rif2}: se i due hanno frequenza diversa, lo sfasamento non risulta costante, e cambia l'uscita del rivelatore che pilota il VCO, finché non viene raggiunta la condizione di aggancio $f_{VCO} = f_r + f_i$. Infatti, solo in questo caso l'uscita del *mixer* 2 è un segnale di frequenza f_r pari a quella di riferimento. Generata la f_{VCO}, il *mixer* 1 trasla l'informazione originaria del segnale V_i (di frequenza f_r) alla frequenza intermedia f_i. Il blocco "amplificatore e filtro f_i", essendo molto selettivo, è in pratica quello che attenua il più possibile i disturbi presenti su V_i. Infine, per ricavare le componenti X e Y, rispettivamente in fase e in quadratura del segnale, vengono utilizzate le due sezioni di demodulazione costituite dai *mixer* 3 e 4 e dai relativi passa-basso in uscita. Una volta amplificati, i segnali possono essere presentati in uscita, poiché posseggono l'informazione completa delle componenti X e Y del segnale V_i originario.
I filtri passa-basso in uscita ai *mixer* 3 e 4 rappresentano un'altra sezione fondamentale dell'intero amplificatore, poiché la loro presenza migliora notevolmente il rapporto segnale-rumore. In pratica, qualunque componente di rumore nell'intorno della frequenza intermedia f_i viene traslata a bassa frequenza (verso la continua) dai *mixer*: in teoria, scegliendo quindi passa-basso con costante di tempo molto elevata, tali componenti di rumore possono essere attenuate di molto. Chiaramente, ciò aumenta enormemente i tempi di risposta del sistema, e quindi il tempo necessario affinché i segnali X e Y si assestino può diventare esageratamente alto: nei casi pratici, dunque, bisogna raggiungere un compromesso tra rapporto segnale-rumore e tempi di risposta.

3.5.1.2 Controllo e acquisizione dati

Tutto il banco è automatizzato grazie al controllo da parte di un PC interfacciato con la strumentazione: più precisamente, il *lock-in* e il monocromatore comunicano col PC mediante due interfacce RS-232, mentre la sorgente di polarizzazione usa un'interfaccia IEEE-488. E'stato realizzato un programma che rendesse il più possibile autonoma la misura: infatti, nel caso di livelli di fotocorrente molto bassi, occorre scegliere costanti di tempo elevate (1 ÷ 3 s), e un intero spettro può richiedere più di un'ora per essere completato.
Il programma sceglie la scala di sensibilità e la costante di tempo in modo automatico: più precisamente, per ogni lunghezza d'onda, si cerca il valore di sensibilità per il quale lo strumento misura $a_1 FS \leq |V| \leq a_2 FS$, dove a_1 e a_2 sono parametri che possono essere impostati (generalmente pari a 1/3 e 2/3 rispettivamente) e FS è il fondo-scala per la fissata sensibilità. Poiché la misura è comunque affetta da rumore, la ricerca della sensibilità ottima è effettuata campionando un certo numero di punti sperimentali (impostabile, e tipicamente pari a 10). Trovata la scala di sensibilità, per la costante di tempo fissata dal punto sperimentale relativo alla lunghezza d'onda λ precedente, il programma passa alla misura di N campioni (con N scelto dall'utente), per ricavare i valori medi del modulo $|V(\lambda)|$ e della fase $\varphi(\lambda)$ del segnale, e gli scarti quadratici medi $\sigma_V(\lambda)$ e $\sigma_\varphi(\lambda)$ ad essi relativi. Gli scarti quadratici medi risultano fondamentali per rendere intelligente la misura in termini temporali: infatti, se lo scarto relativo al modulo del segnale è molto più piccolo della sensibilità impostata (secondo un fattore che può essere scelto), evidentemente la costante di tempo è troppo elevata, e dunque il programma modifica la costante di tempo, passandola al valore immediatamente

inferiore; nel caso opposto, qualora lo scarto sia confrontabile con la sensibilità stessa, la costante di tempo viene aumentata e, in questo caso, viene ripetuta la misura di N nuovi campioni alla stessa lunghezza d'onda.
I valori λ, $|V(\lambda)|$, $\varphi(\lambda)$, $\sigma_V(\lambda)$ e $\sigma_\varphi(\lambda)$ sono registrati su disco rigido per ogni λ, in modo da limitare i danni derivanti da un'improvvisa interruzione dell'alimentazione di rete.

3.5.2 Risultati sperimentali

In fig. 3.12 è mostrata la fotoconducibilità spettrale (in termini di efficienza quantica esterna normalizzata al valore massimo) del campione di diamante utilizzato, calcolata in funzione dell'energia dei fotoni incidenti.
Si può notare che la qualità del campione RA270 è paragonabile a quella di un diamante naturale di tipo IIa, caratterizzato da bassissime concentrazioni di impurità di azoto: la coda esponenziale di Urbach in prossimità della *gap* a 5.5 eV, che rappresenta la deviazione dall'andamento idealmente a gradino della fotoconducibilità (a causa della presenza di stati localizzati nella *gap* in prossimità dei limiti di banda [35]), presenta un'uguale larghezza in entrambi i materiali, ed è di soli 80 meV.

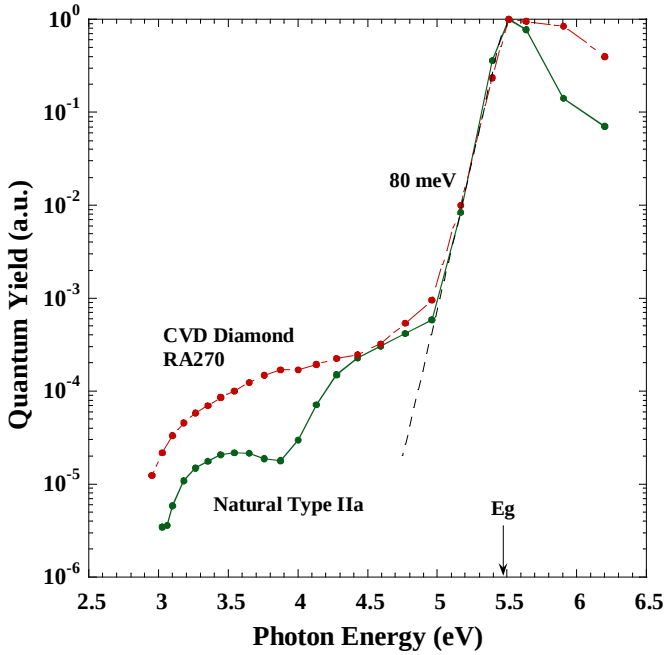


Fig. 3.12 *Confronto fra la fotoconducibilità spettrale del RA270 e quella di un diamante naturale*

L'aspetto più interessante è l'andamento della curva al di sopra della *gap*: la diminuzione della fotoconducibilità oltre i 5.5 eV, dovuta a fenomeni di ricombinazione superficiale, è più marcata nel diamante naturale che nel campione CVD lappato

meccanicamente in superficie; sotto questo aspetto, dunque, il RA270 si presenta migliore rispetto ad un campione di diamante naturale.
Da notare è infine la discriminazione UV-VIS, pari a ben 4 ordini di grandezza: questo risultato testimonia l'efficacia dell'utilizzo di rivelatori in diamante in tutte le applicazioni *solar blind* senza aver bisogno, come nel caso del silicio, di impiegare soluzioni di filtraggio.
A ulteriore prova dell'ottima discriminazione UV-VIS, è stata misurata, con lo stesso *setup* di misura utilizzato per la fotoconducibilità spettrale, una caratteristica corrente-tensione sotto illuminazione UV (lunghezza d'onda λ = 235 nm, potenza incidente P_{INC} ~ 1μW), mostrata in fig. 3.13: la fotocorrente è superiore alla corrente di buio di circa 3 ordini di grandezza, anche in condizioni d'illuminazione UV di debole intensità.

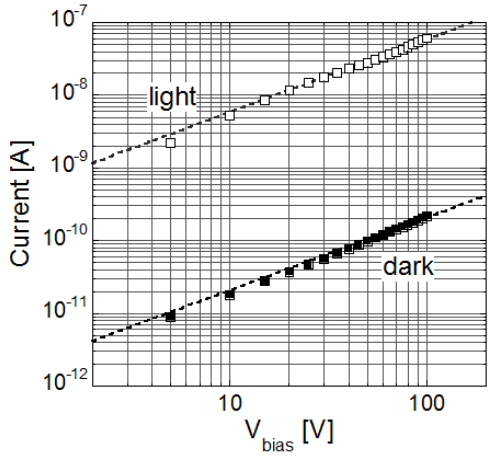


Fig. 3.13 *Caratteristica I-V in condizioni di buio e debole illuminazione UV*

3.6 Misure di fotoconducibilità UV impulsata

Al fine di valutare la velocità di risposta del rivelatore ad un'eccitazione UV impulsata e di verificarne la linearità in un ampio *range* di intensità, sono state compiute misure di fotoconducibilità sotto laser a eccimeri. Il banco di misura impiegato prevede le seguenti attrezzature (fig. 3.14):

- una sorgente laser ad eccimeri ad ArF* (193 nm);
- un sistema ottico per la focalizzazione e separazione del fascio laser emesso dalla sorgente in quattro *spot* (con intensità gradualmente minore in modo da sfruttare un ampio *range*) selezionabili per mezzo di un diaframma con fenditura di dimensioni regolabili;
- un *bias tee* necessario per la polarizzazione del campione;
- un oscilloscopio digitale ad ampia banda per la visualizzazione del segnale;
- un fotodiodo *Schottky* per fornire il segnale di *trigger* all'oscilloscopio.

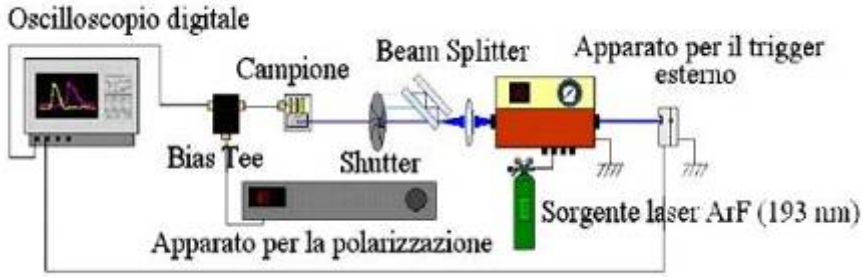


Fig. 3.14 *Setup sperimentale per le misure di fotoconducibilità impulsata UV*

3.6.1 Laser a eccimeri Neweks PSX100

La sorgente utilizzata è il Neweks PSX100, un laser ad eccimero ad ArF* che emette radiazione UV impulsata di lunghezza d'onda pari a λ = 193nm (fig. 3.15): il mezzo attivo è costituito da atomi di argon e da molecole F_2 di fluoro, mentre come gas inerte è utilizzato neon, che costituisce il 90 ÷ 99% della miscela, e che gioca il ruolo di "*carrier*" (partecipa come terzo corpo alle collisioni tra gli ioni delle specie attive che portano alla formazione delle molecole dell'eccimero).

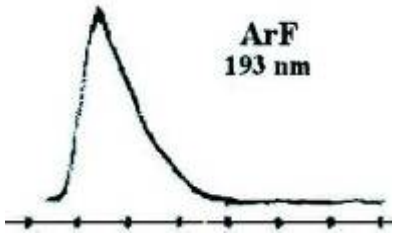


Fig. 3.15 *Impulso laser ad eccimeri ArF (10ns/div)*

Il tempo di salita di un impulso generato da questo tipo di sorgente è tipicamente dell'ordine del ns, con una durata in termini di FWHM tra 1 e 5ns.
Le principali caratteristiche del Neweks PSX100 sono:

- durata tipica degli impulsi emessi pari a 3 ns (FWHM);
- energia nominale del singolo impulso di circa 3.5 mJ;
- *rate* di ripetizione degli impulsi variabile tra 10 e 100 Hz;
- *spot* luminoso di area 3 x 3 mm^2 e semiapertura angolare del fascio pari a circa 3.5 mrad.

3.6.2 Ottica e *beam splitter*

La scelta di una sorgente ad ArF* è dettata anche dal fatto che consente ancora l'uso di un'ottica in quarzo: radiazioni di lunghezza d'onda inferiore ai 193nm consentirebbero di ottenere una fotogenerazione più elevata, ma richiederebbero l'utilizzo di una costosa ottica in fluoruro di calcio (a tali lunghezze d'onda la percentuale di radiazione trasmessa del quarzo risulterebbe troppo bassa, come evidenziato nella fig. 3.16)

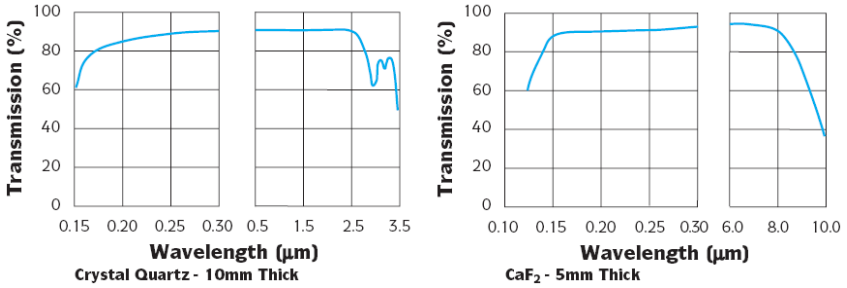


Fig. 3.16 *Confronto dei coefficienti di trasmissione del quarzo e del fluoruro di calcio (CVI Laser Optics)*

Il laser utilizzato presenta un'elevata divergenza del fascio in uscita, e la bassa coerenza spaziale non consente una focalizzazione diretta della radiazione; risulta dunque conveniente far convergere il fascio in un fuoco, per poi farlo divergere: si utilizza per questo una lente convergente in quarzo di focale pari a 10 cm. Approssimando il fascio ad un'onda sferica (modulata in ampiezza sul piano traverso da una gaussiana bidimensionale), l'intensità varierà come $1/x^2$, dove x è in prima approssimazione la distanza dal fuoco: in questo modo, allontanando progressivamente dalla sorgente il campione da valutare, è possibile analizzare la fotoconducibilità del materiale al variare dell'intensità luminosa incidente.
La sorgente laser utilizzata presenta due fenditure di uscita: una è quella principale, utilizzata per illuminare il campione (è su questa uscita che viene posizionata la lente convergente), mentre l'altra, posteriore, è caratterizzata da una minore intensità ed è utilizzata per illuminare un fotodiodo in silicio che fornisce un segnale in corrente utilizzato come *trigger* esterno per l'oscilloscopio. Al fine di aumentare il *range* di intensità disponibile in uscita, si sfrutta un *beam splitter* (una lamina di quarzo inclinata di 45° rispetto al fascio, preceduta da uno specchio riflettente di alluminio di area 30 x 20 mm^2 , spessore 10 mm e coefficiente di riflessione 80 ÷ 90% fino a $\lambda \approx 157$ nm): tale lamina, attraverso una serie di trasmissioni e riflessioni multiple, consente di ottenere più repliche del fascio, tra loro attenuate di un opportuno fattore.
Il coefficiente di trasmissione del quarzo nello spettro UV di interesse è ≈ 0.9, quindi la radiazione trasmessa F_{T1} e quella riflessa verso lo specchio F_{R1} saranno rispettivamente il 90% il 10% di quella incidente F_I. A questo punto (fig. 3.17) il fascio riflesso dal quarzo F_{R1} viene nuovamente riflesso dallo specchio: il 90% di F_{R1} va a formare un secondo fascio (F_{T2}) trasmesso dal quarzo, e il 10% di F_{R1} viene nuovamente riflesso (F_{R2}), e così via.

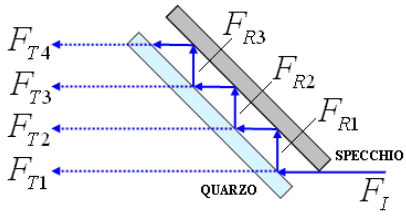


Fig. 3.17 *Schema di funzionamento del beam splitter*

Nell'ipotesi che né il quarzo né lo specchio assorbano radiazione, la separazione in intensità tra un fascio e quello adiacente è pari ad un fattore ≈ 100, come è evidente dalla fig. 3.18 (in realtà il *beam splitter* comporta sempre un'attenuazione nominale del 10%). Di seguito al *beam splitter* è posto poi un diaframma con cui si va a selezionare singolarmente uno dei quattro fasci trasmessi, ampliando fortemente il *range* di intensità con cui è possibile illuminare il rivelatore.

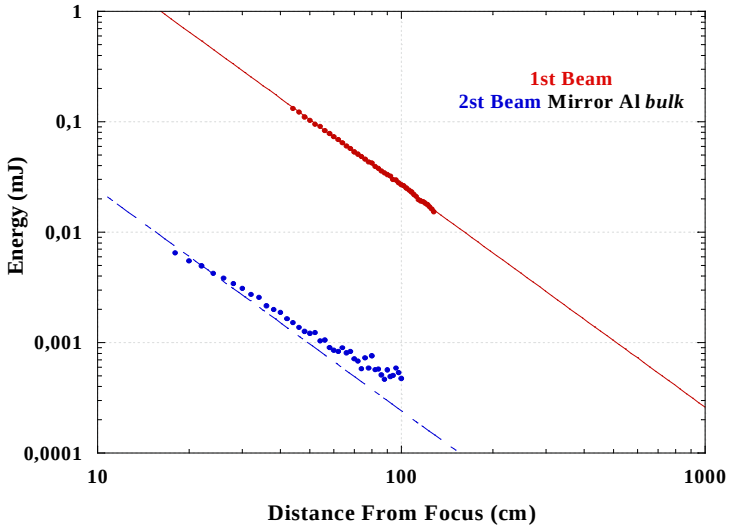


Fig. 3.18 *Energia del primo e secondo fascio in funzione della distanza dal fuoco della lente*

3.6.3 *Bias tee* Picosecond 5575A

Il campione deve essere opportunamente polarizzato in modo tale che la fotocorrente UV possa essere raccolta dagli elettrodi ed inviata all'oscilloscopio. Abbiamo l'assoluta necessità di inviare all'oscilloscopio esclusivamente il contributo di fotocorrente, termine evidentemente in alternata: il *bias tee* è un componente appositamente progettato per consentire di collegare un generatore in continua ad un cavo coassiale senza alterare la trasmissione del segnale AC o RF attraverso il cavo.

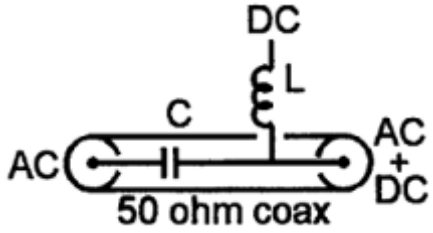


Fig. 3.19 *Schema del bias tee*

Nello schema riportato in fig. 3.19 risulta evidente come il condensatore posto sul centrale del cavo coassiale da 50 Ω costituisca un blocco per la componente DC del segnale, impedendo alla corrente continua erogata dal generatore di arrivare all'uscita AC del *bias tee*; l'induttanza ha invece lo scopo di evitare che la componente alternata della corrente fluisca verso l'alimentazione.
Per la polarizzazione del nostro campione abbiamo utilizzato un *bias tee* della Picosecond (modello 5575A) con larghezza di banda a -3 dB tra i 10 kHz e i 12 GHz, progettato appositamente per consentire il passaggio di impulsi con tempi di salita molto brevi, e introducendo una distorsione minima della forma d'onda. Il tempo di salita del dispositivo è garantito essere intorno ai 30 ps.

3.6.4 Oscilloscopio digitale LeCroy Wavepro 960

La natura intrinsecamente impulsata della radiazione UV determina nei semiconduttori dinamiche di fotogenerazione che danno luogo a segnali elettrici variabili nel tempo con una rapidità confrontabile a quella dell'eccitazione fotonica: per rivelare tali segnali è indispensabile quindi una strumentazione dotata soprattutto di un'ampia larghezza di banda. Lo strumento utilizzato per operare la lettura del segnale in uscita dal rivelatore è un oscilloscopio digitale LeCroy WavePro 960, le cui caratteristiche principali sono:

- banda analogica a –3 dB di 2 GHz, con possibilità di inserire filtri per limitare la banda a 20 MHz o 200 MHz allo scopo di ridurre il rumore.
- impedenza di ingresso standard di circa 50 Ω per consentire l'adattamento al cavo coassiale (è comunque possibile configurare la resistenza d'ingresso a un valore pari a 1MΩ).
- risoluzione verticale di 8 *bit* (è possibile arrivare anche ad 11 *bit*).
- 4 canali di ingresso e frequenze di campionamento di 16 *Gsample*/s (per un singolo canale), 8 *Gsample*/s (per due canali), 4 *Gsample*/s (per tre o quattro canali).
- controllo remoto delle funzioni interne e del pannello di controllo attraverso l'uso di interfacce di tipo IEEE-488, RS-232 o *Ethernet*.

3.6.5 Fotodiodo Schottky per il *trigger* esterno

L'oscilloscopio digitale consente di utilizzare un segnale di *trigger* automatico, che risulterebbe però asincrono rispetto all'evento di interesse (l'impulso laser): questo risulta poco conveniente, in quanto non consente di visualizzare sullo schermo "con continuità" la forma d'onda di interesse. Si utilizza pertanto un segnale di *trigger* esterno**,** fornito da un fotodiodo Schottky di silicio (inversamente polarizzato) sul quale va ad incidere il fascio proveniente dalla seconda fenditura del Neweks PSX100: si ottiene così un *trigger* sincrono col segnale utile (l'evento che genera i due segnali è lo stesso, ovvero l'impulso laser), e quindi il segnale in uscita dal rivelatore può essere efficacemente campionato ogniqualvolta il laser emette un impulso.

3.6.6 Risultati sperimentali

La verifica sperimentale consiste nel valutare il tempo di risposta del rivelatore, e al contempo nello studio di tale risposta in un ampio *range* di intensità. L'andamento nel tempo dell'impulso laser è stato preliminarmente misurato utilizzando un rivelatore commerciale (tubo fotomoltiplicatore) Hamamatsu H8496-11 [36]: dalla fig. 3.20 è possibile notare come il tempo di salita dell'impulso in uscita dal tubo a vuoto (tempo necessario per passare dal 10% al 90% del valore di picco) sia di circa 1.3 ns, mentre la discesa è approssimabile ad una coda esponenziale con costante di tempo compresa tra 2 e 3 ns (le oscillazioni alla fine dell'impulso sono da imputare soprattutto a un'interferenza elettromagnetica introdotta dalla scarica laser).

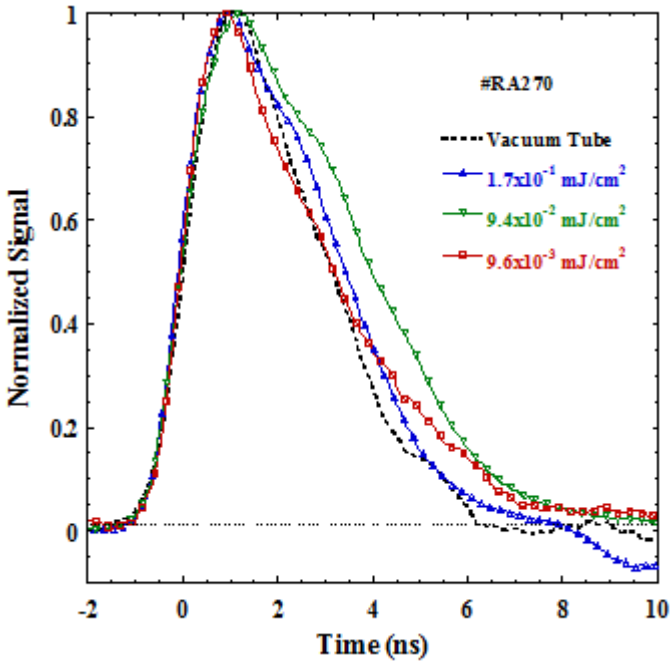


Fig. 3.20 *Impulsi di fotocorrente UV per diverse energie incidenti*

In seguito le misure sono state compiute inviando gli impulsi laser sul rivelatore in diamante, al quale è stata applicata una tensione di *bias* di 30 V: la fotocorrente è stata misurata come caduta di tensione sui 50 Ω d'ingresso dell'oscilloscopio (tale valore di resistenza è infatti trascurabile rispetto alla resistenza del campione, e dunque il segnale misurato si può sempre considerare direttamente proporzionale alla fotocorrente generata nel campione a alla sua resistenza). Come si può osservare dalla fig. 3.20, gli impulsi di fotocorrente per diversi valori di intensità incidente hanno praticamente tutti un tempo di salita paragonabile a quello (~ 1.3 ns) dell'impulso laser misurato col tubo a vuoto, a testimonianza dunque dell'ottima velocità di risposta del nostro dispositivo.
E'interessante notare come la FWHM (*Full Width Half Maximum*) dell'impulso di fotocorrente aumenti con l'intensità incidente: una possibile interpretazione di questo comportamento può essere fornita osservando la fig. 3.21, nella quale sono stati riportati i valori di picco della fototensione in funzione dell'energia incidente (l'energia dell'impulso laser è stata fatta variare in un *range* di tre decadi utilizzando i primi due fasci in uscita dal *beam splitter*, vedi par. 3.6.2).
Si può osservare un comportamento pressoché lineare ($\beta = 1$) del picco di fototensione fino ad energie incidenti dell'ordine di $5 \cdot 10^{-3}$ mJ, mentre per energie superiori si può notare una transizione verso una risposta sub-lineare ($\beta = 0.5$); in accordo con la teoria di Rose (vedi par. 2.7), questa transizione evidenzia un passaggio da un meccanismo di ricombinazione puramente monomolecolare (assistito da difetti) ad una ricombinazione puramente bimolecolare (tipica dei semiconduttori puri): questo ci fa dedurre che per aumentare il *range* di linearità della risposta impulsiva abbiamo bisogno di un materiale più difettato.

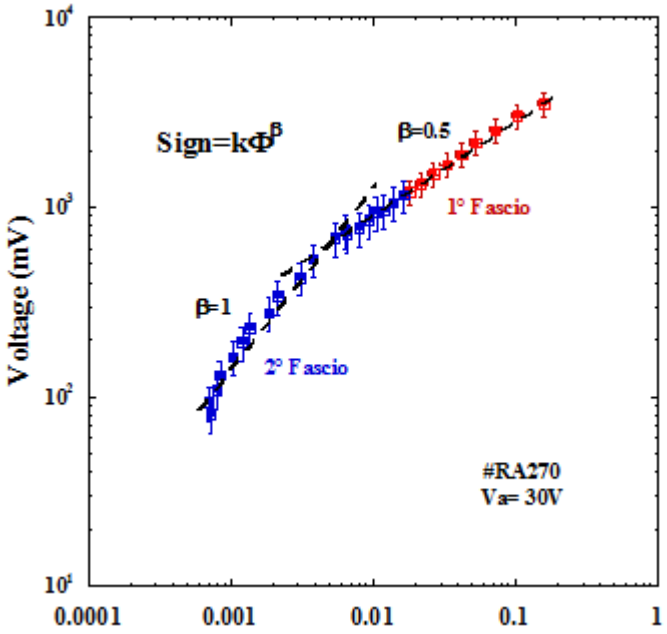


Fig. 3.21 *Linearità in energia della risposta di fotocorrente*

L'allargamento della FWHM degli impulsi di fotocorrente all'aumentare dell'energia dell'impulso laser (fig. 3.20) si può infine spiegare ipotizzando un aumento del tempo di vita medio dei portatori: la transizione verso un regime di ricombinazione bimolecolare fa sì che i difetti presenti nel materiale influenzino in misura sempre minore i meccanismi di ricombinazione, con la conseguenza diretta di un aumento del tempo di vita medio delle cariche.

3.7 Misure di fotocorrente X

Per studiare la possibilità di realizzare un rivelatore per l'analisi della forma e dell'intensità di un raggio X, sono state effettuate misure di fotocorrente, ed è inoltre stata studiata la linearità della risposta col flusso della radiazione incidente. Per i raggi X, si è utilizzata una sorgente commerciale Isodebyeflex 2001 (tubo di *Coolidge* con *target* di rame e filtro di nickel per selezionare l'emissione K_α = 8.06 keV, *spot* focale 10 x 10 mm) alimentata a 40 kV e 30 mA. Un collimatore di diametro 1 mm è stato utilizzato per convogliare la radiazione sul rivelatore. Un *revolver* con 15 filtri di alluminio di spessore crescente è stato interposto fra la sorgente e il rivelatore, al fine di ridurre l'intensità del fascio in maniera controllata. Infine, le misure corrente-tensione (fino a 100 V) sono state effettuate con un picoamperometro Keithley 6517A.

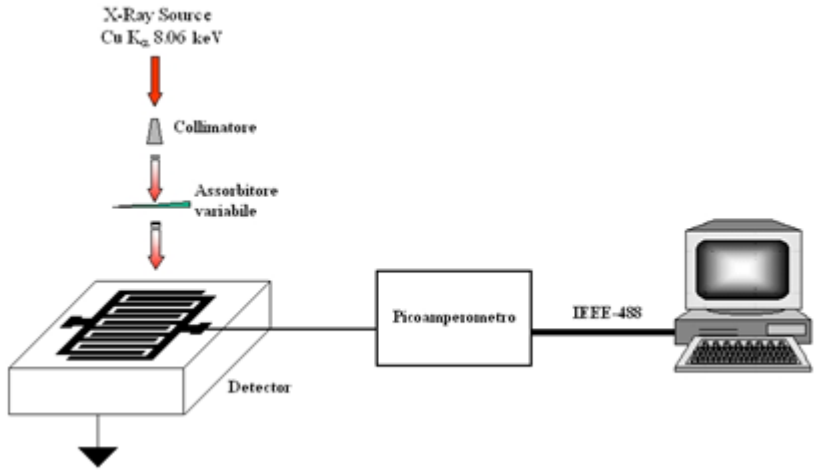


Fig. 3.22 *Setup sperimentale per le misure di fotocorrente X*

3.7.1 Sorgente di raggi X e sottosistema di filtraggio

La sorgente di raggi X utilizzata è un tubo di *Coolidge* con *target* realizzato in rame, angolo di *take-off* di 6° e tensione di innesco attorno ai 40 kV. All'uscita del tubo è posto un collimatore (fig. 3.23) che ha il compito di forzare le dimensioni trasversali del fascio, così da confinarlo spazialmente.

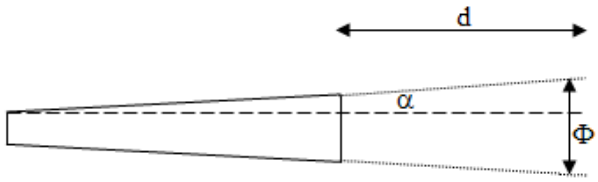


Fig. 3.23 *Collimatore per la radiazione X*

Nel caso specifico si è utilizzato un collimatore di lunghezza 60 mm, che presenta un foro circolare di diametro 0.8 mm in ingresso e un'uscita circolare di diametro pari a 1.1 mm. Le dimensioni trasversali del fascio, trascurando effetti di diffusione e diffrazione, seguono approssimativamente, per piccole distanze dall'uscita del collimatore, la relazione :

$$\Phi = 0.8 + 2(60 + d)tg\alpha \qquad \text{[III.2]}$$

Lo spettro di emissione della sorgente è mostrato in fig. 3.24: sono indicati lo spettro totale prima del filtraggio della riga K_β operato da uno spessore di nickel di 21 μm, lo spettro di assorbimento del filtro di nickel, ed infine è indicato lo spettro filtrato, in cui è presente la sola K_α e un fondo residuo di *Bremsstrahlung*.

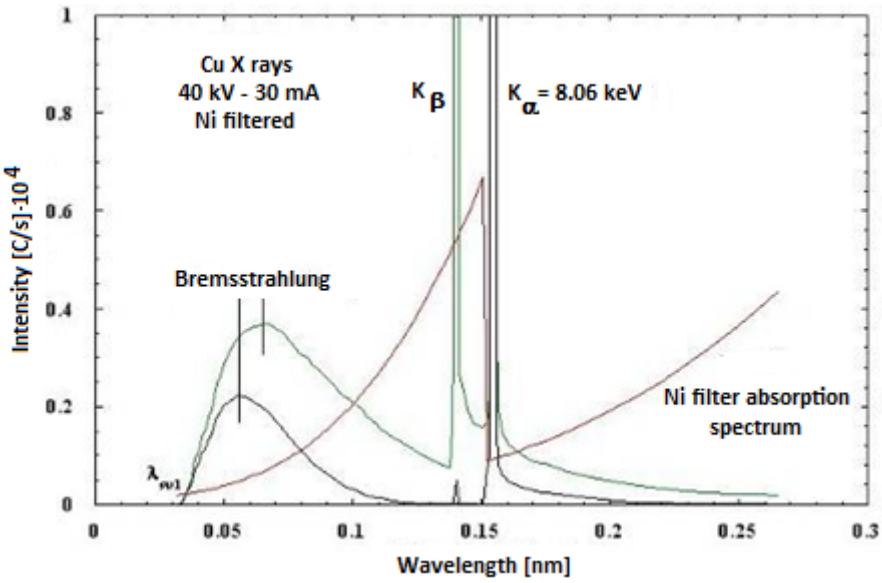


Fig. 3.24 *Filtraggio della radiazione X*

Osserviamo dunque che la radiazione che emerge si compone soltanto dello spettro continuo di *Bremsstrahlung* (con una lunghezza d'onda di taglio inferiore $\lambda_{swl} \approx 0.03$

nm) e della riga K_α a 8.06 keV: si può però ritenere la radiazione praticamente monocromatica, poiché il picco K_α possiede un'intensità circa 45 volte maggiore dell'intensità massima della radiazione di *Bremsstrahlung* sopravvissuta. Inoltre si deve considerare che la radiazione di *Bremsstrahlung* è maggiormente energetica (è a lunghezza d'onda minore) e il coefficiente di assorbimento del diamante policristallino e dell'alluminio dei filtri diminuisce di un fattore 50 all'aumentare dell'energia della radiazione incidente (poiché diminuisce la sezione di cattura per i fotoni).
Uno degli obiettivi di questa tesi è dimostrare la possibilità di impiegare una rivelatore in diamante CVD a risposta lineare con l'intensità del raggio X incidente: è necessario quindi realizzare un *test* accurato sotto varie condizioni di illuminazione, così da far emergere tanto la risposta lineare quanto eventuali comportamenti sub-lineari. L'attenuazione variabile del fascio è realizzata mediante una serie di 15 filtri in alluminio di opportuno spessore, montati su di una ruota (un *revolver* anch'esso in alluminio) gestita completamente in automatico; al passaggio attraverso uno dei filtri, la radiazione X verrà esponenzialmente attenuata secondo la legge di Lambert:

$$\Phi = \Phi_0 \cdot \exp[-\mu(E)x] \qquad \text{[III.3]}$$

dove Φ è il flusso della radiazione uscente (in fotoni/cm^2·s), Φ_0 è il flusso sulla sezione d'ingresso del filtro, $\mu(E)$ è il coefficiente d'assorbimento lineare dell'alluminio per un fascio di energia E, e x è lo spessore del filtro.

3.7.2 Risultati sperimentali

In base alla teoria di Rose [24], la fotocorrente complessiva si può sempre calcolare dall'integrale (su tutto lo spessore d del dispositivo) del prodotto fra l'efficienza di generazione η_g e l'efficienza di raccolta η_c delle cariche:

$$I_{ph}(E,V) = q\Phi_0 \int_0^d \eta_g(E,x)\eta_c(x,V)dx \qquad \text{[III.4]}$$

dove E è l'energia della radiazione incidente, Φ_0 è il suo flusso sulla superficie del campione, q è la carica dell'elettrone e V è la tensione di *bias*. L'efficienza di generazione η_g è data da:

$$\eta_g(E,x) = [1-R(E)]\alpha(E)\exp[-\alpha(E)x] \qquad \text{[III.5]}$$

Per valutare l'espressione dell'efficienza di raccolta, assumiamo che il contatto elettrico fra l'argento e il diamante policristallino sia ohmico (come evidenziato nel par. 3.4.2), e supponiamo che il campo elettrico sia uniforme lungo tutto lo spessore del campione; inoltre, assumendo che tutto il materiale attivo sia investito dai raggi X, trascurando fenomeni di trasporto per diffusione, la distanza di raccolta delle cariche fotogenerate è semplicemente data dalla lunghezza di *drift* $l_c = \mu\tau F$, dove $\mu\tau$ è il prodotto mobilità-

tempo di vita e F è il campo elettrico. Supponendo infine una dipendenza esponenziale della probabilità di raccolta dei portatori di carica generati ad una distanza x dalla superficie d'incidenza, l'efficienza di raccolta sarà data da:

$$\eta_c(x,V) = \frac{\mu\tau V}{x^2}\left[1-\exp\left(-\frac{x^2}{\mu\tau V}\right)\right] \qquad \text{[III.6]}$$

Sostituendo la [III.2] e la [III.3] nella [III.1], e integrando, si ottiene:

$$I_{ph}(E,V) = q\Phi_0\alpha(E)\frac{\mu\tau V}{d}\left[1-\exp\left(-\frac{d^2}{\mu\tau V}\right)\right] \qquad \text{[III.7]}$$

La fig. 3.25 mostra la tipica risposta di fotocorrente in seguito all'irradiazione con raggi X di energia 8.06 keV, per differenti intensità di radiazione ottenute variando lo spessore (da 0 a 0.22 cm) dei filtri attenuatori di alluminio:

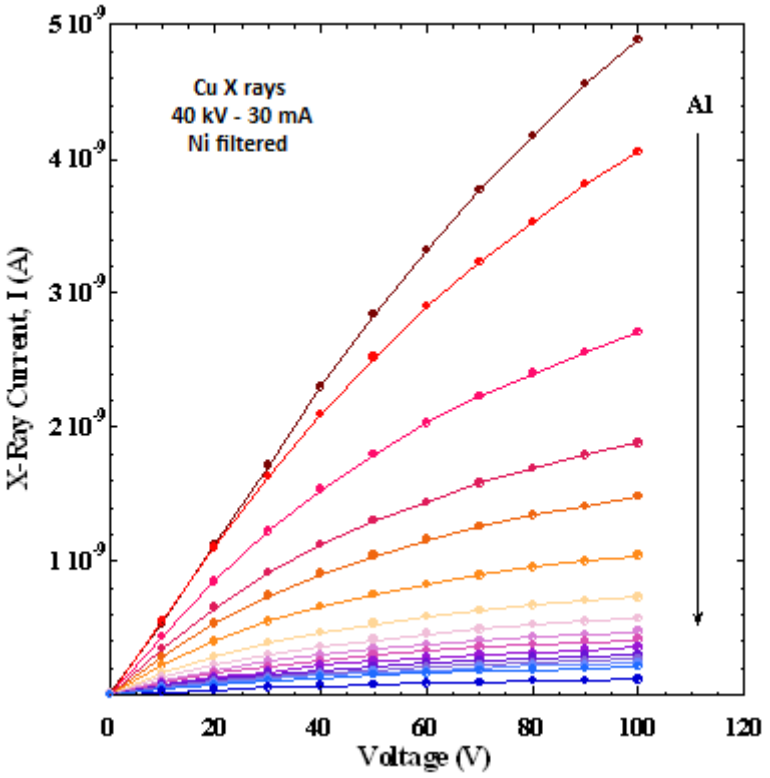


Fig. 3.25 *Fotocorrente vs. tensione di bias per diverse intensità di radiazione*

Si vede come la fotocorrente X aumenti in maniera proporzionale al campo elettrico per basse tensioni (100 V/cm), mentre un ulteriore aumento della tensione di *bias* porta ad una saturazione della fotocorrente. Questo andamento sperimentale è coerente con l'espressione teorica [III.7].
Poiché la sorgente X utilizzata non è monocromatica, all'aumentare dello spessore dei filtri di alluminio la radiazione K_α e la componente di *Bremsstrahlung* vengono attenuate con un diverso coefficiente lineare di assorbimento, rispettivamente 135 cm^{-1} e 6 cm^{-1}. Tale effetto è mostrato in fig. 3.26, dove è riportato l'andamento della fotocorrente X in funzione dello spessore dei filtri di alluminio (e per diverse tensioni di *bias*): è evidente la sovrapposizione di due regimi esponenziali. I coefficienti di assorbimento sono stati calcolati attraverso un *fit* delle curve sperimentali con l'espressione:

$$I_{ph} = I_D + I_{K\alpha} \exp(-\mu_{K\alpha} d) + I_B \exp(-\mu_B d) \qquad \text{[III.8]}$$

dove I_D è la corrente di buio, $I_{K\alpha}$ e I_B sono le componenti di fotocorrente dovute rispettivamente alla sola K_α a 8.06 keV e alla sola radiazione di *Bremsstrahlung*, mentre $\mu_{K\alpha}$ e μ_B sono i rispettivi coefficienti lineari di assorbimento.

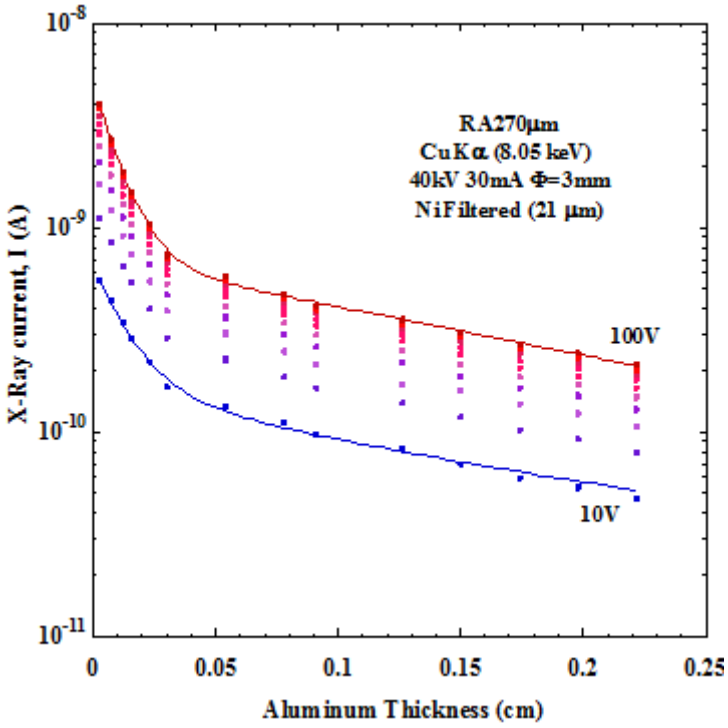


Fig. 3.26 *Risposta di fotocorrente per diverse intensità di radiazione (ottenute per filtraggio con alluminio)*

Mentre μ_B rimane costante stabilizzandosi attorno 6 ± 1 cm^{-1}, $\mu_{K\alpha}$ varia con l'intensità di radiazione da 74 a 108 cm^{-1}: questo comportamento suggerisce una risposta non-lineare del rivelatore alla radiazione Cu K_α . Supponendo la fotocorrente dipendente dall'intensità di radiazione (sempre in accordo con la teoria di Rose), la risposta del rivelatore al flusso Φ di raggi X può essere espressa come:

$$I_{ph} \propto \Phi^{\beta} \qquad \text{[III.9]}$$

dove $0.5 < \beta < 1$ è un parametro indicativo del regime di ricombinazione: $\beta = 0.5$ è associato ad un meccanismo di ricombinazione bimolecolare tipico dei semiconduttori privi di difetti, mentre $\beta = 1$ si ottiene per processi di ricombinazione monomolecolare, quando cioè gli stati interni alla *gap* partecipano alla ricombinazione. Nel nostro caso, il coefficiente β può essere calcolato come il rapporto fra il $\mu_{K\alpha}$ trovato sperimentalmente e il suo valore teorico 135 cm^{-1}: questo porta ad un valore massimo di β pari a 0.8, e dunque ad una risposta leggermente sub-lineare del rivelatore alla radiazione Cu K_α .
Una volta isolato, il contributo della sola K_α alla fotocorrente è stato utilizzato per valutare il prodotto mobilità-tempo di vita, facendo riferimento all'espressione [III.7].

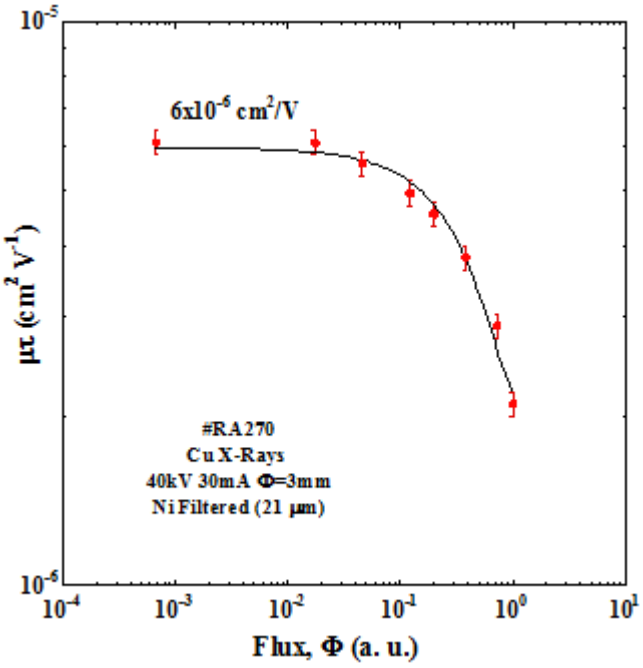


Fig. 3.27 *Prodotto mobilità-tempo di vita vs. flusso di radiazione X*

La fig. 3.27 riporta il risultato del *fit*. Per bassi *rate* di generazione il prodotto $\mu\tau$ rimane costante attorno ad un valore di $6 \cdot 10^{-6}$ cm^2/V, mentre aumentando il flusso di radiazione si osserva un decremento esponenziale: un comportamento di questo tipo può essere interpretato supponendo che per alti *rate* di generazione (elevata densità di cariche

fotogenerate) intervengano fenomeni di *scattering* elettrone-elettrone che riducono drasticamente sia la mobilità delle cariche che il loro tempo di vita (sarà allora necessario applicare un campo elettrico più intenso per migliorare la raccolta delle cariche).

3.8 Misure di spettroscopia a raggi X

Obiettivo di queste misure (vedi fig. 3.28) è la rivelazione del singolo fotone X, al fine di fornire uno spettro di energia della radiazione incidente e dimostrare la possibilità di utilizzare rivelatori in diamante in applicazioni spettroscopiche.

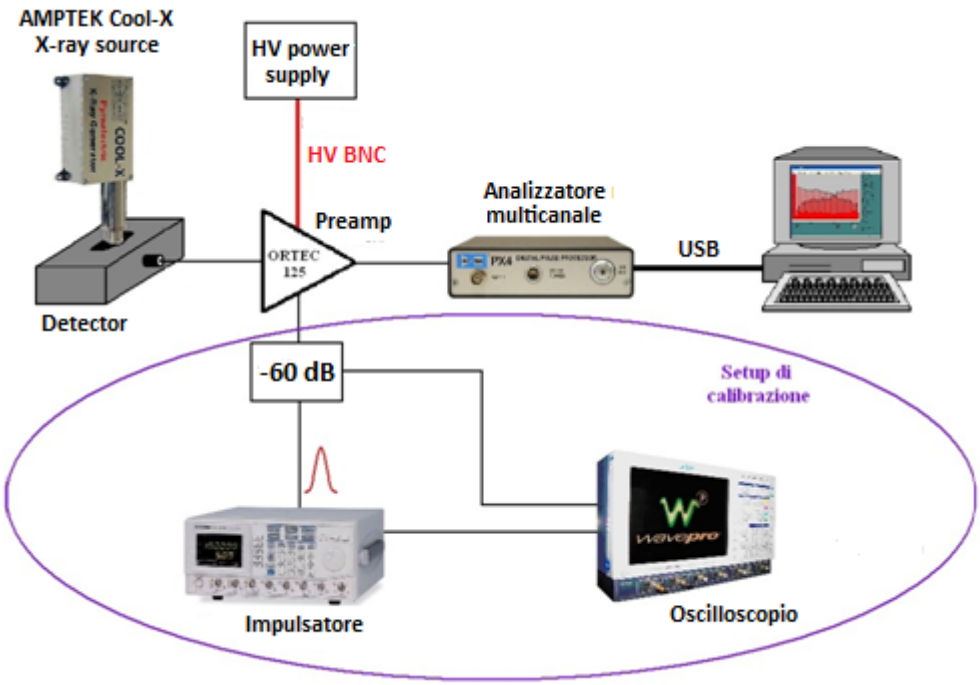


Fig. 3.28 *Setup di misura per la rivelazione del singolo fotone X*

Il singolo fotone X (prodotto dalla sorgente AMPTEK *Cool-X*) viene assorbito dal diamante del rivelatore, e la sua energia viene convertita in un certo numero di coppie elettrone-lacuna, che vengono poi raccolte dagli elettrodi sotto l'azione di un campo elettrico sufficientemente intenso (per questo occorre applicare una tensione di *bias* fino a 500 V, e a tale scopo si è utilizzato un ORTEC 556 *High Voltage Power Supply*); un preamplificatore di carica (ORTEC 125) converte la quantità di carica raccolta in un valore di tensione, producendo in uscita un impulso di ampiezza direttamente proporzionale all'energia del fotone X assorbito. Tale impulso viene poi elaborato e

digitalizzato da un analizzatore multicanale (AMPTEK PX4), che opera sostanzialmente come un discriminatore a soglia: il cuore dell'analizzatore multicanale è infatti un convertitore A/D, che converte l'ampiezza dell'impulso in ingresso in un numero (canale) ad essa proporzionale. Associata al convertitore A/D c'è una memoria costituita da tante locazioni quanti sono i canali: in seguito all'arrivo di un impulso nel canale *n* (ovvero un impulso la cui ampiezza è stata convertita nel valore numerico *n*), il contenuto della locazione di memoria corrispondente al canale *n* viene incrementato di un'unità. Dopo un certo tempo di acquisizione di eventi, dunque, la visualizzazione dei contenuti di ogni locazione di memoria (conteggi per canale) consente di ottenere uno spettro di ampiezza d'impulsi (PHA = *Pulse Height Analysis*) che può essere interpretato come lo spettro di energia della radiazione X, in quanto si ha una diretta corrispondenza fra l'energia del singolo fotone X, la quantità di carica raccolta, l'ampiezza dell'impulso in uscita dal preamplificatore di carica e il numero (canale) in cui questo impulso viene digitalizzato.
La catena di misura ha bisogno di un'opportuna calibrazione, al fine di trovare la relazione che sussiste fra canale e ampiezza dell'impulso, e di verificarne la linearità: a tale scopo è stato allestito un *setup* di calibrazione costituito da un impulsatore (GW INSTEK GFG-3015), un attenuatore, e un oscilloscopio digitale (LeCroy Wavepro 960, vedi par 3.6.4) per il monitoraggio della forma d'onda prodotta e di quella attenuata. L'impulsatore ha la funzione di produrre un impulso che simuli l'uscita del preamplificatore di carica in corrispondenza dell'arrivo di un fotone, impulso che (nell'ipotesi di completa raccolta delle cariche) raggiunge un ampiezza massima di qualche decina di μV: visto che l'impulsatore usato produce segnali di ampiezza minima di decine di mV, si è reso necessario l'utilizzo di un attenuatore (realizzato nel nostro laboratorio) con fattore di attenuazione di -60dB.

3.8.1 Sorgente Amptek *Cool-X*

Il *Cool-X* della AMPTEK è un generatore di raggi X in miniatura che non utilizza radioisotopi o tubi a raggi X ad alta potenza: è un sistema a stato solido che genera raggi X quando un cristallo piroelettrico, posto all'interno di una camera riempita di un gas a bassa pressione (fig. 3.29) viene sottoposto a cicli di riscaldamento e raffreddamento [37]. In fase di riscaldamento, il cristallo piroelettrico esibisce una spontanea diminuzione di polarizzazione, e dunque all'aumentare della temperatura si sviluppa un campo elettrico al suo interno: per una specifica orientazione del cristallo, la superficie si carica positivamente ed attrae elettroni dal gas a bassa pressione, i quali colpiscono la superficie del cristallo producendo raggi X caratteristici del tantalio (emissione principale a 8.14 keV) e radiazione di *Bremsstrahlung* (fig. 3.30). In fase di raffreddamento, la polarizzazione del cristallo aumenta, e gli elettroni sono accelerati dalla superficie del cristallo verso un *target* di rame, producendo raggi X caratteristici del rame (emissione principale a 8.05 keV) e ancora una volta radiazione di *Bremsstrahlung* (fig. 3.30). Una finestra di berillio permette la trasmissione dei raggi X prodotti, mentre un sistema di controllo consente di monitorare la temperatura del cristallo e di passare automaticamente dalla fase di riscaldamento a quella di raffreddamento.

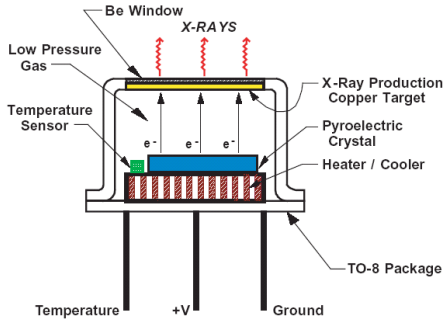


Fig. 3.29 *Struttura della sorgente AMPTEK Cool-X*

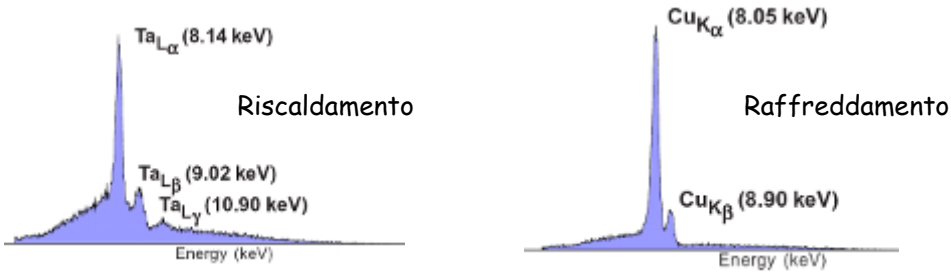


Fig. 3.30 *Spettri di emissione della sorgente nelle due fasi di riscaldamento e raffreddamento*

La fig. 3.31 mostra un tipico andamento temporale del flusso di uscita della sorgente *Cool-X* per diverse fasi di riscaldamento e raffreddamento [38]: si hanno cicli di 3 minuti in cui le fasi si alternano, e possiamo notare come il flusso possa variare da ciclo a ciclo, e inoltre come si abbia un'emissione leggermente più intensa in fase di riscaldamento. Il flusso massimo ottenibile dalla sorgente è di 10^8 fotoni al secondo, l'equivalente di una sorgente radioattiva di attività 2 mCi.

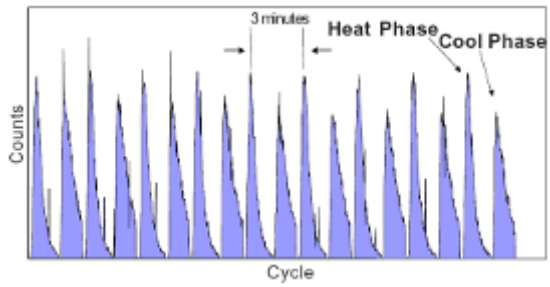


Fig. 3.31 *Tipico flusso di uscita della sorgente AMPTEK Cool-X*

3.8.2 ORTEC 556 *High Voltage Power Supply*

Il generatore di alta tensione serve a polarizzare il rivelatore per un'adeguata raccolta delle cariche fotogenerate. E' stato utilizzato un ORTEC 556 *High Voltage Power Supply*, che consente di ottenere una tensione di uscita regolabile in maniera continua da ± 10 V a ± 3000 V (una manopola sul retro del generatore consente di selezionare la polarità), con una corrente di carico compresa fra 0 e 10 mA. Il rumore sull'uscita è inferiore ai 10 mV picco-picco, e questo assicura un'ottimale stabilità della tensione. L'ORTEC 556 è dotato di un pannello di controllo sul quale un *display* consente di monitorare la tensione o la corrente di uscita; la tensione d'uscita è regolabile attraverso uno *switch* a 6 posizioni (con passo 500 V), uno *switch* a 5 posizioni (con passo 100 V) e un potenziometro di precisione da 0 a 100 V (la tensione voluta si ottiene dalla somma delle tre impostazioni). La tensione di polarizzazione è fornita al rivelatore attraverso il preamplificatore di carica: un cavo SHV collega l'uscita dell'ORTEC 556 ad un apposito ingresso di *bias* del preamplificatore ORTEC 125 (che accetta fino a ± 1.5 kV di *bias*), il quale provvederà poi a fornire l'alta tensione (attraverso una resistenza di carico di 100 MΩ) al connettore d'ingresso collegato al rivelatore.

3.8.3 Preamplificatore di carica ORTEC 125

Il preamplificatore di carica (CSA, *Charge Sensitive Preamplifier*) converte la quantità di carica prodotta dall'arrivo del singolo fotone X (e raccolta dagli elettrodi del rivelatore) in un valore di tensione ad essa proporzionale. Lo schema circuitale semplificato di un preamplificatore di carica è riportato in fig. 3.32:

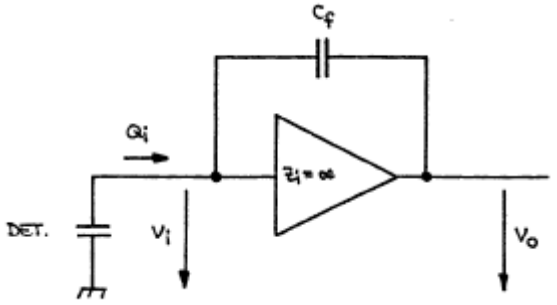


Fig. 3.32 *Schema circuitale semplificato di un preamplificatore di carica*

In questo schema il rivelatore è rappresentato da una capacità in ingresso al preamplificatore, il quale è costituito da un amplificatore operazionale (in configurazione invertente) con una capacità in controreazione (capacità di *feedback* C_f). Nell'ORTEC 125, abbiamo C_f = 3.9 pF .
La tensione V_i in ingresso al preamplificatore può essere espressa come:

$$V_i = \frac{Q_i}{C_i} \qquad \text{[III.10]}$$

dove Q_i è la carica in ingresso al preamplificatore e C_i è la cosiddetta capacità dinamica d'ingresso (*dynamic input capacitance*) del preamplificatore, ovvero la capacità effettivamente vista dal rivelatore. Per il calcolo di C_i dobbiamo tener presente che la tensione ai capi della capacità C_f è data da:

$$V_o = C_f (A+1) V_i \qquad \text{[III.11]}$$

dove A è il guadagno a circuito aperto (in genere maggiore di 10^4) dell'operazionale.
La carica Q_f che viene trasferita sulla capacità di *feedback* è:

$$Q_f = C_f V_0 = C_f (A+1) V_i \qquad \text{[III.12]}$$

Nell'ipotesi di impedenza d'ingresso infinita ($Z_i = \infty$) dell'operazionale abbiamo $Q_i = Q_f$, pertanto:

$$C_i = \frac{Q_i}{V_i} = \frac{Q_f}{V_i} = C_f (A+1) \qquad \text{da cui:} \qquad V_i = \frac{Q_i}{C_f (A+1)} \qquad \text{[III.13]}$$

Essendo $V_o = -AV_i$ si ottiene alla fine il valore della tensione d'uscita:

$$V_o = -\frac{A}{A+1}\frac{Q_i}{C_f} \approx -\frac{Q_i}{C_f} \qquad \text{nell'ipotesi } A >> 1 \qquad \text{[III.14]}$$

Ecco dunque dimostrato come la tensione d'uscita sia direttamente proporzionale alla carica in ingresso al preamplificatore. Naturalmente Q_i non corrisponde esattamente alla carica Q_{det} raccolta dagli elettrodi del rivelatore, ma ad una frazione di essa; più precisamente:

$$\frac{Q_i}{Q_{\text{det}}} = \frac{C_i}{C_i + C_{\text{det}}} = \frac{1}{1+\frac{C_{\text{det}}}{C_i}} \qquad \text{[III.15]}$$

dove C_{det} è la capacità del rivelatore. Nel nostro caso, la C_i dell'ORTEC 125 è ≥ 50 nF, mentre la capacità del rivelatore è di circa 15 pF: dunque $C_{det} << C_i$ e si può pertanto ritenere che praticamente il 100% della carica raccolta venga trasferito all'ingresso del preamplificatore.
Il guadagno di conversione dell'ORTEC 125 è di 10 mV/MeV per rivelatori in silicio, guadagno che si riduce a 2.8 mV/MeV per rivelatori in diamante (a causa della più elevata energia per la creazione di una coppia elettrone-lacuna): questo vuol dire che l'arrivo sul nostro rivelatore di un fotone di energia 1 MeV, nell'ipotesi di totale raccolta delle cariche fotogenerate, produce in uscita dall'ORTEC 125 un gradino di tensione di

ampiezza pari a 2.8 mV. Il tempo di salita del gradino è inferiore ai 20 ns (per rivelatori di capacità inferiore ai 100 pF).
L'ORTEC 125 è un preamplificatore di carica di tipo *pulsed-reset*. L'interazione con un fotone X genera in uscita un gradino di tensione, dopodiché l'uscita rimane costante, non avendo la capacità di *feedback* alcuna possibilità di scaricarsi; pertanto, ogni susseguente impulso di corrente in ingresso (corrispondente alla raccolta di una certa quantità di carica Q_i prodotta da un successivo fotone X) incrementerà ulteriormente la tensione di uscita di un altro gradino di tensione proporzionale a Q_i, cosicché la forma d'onda d'uscita del preamplificatore assumerà nel tempo l'aspetto di una gradinata (vedi fig. 3.33a) fino a raggiungere il limite superiore (o inferiore) del *range* dinamico di uscita del preamplificatore (± 4 V per l'*ORTEC 125*): a questo punto viene attivato da un comparatore un opportuno circuito di *reset*, che permette di scaricare la capacità di *feedback* e di riportare a zero la tensione di uscita.
Esiste un'altra famiglia di preamplificatori di carica, i *resistive-feedback* CSA, che non prevedono circuiti di *reset*: la capacità di *feedback* si scarica attraverso una resistenza R_f posta in parallelo, e dunque in uscita non si avrà un gradino di tensione, bensì un impulso con una lunga coda esponenziale (vedi fig. 3.33b).

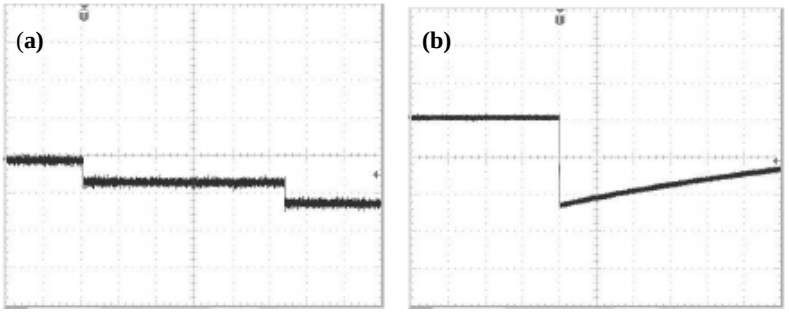


Fig. 3.33 *Tipiche forme d'onda di uscita di preamplificatori di carica: pulsed-reset (a) e resistive-feedback (b)*

Naturalmente, in entrambi i casi i segnali d'uscita del preamplificatore di carica dovranno essere sottoposti prima di tutto ad un opportuno filtraggio analogico, al fine di estrarre le informazioni relative ai singoli eventi (i singoli gradini nel caso *pulsed-reset* e i singoli impulsi nel caso *resistive-feedback*): questo verrà esaminato più dettagliatamente nel prossimo paragrafo.

3.8.4 AMPTEK PX4 *Digital Pulse Processor* (analizzatore multicanale)

L' AMPTEK PX4 *Digital Pulse Processor* è lo strumento principale del *setup* di misura per la rivelazione del singolo fotone X: riceve in ingresso l'uscita del preamplificatore di carica, la digitalizza, esegue un *processing* digitale in tempo reale del segnale, ne rivela digitalmente l'ampiezza di picco e genera un istogramma che rappresenta la distribuzione di ampiezza d'impulso (PHA, *Pulse Height Analysis*) relativa alla

radiazione incidente, ovvero il suo spettro d'energia. Lo spettro è poi trasmesso attraverso un'interfaccia seriale ad un PC. In fig. 3.34 è riportato lo schema a blocchi del PX4. Possiamo individuare due sezioni: un circuito di elaborazione del segnale e una circuiteria di alimentazione (che non verrà descritta in questa sede). Il circuito di elaborazione prevede sei blocchi principali: pre-filtraggio analogico, ADC, formatore digitale d'impulsi, logica di selezione degli impulsi, memoria (*histogram logic*), interfacciamento *hardware* e *software* (microcontrollore + interfaccia seriale).

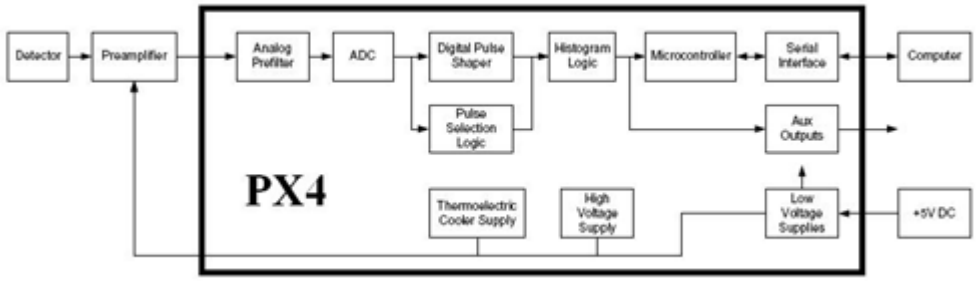


Fig. 3.34 *Schema a blocchi dell'AMPTEK PX4 Digital Pulse Processor*

Passeremo ora a descrivere i singoli blocchi principali.
Il pre-filtraggio analogico, del quale vediamo lo schema in fig. 3.35, prepara il segnale d'uscita del preamplificatore per un'accurata digitalizzazione. Tale segnale è caratterizzato tipicamente da un'ampiezza molto piccola (inferiore al mV), un tempo di salita estremamente basso (< 20 ns nell'ORTEC 125), ed è un gradino di tensione (CSA *pulsed-reset*) o un impulso seguito da una lunga coda esponenziale (CSA *resistive-feedback*); inoltre ad esso è sovrapposto un *offset* variabile nel *range* ± 4V e un significativo contributo di rumore bianco.

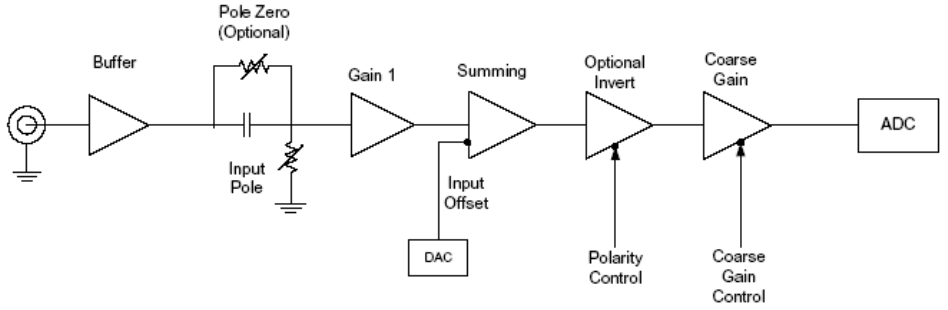


Fig. 3.35 *Schema a blocchi del circuito di condizionamento del segnale analogico prima della conversione A/D*

Il sistema di pre-filtraggio analogico rimuove il largo *offset* e le componenti di rumore di bassa frequenza attraverso un filtro passa-alto CR (in parallelo alla capacità c'è una resistenza detta di *pole-zero*: nel caso di ingressi a gradinata tale resistenza è disconnessa, mentre nel caso di uscite di CSA *resistive-feedback* serve a ridurre

drasticamente la coda esponenziale), amplifica il segnale in due fasi successive (*Gain 1 e Coarse Gain*), lo inverte se necessario (l'ingresso dell'ADC deve infatti essere positivo) e gli sovrappone un *offset* opportuno affinché l'ADC lavori sempre all'interno del suo *range* dinamico d'ingresso (0 ÷ 2 V).
L'ADC digitalizza l'impulso in uscita dal sistema di pre-filtraggio analogico con frequenza 20 MHz, e il flusso di valori digitalizzati viene inviato in tempo reale al formatore digitale d'impulsi, il quale conferisce all'impulso una forma trapezoidale (vedi fig. 3.36): l'impulso viene cioè allargato (per migliorare il rapporto segnale-rumore) e appiattito in corrispondenza del picco (per facilitare la misura dell'ampiezza di picco da parte di un circuito di *peak detecting* interno al formatore d'impulsi). Il valore digitalizzato della tensione di picco viene inviato successivamente ad una memoria, e il contenuto della locazione di memoria (ovvero il "canale") corrispondente a tale valore verrà incrementato di una unità: la memoria agisce dunque come un *set* di contatori indipendenti, ciascuno operante in uno stretto *range* di tensioni (e dunque di energie se pensiamo alla corrispondenza tensione-energia); i conteggi vengono accumulati all'interno dei canali (che possono essere 256, 512, 1024, 2048, 4096 o 8192 a seconda della risoluzione che abbiamo impostato per l'ADC), producendo alla fine del tempo di acquisizione un istogramma che rappresenta lo spettro di energia della radiazione incidente sul rivelatore.

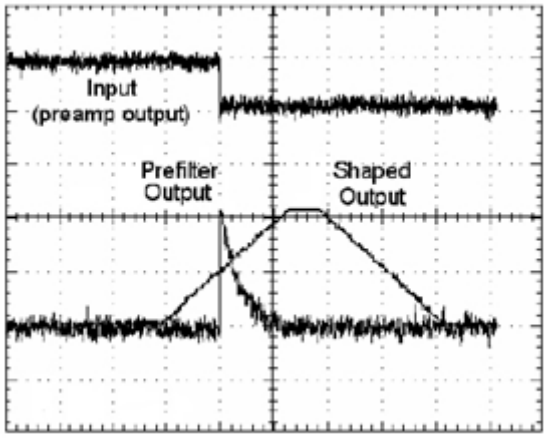


Fig. 3.36 *Processing di un gradino di tensione in ingresso al PX4*

In parallelo al formatore d'impulsi lavora una logica di selezione d'impulsi, il cui compito è quello di "scartare" gli impulsi per i quali non può essere fatta un'accurata elaborazione. Tale logica può effettuare ad esempio la cosiddetta *Pile-Up Rejection* (PUR), ovvero può escludere dall'analisi due impulsi che arrivino quasi contemporaneamente (gli impulsi andrebbero a sovrapporsi, e questo produrrebbe informazioni errate sull'energia del singolo fotone). La logica di selezione permette anche di discriminare gli impulsi d'ingresso in base al loro tempo di salita (RTD, *Rise-Time Discrimination*); se all'interno del materiale del rivelatore ci sono regioni in cui il campo elettrico è più debole, le cariche fotogenerate raggiungeranno più lentamente e in numero minore gli elettrodi, producendo un impulso più lento e di ampiezza minore.

Impulsi di questo tipo possono provocare distorsioni spettrali (picchi "ombra", picchi asimmetrici, etc.), e vanno per questo motivo esclusi dall'analisi: a questo provvede appunto la logica di selezione attraverso la RTD.

3.8.5 Impulsatore GW INSTEK GFG-3015

Il GW INSTEK GFG-3015 è un generatore di forma d'onda programmabile, ed è stato utilizzato nella rivelazione del singolo fotone X per la calibrazione dell'analizzatore multicanale, al fine di trovare una corrispondenza fra canale e tensione d'uscita del preamplificatore di carica. Il GFG-3015 permette di generare forme d'onda sinusoidali, ad onda quadra e ad onda triangolare, di frequenza compresa fra 0.01 Hz e 15 MHz (con accuratezza ± 0.02%) e di ampiezza picco-picco compresa fra 10 mV e 10 V (con accuratezza ≤ 3 % per frequenze comprese fra 10 Hz e 1 MHz e accuratezza ≤ 10 % per frequenze comprese fra 1 MHz e 15 MHz). E'inoltre possibile dare un *offset* alla forma d'onda generata compreso fra -5 V e 5 V, con risoluzione 10 mV e accuratezza ≤ 3 %.
Per testare il preamplificatore di carica e calibrare l'analizzatore multicanale, occorre generare un impulso di *test* con un breve tempo di salita (inferiore ai 20 ns) e un tempo di discesa molto lungo (maggiore di 100 μs), così da apparire come un gradino di tensione: per questo abbiamo scelto come segnale di *test* un'onda quadra (per la quale il GFG-3015 assicura un tempo di salita < 18 ns) di frequenza 500 Hz, ampiezza picco-picco variabile fra 100 mV e 1 V, e *offset* pari alla metà di tale ampiezza (per far sì che la tensione di uscita sia nulla per metà periodo).

3.8.6 Rete di attenuazione

Il *range* di tensioni che ci aspettiamo in uscita dal preamplificatore di carica in corrispondenza dell'arrivo di una radiazione dell'ordine dei keV va da poche decine a centinaia di μV, mentre l'impulsatore produce forme d'onda di ampiezza maggiore di almeno due ordini di grandezza. Per questo è stato posto all'uscita dell'impulsatore un attenuatore puramente resistivo con coefficiente di attenuazione pari a -60dB. In fig. 3.37 è riportato lo schema circuitale dell'attenuatore progettato e realizzato in laboratorio:

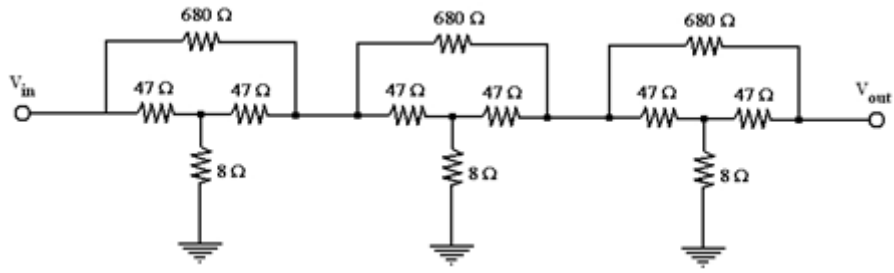


Fig. 3.37 *Schema circuitale della rete di attenuazione a -60dB*

L'attenuatore è di tipo *bridge T* ed è composto di tre stadi, ciascuno con coefficiente di attenuazione -20dB; le resistenze da 47 Ω servono per adattare l'impedenza d'ingresso e

quella d'uscita ai cavi BNC da 50 Ω utilizzati nel *setup* di misura. Al fine di ridurre il più possibile l'effetto di sorgenti esterne di rumore l'attenuatore è stato realizzato all'interno di una scatola metallica, sulla quale sono stati successivamente inseriti due connettori BNC.

3.8.7 Risultati sperimentali

La rivelazione del singolo fotone X consente di poter analizzare lo spettro di energia della radiazione incidente attraverso la *Pulse Height Analysis* (*PHA*). Nei successivi paragrafi verrà dapprima descritta la procedura di calibrazione dell'analizzatore multicanale (al fine di valutare la corrispondenza canale-tensione-energia) e in seguito verranno illustrati i risultati sperimentali relativi alla PHA di una radiazione X di energia media 8.1 keV, con particolare riferimento alla risoluzione in energia (indice della sua efficacia nel rivelare fotoni X di diversa energia).

3.8.7.1 Calibrazione dell'analizzatore multicanale

L'operazione di calibrazione dell'analizzatore multicanale (MCA, *Multi-Channel Analyzer*) parte con una procedura di *test* [39] del preamplificatore di carica (CSA) ORTEC 125: il *test* consiste nell'iniettare nel CSA una specifica quantità di carica e misurare la corrispondente ampiezza del gradino di tensione in uscita, secondo lo schema riportato in fig. 3.38:

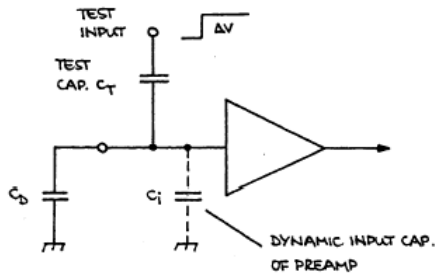


Fig. 3.38 *Procedura di test del preamplificatore di carica*

L'ingresso di *test* dell'ORTEC 125 termina su una capacità di *test* C_T = 2pF: ponendo in ingresso un gradino di tensione di ampiezza ΔV (generato dal sistema impulsatore + attenuatore), sulla capacità di *test* si depositerà una quantità di carica di *test* (nell'ipotesi verificata $C_i >> C_T$) pari a $Q_T = C_T \Delta V$. Tale quantità di carica verrà convertita dal CSA in un nuovo gradino di tensione, la cui ampiezza è valutabile conoscendo il guadagno di conversione del CSA (2.8 mV/MeV nel diamante), e con la quale andremo a calibrare il multicanale per trovare la corrispondenza canale-tensione.
In fig. 3.39 appare la schermata del multicanale al termine della procedura di calibrazione: ciascuno dei picchi è centrato su un canale corrispondente ad uno strettissimo intervallo di tensioni di *test* in uscita dal preamplificatore di carica. I picchi sono distanti in tensione di 20 μV, ed è stato coperto un *range* che va da 40 μV (minima tensione rivelabile a destra della spalla di rumore, introdotto quasi completamente dal sistema impulsatore + attenuatore) a 440 μV, fino alla saturazione del multicanale. La

spalla di rumore occupa circa 500 canali, ma in fase di analisi ai raggi X, grazie alla rimozione dalla catena di misura dell'impulsatore e dell'attenuatore, si riduce a circa 100 canali, consentendo così di rivelare tensioni di uscita dall'ORTEC 125 inferiori ai 40 μV.

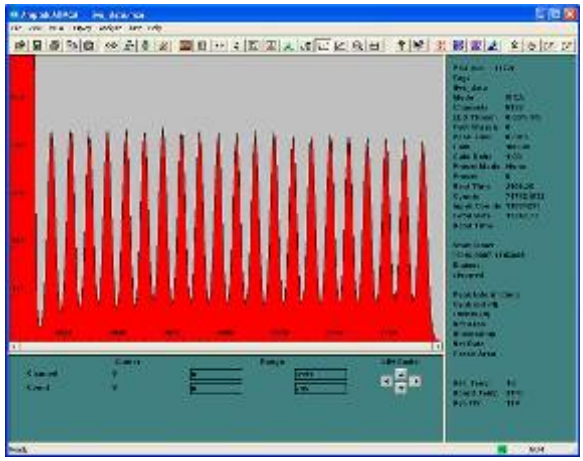


Fig. 3.39 *Display dell'AMPTEK PX4 al termine della procedura di calibrazione*

Dopo questa fase di acquisizione, il *file* di calibrazione è stato elaborato con l'utilizzo del software Kaleidagraph 4.0, al fine di costruire una curva di calibrazione in cui sono riportati i "centroidi" dei picchi (ovvero i canali su cui sono centrati i picchi) in funzione delle corrispondenti tensioni di *test* in uscita dal CSA. La curva di calibrazione (fig. 3.40) è una retta, che conferma quindi la relazione di linearità che sussiste fra canali e tensioni: la pendenza della retta è stata valutata in 18 ± 1 canali/μV (l'incertezza sul numero di canali è dovuta all'accuratezza del sistema impulsatore + attenuatore).

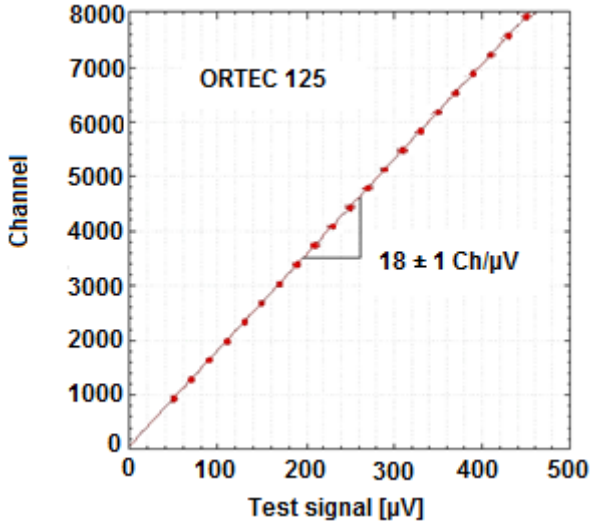


Fig. 3.40 *Retta di calibrazione dell'analizzatore multicanale*

3.8.7.2 Pulse Height Analysis (PHA)

Una volta completata la procedura di calibrazione dell'analizzatore multicanale, si è passati alla caratterizzazione finale ai raggi X, utilizzando per l'MCA le stesse impostazioni della fase di calibrazione. Nei singoli *run* di misura il rivelatore, opportunamente polarizzato con tensioni di *bias* fino a 500 V, è stato sottoposto per 30 minuti all'azione della radiazione X (successivi cicli di riscaldamento e raffreddamento della sorgente AMPTEK *Cool-X*, per un'energia media di emissione pari a 8.1 keV). In base al guadagno di conversione dell'ORTEC 125 (2.8 mV/MeV nel diamante), un singolo fotone X di energia 8.1 keV, nell'ipotesi di completa raccolta delle cariche, produce in uscita dal preamplificatore di carica un gradino di tensione di ampiezza ~23 μV. Consultando la retta di calibrazione, e tenendo debitamente conto dell'incertezza sulla pendenza, questo valore di tensione in ingresso al multicanale dovrebbe corrispondere ad un picco il cui "centroide" è compreso fra il canale 491 e il canale 537: questa previsione è confermata dallo spettro di energia (*Pulse Height Analysis*) della radiazione X, che compare in fig. 3.41.

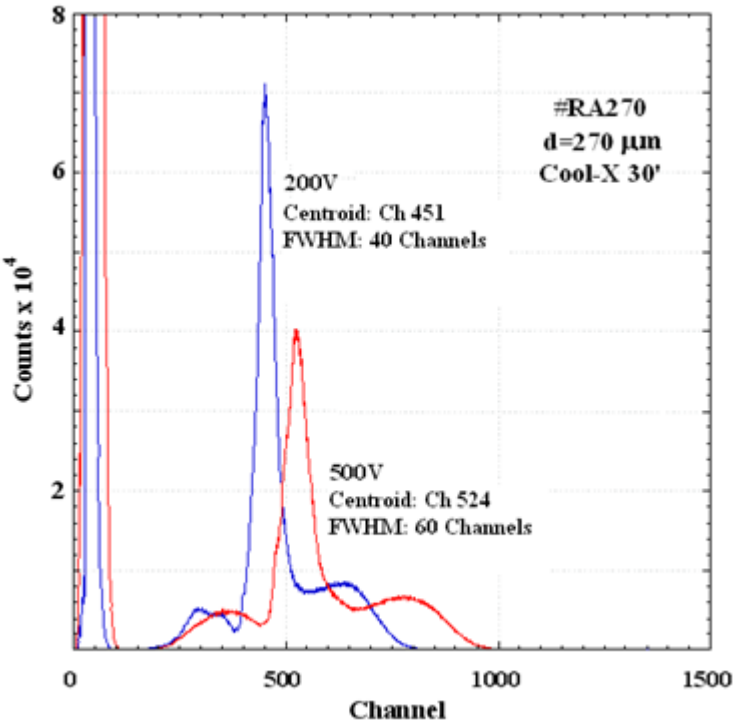


Fig. 3.41 *Pulse Height Analysis (spettro di energia) della radiazione X*

Come si può constatare, con una tensione di *bias* di 500 V il picco corrispondente a 8.1 keV è centrato sul canale 524, dunque all'interno dell'intervallo di canali previsto dalla retta calibrazione. I lobi laterali osservabili alla base del picco sono relativi ai contributi del *Bremsstrahlung* e delle righe caratteristiche secondarie della radiazione X (che, come evidenziato nel par. 3.8.1, non proviene da una sorgente monocromatica). La

larghezza a mezza altezza (FWHM) del picco a 500 V è di 60 canali: questo vuol dire che la risoluzione in energia a 8.1 keV, definita dal rapporto FWHM /centroide, è pari a 0.11 (ovvero circa 900 eV). Applicando una tensione di *bias* di 200 V, invece, il picco corrispondente a 8.1 keV è centrato sul canale 451, ovvero al di sotto del limite inferiore dell'intervallo di canali previsto dall'analisi di calibrazione: tale analisi è stata però condotta nell'ipotesi di totale raccolta delle cariche fotogenerate, e dunque se ne deduce che a 200 V si ha una minore efficienza di raccolta, dovuta molto verosimilmente ad un campo elettrico non abbastanza intenso. La FWHM del picco a 200 V è di 40 canali, da cui si ottiene una risoluzione in energia a 8.1 keV pari a 0.09 (ovvero circa 730 eV), leggermente migliore rispetto a quella ottenuta a 500 V (forse per una minore corrente superficiale di perdita nel rivelatore): la migliore risoluzione a 200 V spiega anche il maggior numero di conteggi in corrispondenza del centroide del picco, numero che comunque dipende fortemente anche dal flusso di radiazione X sul rivelatore, variabile da ciclo a ciclo. In ogni caso, per entrambe le tensioni di polarizzazione si ottiene una risoluzione in energia inferiore al keV, un risultato in linea con le prestazioni tipiche dei rivelatori a semiconduttore in silicio, e nettamente migliore rispetto alle risoluzioni ottenibili con i rivelatori a scintillazione [40]. A ulteriore conferma delle buone prestazioni del rivelatore in termini di sensibilità al singolo fotone, è stata valutata la risposta del dispositivo ai raggi X emessi nelle singole fasi di riscaldamento e raffreddamento del *Cool-X* (ciascuna della durata di 90 secondi) all'interno di un unico ciclo (fig. 3.42).

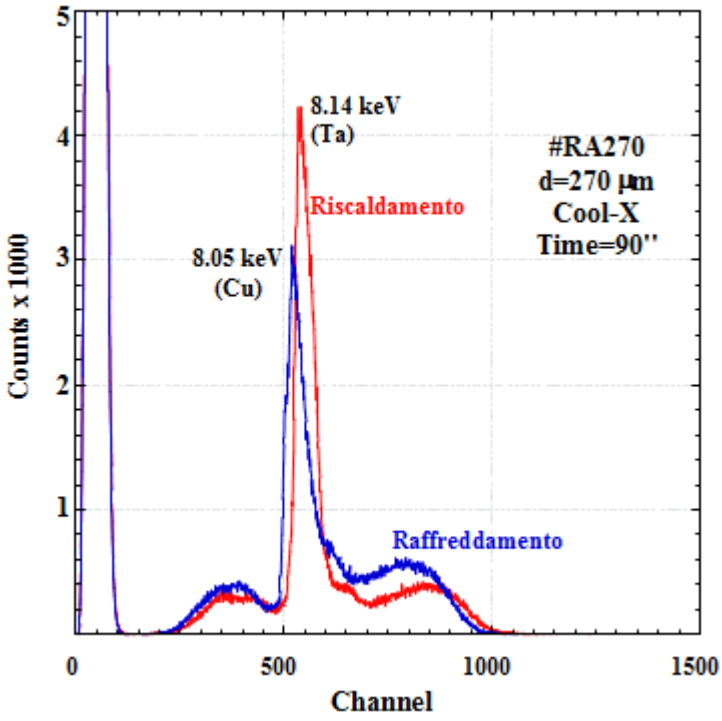


Fig. 3.42 *Spettri di energia delle radiazioni emesse dal Cool-X nelle due fasi di funzionamento*

La tensione di *bias* applicata al rivelatore è ancora di 500 V, per assicurare la massima efficienza di raccolta. Dalla fig. 3.42 è possibile apprezzare come, seppure in due fasi di monitoraggio successive (in quanto siamo ben al di sotto del limite di risoluzione), il rivelatore sia in grado di discriminare in energia le due righe caratteristiche principali del rame (Cu K_α a 8.05 keV) e del tantalio (Ta L_α a 8.14 keV), due radiazioni che differiscono in energia di soli 90 eV. Il maggior numero di conteggi registrato per il picco associato agli 8.14 keV della radiazione Ta L_α è dovuto alle caratteristiche di emissione della sorgente *Cool-X*, che genera in fase di riscaldamento una radiazione mediamente più intensa di quella generata in fase di raffreddamento.

Capitolo IV

Progettazione e realizzazione di un sistema elettronico multicanale per l'acquisizione e il condizionamento di deboli segnali di corrente

4.1 Introduzione

L'idea di un'elettronica di condizionamento per segnali in corrente di debole intensità provenienti da campioni di diamante sotto radiazione è precedente al presente lavoro di tesi di dottorato. All'interno del laboratorio S2DEL, nel quale ho svolto anche la tesi di laurea, si lavorava già per caratterizzare campioni di diamante sotto irraggiamento UV, e si pensava anche ad un'applicazione dei sensori quali dosimetri per radiazioni ionizzanti: dovendo continuamente effettuare misure in diversi laboratori, compiendo dunque spostamenti, l'idea di un'applicazione portatile e compatta cominciò a prendere piede. Fino alla realizzazione del sistema da me progettato, l'unico modo per effettuare misure di questo tipo era l'utilizzo di strumenti da banco come il *lock-in* (per misure in alternata) o l'elettrometro Keithley 6517A (per misure in continua), entrambi indubbiamente ingombranti in termini di portabilità.
Il primo problema fu quello di disporre di un segnale in tensione proporzionale alla corrente fotogenerata: una tensione si prestava meglio alla conversione in digitale, il che avrebbe permesso una più agevole elaborazione e memorizzazione dei dati acquisiti. L'elettronica di *front-end* doveva dunque, come minimo, essere costituita da un convertitore corrente-tensione (un transimpedenza). Per bassi valori di corrente, il transimpedenza da realizzare avrebbe dovuto avere un resistore di reazione di valore molto elevato (10 ÷ 100 GΩ per i nostri scopi), il quale, considerando le non idealità dei componenti, si sarebbe in sostanza comportato come un condensatore: dunque il transimpedenza diventò un integratore. Volendo integrare correnti dell'ordine dei pA, serviva un amplificatore operazionale con correnti di *bias* dell'ordine dei fA [41, 42], dunque un elettrometro. Non sono molti i produttori di elettrometri integrati, ma quei pochi produttori (*Analog Devices*, *Texas Instruments*) realizzano componenti eccellenti [43, 44]; anche il condensatore di reazione deve essere però eccellente (basse perdite, basso assorbimento dielettrico), e per questo tali componenti usano condensatori esterni in teflon, costosi ma soprattutto ingombranti. L'uso di ingombranti condensatori esterni non è un problema per un singolo canale, ma lo diventa per sistemi multicanale [45].
Una prima svolta avvenne nel 2006 con l'utilizzo dell'integratore di precisione IVC102, che fa uso di condensatori di reazione integrati: con tale dispositivo vennero realizzati diversi prototipi, fino ad arrivare al picoamperometro PA109. Il PA109 aveva una struttura compatta, una risoluzione di almeno 10 *bit* e una dinamica d'ingresso di quasi 9

ordini di grandezza (100 fA ÷ 100 μA), e si rivelò del tutto idoneo per l'acquisizione e il condizionamento di segnali provenienti da sensori in diamante: le sue caratteristiche erano del tutto confrontabili con quelle ottenute con un *lock-in* [46].
Questa era la situazione del progetto prima che iniziassi l'attività di dottorato presentata in questa tesi.

4.1.1 Sistemi multicanale per misure di corrente

Il PA109 era un dispositivo a singolo canale, e dunque sorgeva il problema di realizzare un sistema di pari caratteristiche, ma multicanale: l'utilizzo di un integratore IVC102 per ogni canale (o la sua versione duale ACF2101) appariva ingombrante e costoso anche per un numero limitato di canali, e dunque occorreva trovare un'alternativa, ferma restando la validità dell'idea di base di impiegare un integratore (versatile perché col tempo d'integrazione si può regolare la caratteristica di trasferimento).
Il percorso seguito per lo sviluppo del progetto è stato quello che si sarebbe adottato a livello aziendale, partendo dalla definizione delle specifiche e dei requisiti del sistema, per passare poi alla ricerca della documentazione, fino ad arrivare alla realizzazione di un prototipo. Per ovvi problemi di spazio, tutta la parte iniziale del lavoro è riassunta in questa introduzione, mentre il corpo centrale del capitolo riguarderà la descrizione del progetto finale condotto.
Con la collaborazione degli ingg. Stefano Salvatori e Nicola Masarone è stata innanzitutto individuata la famiglia di dispositivi che avrebbe risolto gran parte dei problemi legati alla realizzazione di un'elettronica di *front-end* multicanale compatta: si tratta della famiglia DDC, prodotta dalla Texas Instruments, una serie di componenti a ingresso multiplo in corrente. Il dispositivo DDC101, il più semplice della famiglia, è un convertitore A/D sigma-delta a singolo ingresso in corrente, il cui *front-end* è un integratore [47]; la versione duale, il DDC112, ha due canali d'ingresso in corrente e un unico ADC sigma-delta [48]; all'epoca dell'inizio del progetto, la versione più avanzata era il DDC118, con 8 canali d'ingresso [49].
I motivi per la scelta della famiglia di dispositivi DDC sono stati molteplici: innanzitutto, la presenza dell'integratore consentiva di variare il tempo d'integrazione in base al livello di corrente da convertire; il convertitore A/D sigma-delta offriva la necessaria accuratezza che l'applicazione richiedeva; infine, i DDC davano l'occasione di semplificare notevolmente il sistema di acquisizione.
Per l'implementazione di un sistema multicanale si possono adottare due soluzioni:

- Raccogliere i risultati delle conversioni dei diversi convertitori A/D uno ad uno.
- Utilizzare un numero inferiore di convertitori A/D (al limite uno soltanto) e multiplexare le uscite in tensione dei diversi integratori.

Nel primo caso serve un'elettronica digitale di *multiplexing* non molto complessa, ma che potrebbe risultare ingombrante, anche in termini di *layout* del circuito: al *multiplexer* arrivano infatti *n* x *m* fili, dove *n* è il numero di *bit* del convertitore A/D e *m* è il numero di canali (per esempio, con 10 *bit* e 64 canali si avrebbero ben 640 collegamenti!). Nel secondo caso serve un *multiplexer* analogico di elevata accuratezza: una soluzione meno ingombrante, ma adottando la quale si allungano i tempi di

acquisizione, poiché il tempo complessivo è pari a quello di conversione di un convertitore A/D moltiplicato per *m* canali.
I dispositivi DDC consentivano di ottenere i risultati di conversione dei singoli canali in parallelo, ma il trasferimento dei dati poteva avvenire solo su due fili (*bus* seriale *bit* a *bit*): sicuramente il tempo di trasferimento dei dati sarebbe stato superiore rispetto all'utilizzo di un *multiplexer* digitale, ma il *layout* sarebbe stato notevolmente semplificato. Si presentava però un problema relativo all'utilizzo di DDC a più di due ingressi in corrente (es. DDC114 o DDC118): il tipo di *package* di questi dispositivi è QFN, e dunque per il montaggio ci si sarebbe dovuti rivolgere a una ditta specializzata (il successivo DDC232, a 32 canali d'ingresso, è addirittura *ball-grid array* [50]), e in pochi montano tali componenti. Si decise dunque, prima di affrontare l'impegno in termini di tempo e di spesa che l'affidamento del montaggio a una ditta esterna avrebbe comportato, di valutare l'efficacia dell'uso di DDC a basso numero di canali d'ingresso, come il DDC112: se la valutazione fosse andata a buon fine, si sarebbe passati alla realizzazione in laboratorio di un primo dimostrativo multicanale, e successivamente alla realizzazione di un prototipo ancora più compatto presso una ditta specializzata (con l'impiego, ad esempio, del DDC232).

4.1.2 Scheda di valutazione ECSIC2

L'unico componente DDC che poteva essere gestito interamente in laboratorio era il DDC112, con contenitore SOIC a montaggio superficiale e *pin* a 1.27 mm di spaziatura: per esso è stata sviluppata la scheda di valutazione ECSIC2 (Elettronica di Condizionamento per Segnale Impulsivo di Carica a 2 canali), riportata in fig. 4.1:

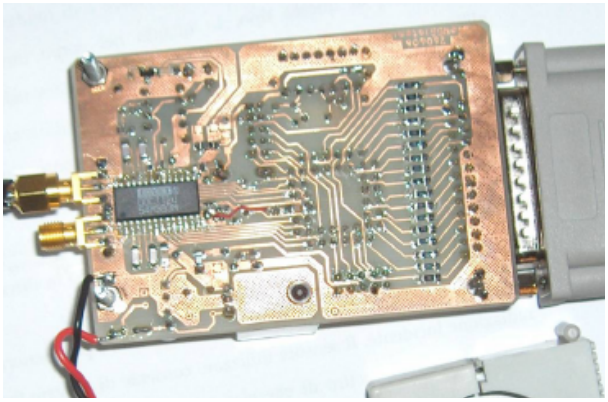


Fig. 4.1 *Scheda di valutazione ECSIC2*

Il DDC112 era pilotato da una logica programmabile, in cui era implementato un circuito di controllo per la generazione dei segnali digitali di gestione del DDC; i dati raccolti venivano poi trasferiti ad un PC tramite porta parallela. Il DDC112 venne caratterizzato con un generatore di corrente di precisione Keithley 6221 [51]; i risultati sono riassunti nelle seguenti due figure:

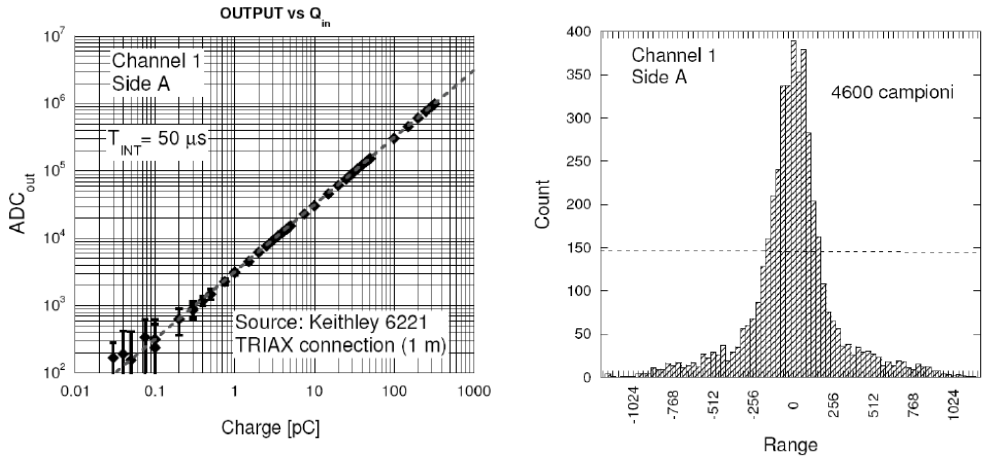


Fig. 4.2 *Caratterizzazione del DDC112 con la scheda ECSIC2*

La prima figura mostra un'ottima linearità per quasi 4 ordini di grandezza (da 100 fC a 500 pC); occorre dire che il tempo d'integrazione era di soli 50 µs, e dunque la linearità su oltre tre ordini di grandezza è sicuramente un buon risultato (correnti da 1 nA a 10 µA). La seconda figura mostra la risoluzione del sistema di conversione (stimata come semplice dispersione dei dati raccolti), che è di almeno 12 *bit*; successive prove, con tempi d'integrazione maggiore (dell'ordine del ms e delle frazioni di s), hanno dato risultati analoghi per correnti anche inferiori al pA.
In sostanza, i risultati indicavano che il DDC112 era idoneo per un sistema multicanale da accoppiare ad un rivelatore multipixel in diamante: terminate le due fasi primarie di ricerca (studio della documentazione e valutazione del componente scelto), si poteva passare allo studio di fattibilità del sistema multicanale.

4.1.3 Studio di fattibilità del sistema multicanale: ECSIC8

Lo scopo prefissato era quello di realizzare un dimostrativo da almeno 64 canali, pensato per un rivelatore a matrice di pixel 8 x 8 (poi ridotto, in fase realizzativa, a un prototipo da 32 canali per un sensore a 6 x 6 pixel, ma solamente per difficoltà nel *wire bonding*). La testa di misura doveva essere progettata affinché si adattasse al collegamento col rivelatore, e inoltre l'elettronica di *front-end* doveva trovarsi (per questioni di rumore) a breve distanza dal campione stesso: la soluzione migliore sarebbe stata quella di impiegare almeno il DDC118 (con soli 8 integrati si poteva realizzare buona parte dell'elettronica di misura), ma per i problemi esposti circa la "lavorabilità" dei componenti a *package* QFN si decise di pensare a un sistema basato su DDC112, al fine di realizzare il prototipo completamente in laboratorio.
Venne avviata la progettazione di un *layout* compatto che integrasse più DDC112: per i 64 canali richiesti erano necessari 32 integrati, e dunque la scheda che li avrebbe alloggiati avrebbe dovuto avere (viste le dimensioni dell'integrato) un'area di almeno 32 x (1.8 x 1) cm^2, a cui aggiungere almeno un 20% per lo sbroglio delle piste. Una scheda

di queste dimensioni appariva troppo ingombrante rispetto alle dimensioni del campione (1 x 1 cm^2), troppo poco maneggevole, e infine di difficile posizionamento di fronte alla sorgente da rivelare. Si optò allora per una struttura modulare: diverse schede, con un numero limitato di integrati, poste immediatamente dietro al campione, in modo da avere una testa di misura sviluppata su tre dimensioni, e non due soltanto. Si decise in pratica di realizzare un modulo di base a 8 canali (4 DDC112), denominato ECSIC8, su una scheda di dimensioni (almeno inizialmente) 3 x 8 cm^2, del tutto idonea dunque a costruire una testa di misura compatta: 8 schede di questo tipo, poste una dietro l'altra, avrebbero permesso di arrivare a 64 canali, con un ingombro in profondità di soli 5 cm circa.
La fig. 4.3 mostra il risultato ottenuto per la singola scheda ECSIC8: il layout attorno ai DDC112 venne ampliato per agevolare il *test* della nuova scheda; in essa vennero anche inseriti i *buffer* d'interfacciamento verso l'elettronica digitale di controllo (nella versione definitiva questi sono stati poi eliminati, per essere inseriti su un *backplane* in cui sono allocate, per formare l'intera testa di misura a 64 canali, più schede ECSIC8).

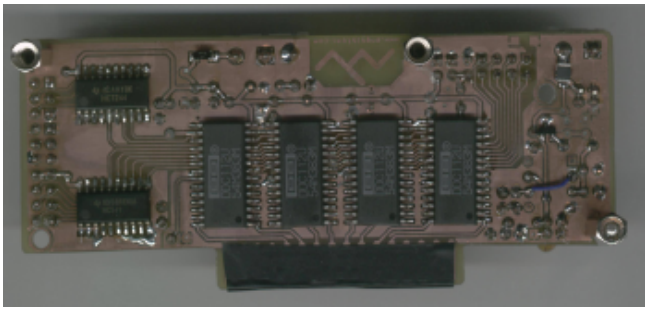


Fig. 4.3 *Scheda ECSIC8 per sistema multicanale a 8 canali*

Il connettore scelto per l'ingresso, rispetto a ECSIC2, è stato trasformato in un pettine da inserire in un connettore femmina standard per schede. La scheda di controllo digitale, realizzata in via preliminare su millefori, era basata fondamentalmente sul vecchio progetto ECSIC2: anche questa scheda di valutazione è stata collaudata col Keithley 6221, confermando i risultati precedentemente ottenuti con ECSIC2.
La prima fase della progettazione del sistema può dirsi conclusa. Nel seguito del capitolo verranno descritti i diversi blocchi che compongono il sistema progettato, denominato ECSIC32, e realizzato nella fase conclusiva del lavoro. Il dimostrativo è stato realizzato a 32 canali, seppure l'elettronica sia stata progettata per lavorare fino a 64 canali: problemi relativi al collegamento fra i singoli pixel e la testa di misura (*wire bonding*) hanno resa necessaria la riduzione del numero di pixel (da 64 a 36, di cui 4 non connessi), e di conseguenza del numero dei canali (da 64 a 32).
Ad ogni modo, dall'anno d'inizio del progetto di ricerca (2006), l'evoluzione in termini di numero di canali è stata esponenziale (vedi fig. 4.4), a testimonianza del raggiungimento degli scopi prefissati.

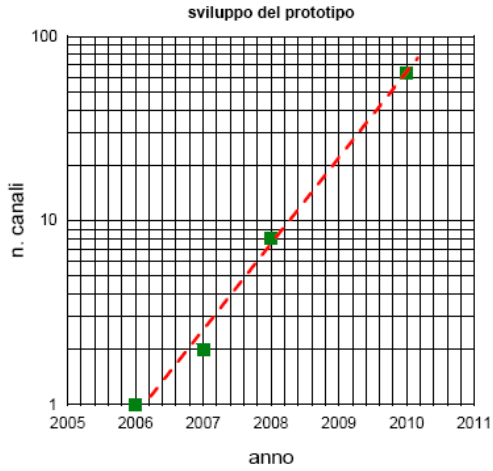


Fig. 4.4 *Evoluzione del progetto ECSIC*

4.2 Schema a blocchi del sistema ECSIC32

ECSIC32 (Elettronica di Condizionamento per Segnali Impulsivi di Carica a 32 canali), progettato nell'ambito di quest'attività di ricerca, è un sistema compatto per il condizionamento e l'elaborazione dei segnali provenienti dal rivelatore a pixel, ed è in grado di fornire in uscita dati che permettono di ricostruire in tempo reale il profilo delle radiazione incidente, nonché informazioni utili riguardanti la posizione del centroide del fascio, i valori d'intensità massima e minima, e lo spostamento del fascio rispetto ad una posizione di riferimento.
In fig. 4.5 è riportato lo schema a blocchi del sistema ECSIC32. Il blocco di condizionamento dei segnali provenienti dal rivelatore (DDC) è costituito da un banco di 16 DDC112, convertitori A/D a doppio ingresso in corrente e 20 *bit* di risoluzione. Si possono dunque utilizzare rivelatori fino a 32 pixel; ad ogni modo, sulla *main board* di ECSIC32, sono previsti una serie di connettori per eventuali future espansioni del sistema a 64 o più canali (ECSIC64, etc.)
I segnali necessari al corretto funzionamento e alla gestione del banco di DDC (CONV, TEST, DXMIT, DCLK) vengono forniti da una coppia di dispositivi logici programmabili (CPLD). La scelta di utilizzare una coppia di CPLD al posto del microcontrollore è dovuta al fatto che i segnali di controllo del banco di DDC devono essere sincronizzati entro 10 ns rispetto ai fronti del segnale di *clock* [48] (generato da un quarzo di precisione a 10 MHz): il μC LPC2103, utilizzato in ECSIC32 per altri scopi, non consente di soddisfare queste stringenti specifiche temporali [52]. Inoltre, la coppia di CPLD rende l'intero sistema più veloce, in quanto genera segnali in parallelo, mentre il μC è notoriamente un sistema sequenziale.

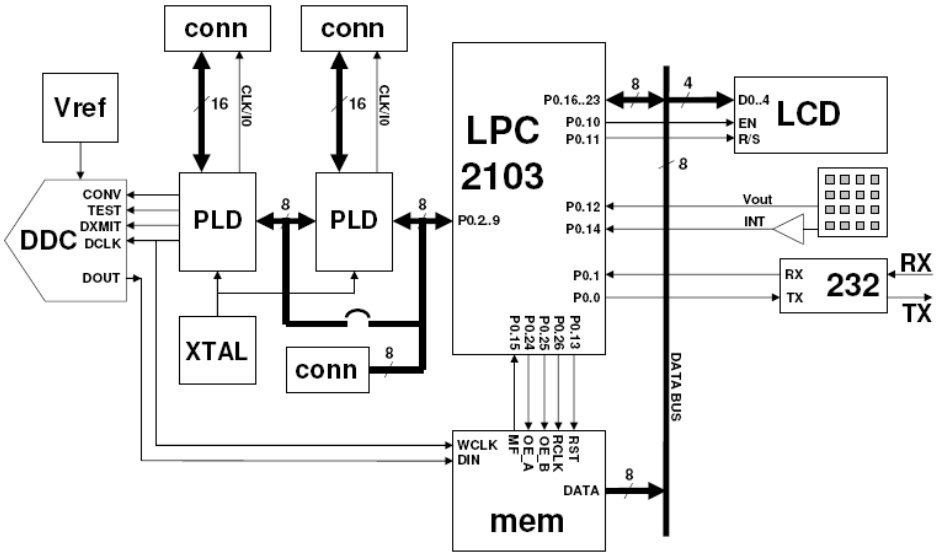


Fig. 4.5 *Schema a blocchi del sistema ECSIC32*

Il cuore del sistema ECSIC32 è il microcontrollore LPC2103, utilizzato per l'affidabilità e la versatilità del suo microprocessore ARM integrato. L'LPC2103 rappresenta lo stato dell'arte dei microcontrollori, essendo dotato di un μP integrato a 14.4 x 4 MHz [53]. Il μC svolge una serie di funzioni fondamentali per l'intera elettronica di condizionamento e di lettura :

- si occupa di impostare e gestire la comunicazione con la coppia di CPLD, fornendo le informazioni utili per la generazione dei segnali di controllo del banco di convertitori: tempo d'integrazione, *range* di carica, numero di impulsi di *test*, quantità di dati (da 32, corrispondenti ad una sola acquisizione, a 2^{10}·32)
- gestisce l'interfacciamento del sistema con l'utente esterno: si occupa del pilotaggio di un *display* a cristalli liquidi (LCD) sul quale è possibile controllare le impostazioni dell'utente ed andare a leggere i risultati dell'elaborazione dati, dell'interfacciamento di una tastiera analogica a 16 tasti [54] con la quale l'utente può impostare i parametri di misura, e infine dell'interfacciamento con un PC attraverso un'interfaccia seriale RS-232 (scelta perché ben collaudata rispetto allo standard USB, e anche perché il μC LPC2103 integra un dispositivo UART).
- si occupa della gestione e del controllo della memoria, nella quale vengono temporaneamente immagazzinati i dati delle diverse acquisizioni prima di essere inviati sul *data bus*.

Il banco di memoria è costituito da 5 SRAM CMOS ad alta velocità e basso consumo di potenza [55]. Ogni *chip* di memoria ha una capacità di 32 kB, è dunque è possibile memorizzare nell'intero banco 2048 *frames* (quadri) in caso di sensore a 32 canali (20 *bit* per canale). La memoria è dotata di una sua logica CPLD: senza controllo da parte del μC, riceve autonomamente *bit* a *bit* i dati dal banco dei DDC, ed è dunque vista dal

banco dei DDC come un registro a scorrimento, la cui dimensione è impostata dal μC (da 32 a 2^{10}·32).
Quando il registro a scorrimento è pieno, viene attivato il segnale MF (*Memory Full*), che segnala con un *interrupt* al μC la possibilità di prelevare i dati. I dati vengono letti serialmente ma per *byte*, ovvero gli 8 *bit* che compongono il *byte* viaggiano su un unico *data bus*: questo consente di aumentare la velocità di lettura rispetto al caso *bit* a *bit*, e inoltre permette di risparmiare i *pin* del μC necessari all'indirizzamento della memoria. E' inoltre prevista la possibilità di poter leggere la memoria più volte (a meno che non venga resettata), con l'obiettivo di consentire al μC di ottenere ed elaborare informazioni addizionali sulla radiazione incidente (centroide, valore atteso, etc.) senza dover utilizzare la sua RAM interna, che ha una dimensione di soli 8 kB [53].
Qui di seguito viene riportato un elenco in cui sono evidenziate le specifiche, le potenzialità e la versatilità del sistema ECSIC32:

- Tempo d'integrazione selezionabile da 50 μs a 1 s.
- Fondo-scala di carica selezionabile da 50 pC a 350 pC.
- Sensibilità (LSB) inferiore a 15 pC (a 150 pC di fondo-scala).
- ADC con risoluzione 20 *bit* (*no missing code*), ridotta a 14 *bit* nel caso peggiore (ingressi *floating*).
- Numero di acquisizioni selezionabile da 1 a 1024.
- *Throughput* massimo: 2 kHz (2·64 kSPS).
- Frequenza di trasmissione dati al PC (*baud rate*): 460800 *bit*/s.
- Possibilità di controllo da remoto tramite PC (interfaccia RS-232).
- Capacità di calcolo di parametri della radiazione incidente: centroide, valore massimo, valore minimo, valore atteso, scarto quadratico medio, spostamento.

Nei successivi paragrafi verranno analizzate nel dettaglio le diverse parti del sistema ECSIC32, con particolare riferimento al progetto delle logiche programmabili e delle funzionalità del microcontrollore.

4.3 Convertitore analogico/digitale DDC112

Il DDC112 è un convertitore analogico-digitale (A/D) della Burr Brown, a doppio ingresso in corrente e 20 *bit* di risoluzione. Dispositivi a basso livello di corrente in uscita, come i fotorivelatori, possono essere direttamente collegati ai suoi ingressi. L'integrazione di carica è continua, in quanto ogni ingresso utilizza due integratori: mentre il primo integratore sta integrando, i dati del secondo vengono digitalizzati, e viceversa.
Per ognuno dei due ingressi, il DDC112 effettua: conversione corrente-tensione, integrazione continua su un *range* di carica programmabile fino a 1000 pC, conversione A/D e filtraggio digitale.
Al fine di utilizzare un'alimentazione singola, il convertitore A/D interno usa un ingresso differenziale, con l'ingresso positivo agganciato alla tensione di riferimento V_{REF}. Quando la capacità d'integrazione viene resettata all'inizio di un nuovo ciclo

d'integrazione, essa viene caricata a V_{REF}, dopo di che comincia a scaricarsi in maniera proporzionale alla corrente in ingresso. Alla fine del ciclo d'integrazione, la tensione residua in uscita all'integratore viene confrontata con V_{REF}. Un registro a scorrimento con uscita seriale ad alta velocità memorizza il risultato dell'ultima conversione, e può essere configurato per permettere l'utilizzo di più DDC112 in cascata (*daisy-chained*).
Il funzionamento di base del DDC112 è illustrato in fig. 4.6:

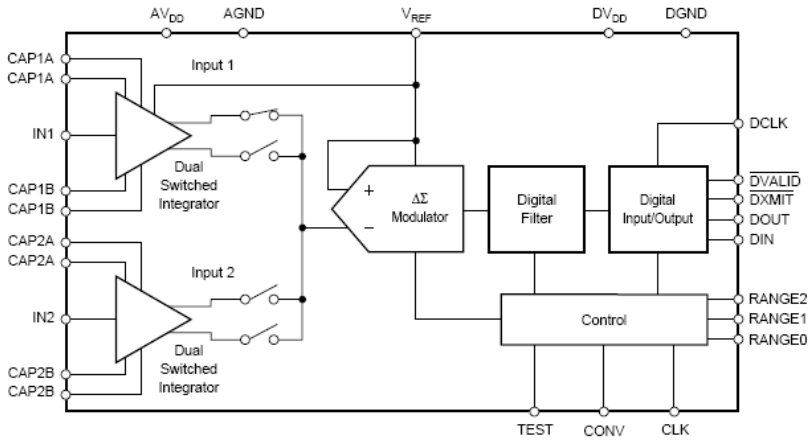


Fig. 4.6 *Schema a blocchi del DDC112*

Il dispositivo contiene due canali d'ingresso identici, in ognuno dei quali avviene un'integrazione corrente-tensione, seguita da una conversione A/D multiplexata. Ogni ingresso contiene due integratori, permettendo di effettuare un'integrazione corrente-tensione continua nel tempo. Le uscite dei quattro integratori vengono inviate singolarmente ad un modulatore Δ-Σ (convertitore A/D) attraverso un *multiplexer* a quattro ingressi. Nella modalità di operazione "continua", le uscite dei due integratori di un lato (es. B) di entrambi gli ingressi vengono digitalizzate, mentre gli altri due integratori (es. A) sono in modalità integrazione, come evidenziato dal digramma temporale in fig. 4.7:

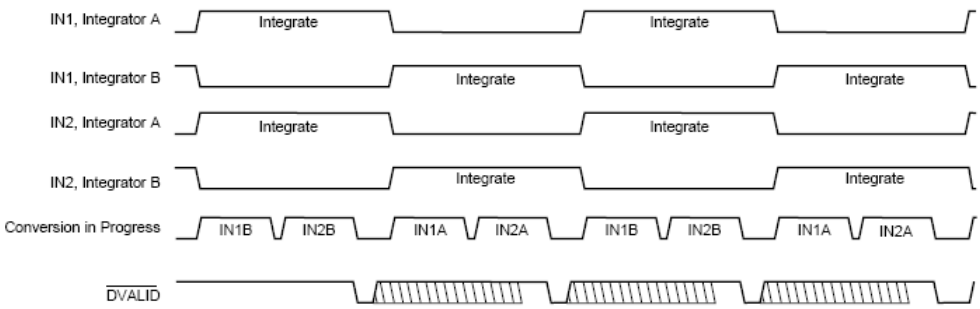


Fig. 4.7 *Cicli di integrazione-conversione a doppio canale*

Questo processo combinato di integrazione-conversione A/D è controllato dal *clock* di sistema, CLK: con un CLK di 10 MHz, l'integratore e il modulatore Δ-Σ compiono una conversione a 20 *bit* in circa 220 μs. I risultati del lato A e del lato B di ogni segnale d'ingresso vengono immagazzinati in un registro a scorrimento a uscita seriale: l'uscita DVALID si attiva (è attiva bassa) quando il registro a scorrimento contiene dati validi. L'interfaccia digitale del DDC112 fornisce i risultati digitali attraverso un'interfaccia seriale sincrona, costituita da un *clock* per il trasferimento dei dati (DCLK), un *pin* per l'abilitazione della trasmissione dei dati (DXMIT), un *pin* per segnalare la presenza di dati validi (DVALID), un'uscita seriale per i dati (DOUT), e un ingresso seriale per i dati (DIN). I processi d'integrazione e conversione sono indipendenti dal processo di acquisizione dei dati: di conseguenza, CLK e DCLK non devono necessariamente avere la stessa frequenza. DIN è usato solo quando sono previsti più DDC112 in cascata, in caso contrario deve essere collegato alla massa digitale (DGND). A seconda del tempo d'integrazione T_{INT}, CLK e DCLK, è possibile collegare in cascata (*daisy-chained*) più di 100 convertitori: questo semplifica enormemente la progettazione del *layout* delle linee d'interconnessione per le uscite digitali nel caso in cui si abbia bisogno di un gran numero di convertitori.

4.3.1 Ciclo base d'integrazione

La struttura dello stadio d'ingresso del DDC112 è quella di un classico integratore analogico (fig. 4.8): amplificatore operazionale, rete capacitiva di controreazione di valore programmabile, e una serie di interruttori per l'implementazione del ciclo d'integrazione .

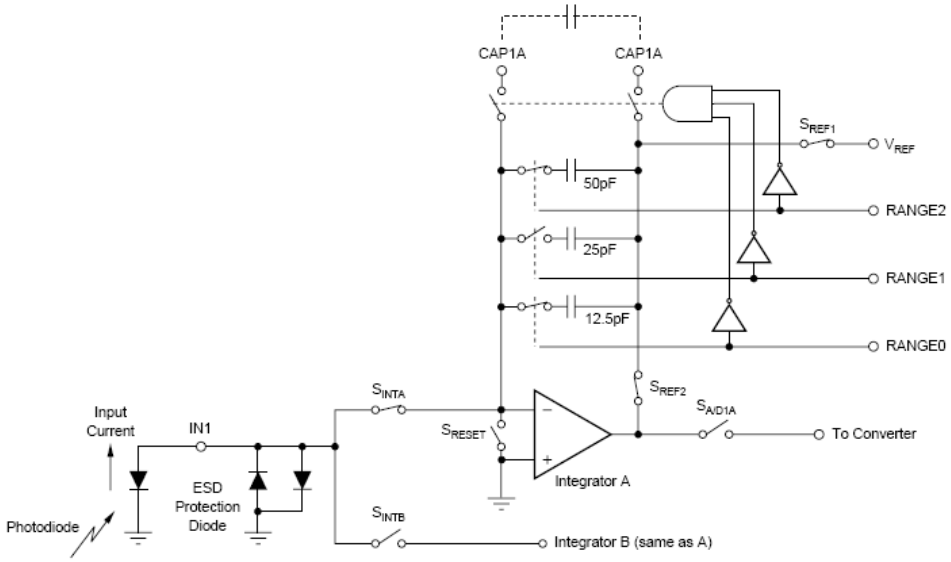


Fig. 4.8 *Stadio d'ingresso del DDC112*

Le relazioni temporali fra i diversi interruttori della fig. 4.8 sono illustrati in fig. 4.9, utilizzata solo a scopo descrittivo per il funzionamento dello stadio d'ingresso dell'integratore. I diagrammi a blocchi relativi agli stati di *reset*, integrazione, conversione e attesa dello stadio d'ingresso sono riportati in fig. 4.10.
La rete interna di commutazione degli interruttori è controllata esternamente con il segnale di conversione (CONV), i *pin* di selezione del *range* di carica da misurare (RANGE 0,1,2) e il *clock* di sistema (CLK). Per migliorare le prestazioni in termini di rumore, CONV dev'essere sincronizzato con il fronte di salita di CLK (con un anticipo o un ritardo massimi di 10 ns).
Gli ingressi non invertenti degli integratori sono internamente collegati a massa, di conseguenza la massa analogica del DDC112 dev'essere il più possibile pulita. Gli interruttori di *range* e le capacità di *feedback* interne ed eventualmente esterne (C_F) sono mostrate in parallelo fra l'ingresso invertente e l'uscita dell'operazionale.

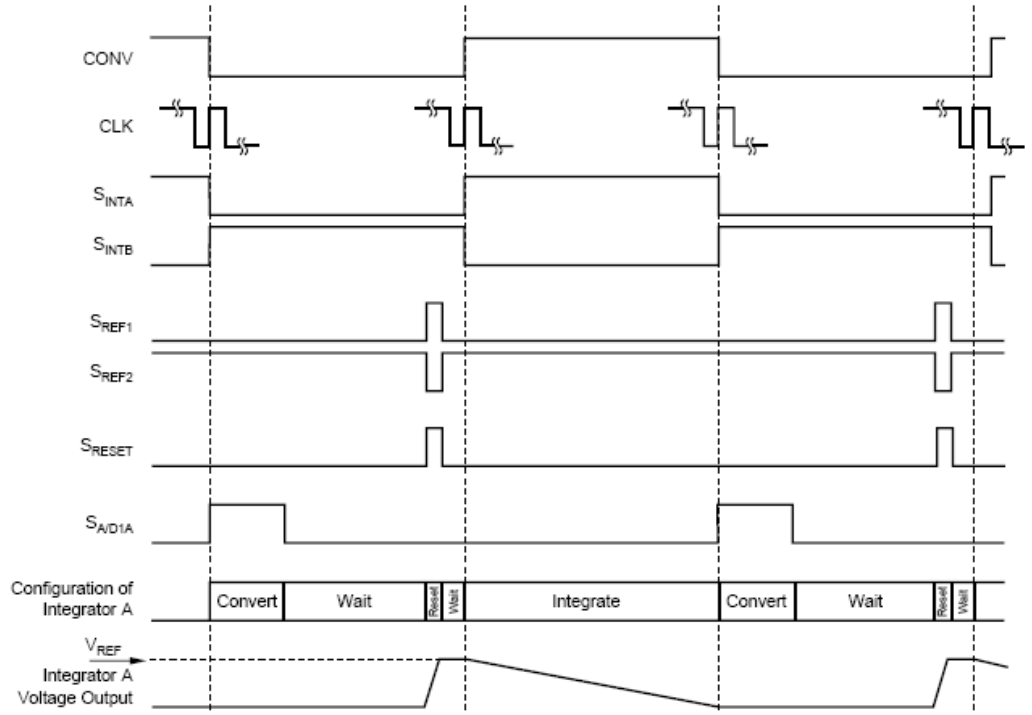


Fig. 4.9 *Diagramma temporale relativo al funzionamento dello stadio d'ingresso del DDC112*

All'inizio di un ciclo di conversione, gli interruttori $S_{A/D}$, S_{INTA}, S_{INTB}, S_{REF1}, S_{REF2} e S_{RESET} sono impostati come in fig. 4.9. Al completamento della conversione A/D, dopo un periodo di attesa, la carica sulla capacità d'integrazione C_F viene resettata (nel tempo di *reset*) chiudendo S_{REF1} e S_{RESET} e aprendo S_{REF2} (fig. 4.10a): in questa maniera, la capacità selezionata si carica alla tensione di riferimento V_{REF}. Una volta completato il caricamento della capacità, S_{REF1} e S_{RESET} si aprono e S_{REF2} si chiude, cosicché V_{REF} non

è più collegata al circuito di amplificazione (stato di attesa prima dell'integrazione, fig. 4.10b). Al successivo fronte di salita di CONV, S_{INTA} si chiude e comincia l'integrazione del lato A: questo pone lo stadio integratore nella modalità integrazione (fig. 4.10c). La carica contenuta nel segnale d'ingresso viene quindi raccolta dalla capacità d'integrazione, causando una caduta di tensione all'uscita dell'amplificatore. Il fronte di discesa di CONV interrompe l'integrazione del lato A, commutando il segnale d'ingresso dal lato A al lato B (S_{INTA} si apre e S_{INTB} si chiude) e dando il via all'integrazione del lato B. Nel periodo di tempo in cui il lato A stava integrando (prima del fronte di discesa di CONV), il segnale presente sul lato B veniva digitalizzato dal convertitore A/D e poi resettato: al fronte di discesa di CONV, il lato B comincia ad integrare il segnale d'ingresso, ed il segnale in uscita dal lato A si presenta all'ingresso del modulatore Δ-Σ (fig. 4.10d).

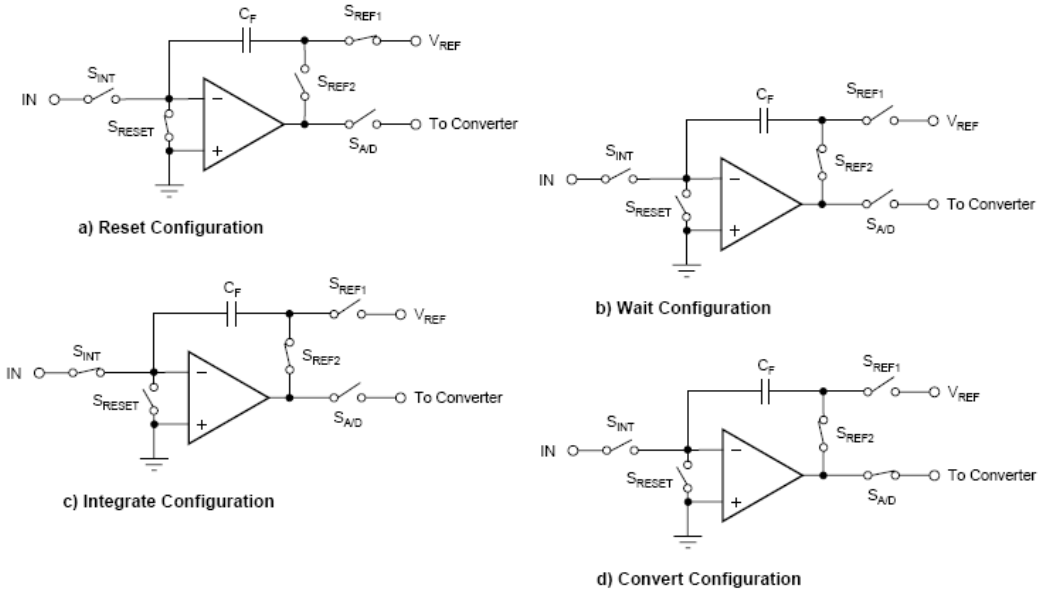


Fig 4.10 *Configurazioni dello stadio d'ingresso del DDC112*

Il valore C_F della capacità di *feedback* dell'integratore, il tempo T_{INT} d'integrazione e la tensione di riferimento V_{REF} determinano il fondo-scala positivo (+FS) del DDC112. Il valore approssimato di corrente I_{FS} corrispondente al fondo-scala positivo può essere ricavato dall'espressione:

$$I_{FS} = 0.96 \frac{C_F V_{REF}}{T_{INT}} \qquad [IV.1]$$

Il fattore 0.96 permette agli integratori dello stadio d'ingresso di raggiungere il fondo-scala senza dover tornare completamente a massa. Il fondo-scala negativo (−FS) è

approssimativamente lo 0.4% di quello positivo: ad esempio, per un fondo-scala positivo di 250 pC abbiamo un fondo-scala negativo di −1pC.
Nel DDC112, ci sono sette diverse capacità (disponibili internamente al *chip*) per ogni lato di ognuno dei due canali. I *pin* di controllo del *range* (RANGE 0,1,2) cambiano il valore della capacità di *feedback* per ognuno dei quattro integratori: di conseguenza, a meno che non si utilizzino capacità esterne, entrambi gli ingressi ed entrambi i lati di ogni ingresso avranno sempre lo stesso valore di fondo-scala.

4.3.2 Tensione di riferimento

La tensione esterna di riferimento viene usata per resettare le capacità di *feedback* prima dell'inizio di un nuovo ciclo d'integrazione. E'inoltre usata dal modulatore Δ-Σ durante le operazioni di misura della tensione accumulata sugli integratori dopo la fine del ciclo d'integrazione. Durante questa operazione di campionamento, il riferimento esterno deve fornire al modulatore Δ-Σ la carica necessaria: per un tempo d'integrazione di 500 μs, tale carica si traduce in una corrente media di riferimento di approssimativamente 150 μA. La quantità di carica necessaria al modulatore Δ-Σ è indipendente dal tempo d'integrazione: pertanto, un aumento del tempo d'integrazione determina una diminuzione della corrente media di riferimento. Ad esempio, un tempo d'integrazione di 1000 μs abbassa I_{REF} a un valore medio di 75 μA.
E' molto importante che la tensione V_{REF} sia stabile durante le diverse modalità di funzionamento (*reset*, integrazione, conversione e attesa). Il modulatore Δ-Σ misura infatti la tensione sull'integratore confrontandola con V_{REF}: visto che le capacità d'integrazione sono inizialmente resettate a V_{REF}, un'eventuale caduta di V_{REF} (dall'istante di *reset* delle capacità all'istante in cui il convertitore misura l'uscita dell'integratore) introdurrebbe un *offset*. E' importante anche che V_{REF} sia stabile per lunghi periodi di tempo, in quanto dei cambiamenti in V_{REF} corrispondono direttamente a modifiche nel fondo-scala. Infine, V_{REF} dovrebbe introdurre il minor rumore addizionale possibile.
Per tutte le ragioni menzionate finora, è importante che l'accoppiamento fra la sorgente di riferimento esterna e il DDC112 sia effettuato attraverso un *buffer*, come mostrato in fig. 4.11:

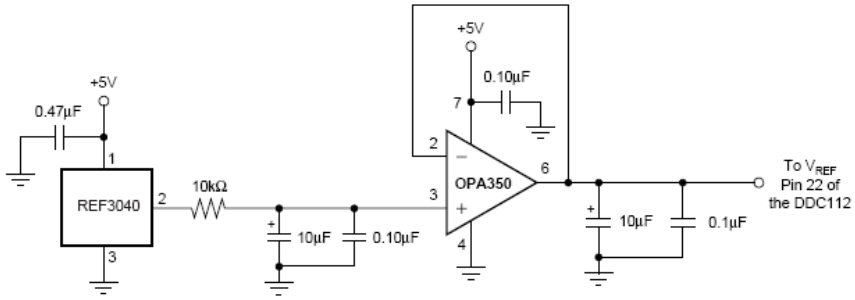


Fig. 4.11 *Generatore della tensione di riferimento per il DDC112*

In questo circuito, il riferimento di tensione a 4.096 V è generato dal componente REF3040. Un filtro passa-basso per l'attenuazione del rumore ad alta frequenza è interposto fra il riferimento e un amplificatore operazionale configurato come *buffer*: tale amplificatore deve avere un prodotto guadagno-banda superiore a 4 MHz, basso rumore, e un *range* di modo comune (per l'ingresso e per l'uscita) che sia compatibile con V_{REF}. A valle del *buffer* ci sono delle capacità di filtro da posizionare vicino al *pin* del DDC112 destinato a ricevere V_{REF}: queste grandi capacità potrebbero creare problemi d'instabilità, ma questo non si verifica per la maggior parte degli amplificatori operazionali. Per migliorare la stabilità del circuito, si potrebbe pensare di mettere in uscita una resistenza serie, ma questo in genere si evita perché potrebbe provocare una caduta di V_{REF} (causando larghi *offset*).

4.3.3 Risposta in frequenza

La risposta in frequenza del DDC112 è determinata dagli integratori in ingresso, ed è quella di un classico integratore tempo-continuo, come mostrato in fig. 4.12.

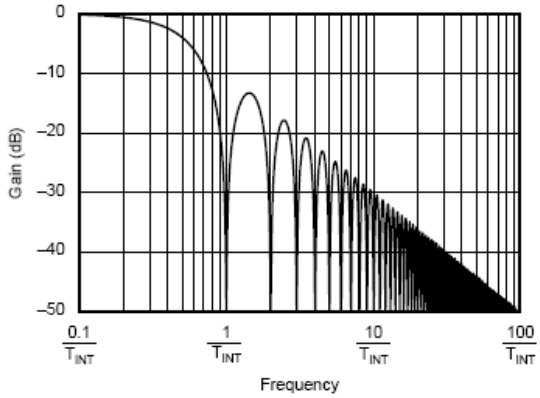


Fig. 4.12 *Risposta in frequenza del DDC112*

Regolando il tempo d'integrazione, è possibile variare le posizioni di f_{3dB} e dei *notch* nella risposta. La risposta in frequenza del modulatore Δ-Σ non ha alcun tipo di conseguenza, perché il modulatore campiona un segnale "tenuto" dagli integratori: quindi, l'ingresso del modulatore Δ-Σ è un segnale in continua. Visto che l'uscita degli integratori viene campionata, possono accadere fenomeni di *aliasing* se la frequenza del segnale d'ingresso eccede la metà della frequenza di campionamento.

4.3.4 Funzionamento in modalità "*test*"

Nella modalità "*test*", i *pin* d'ingresso degli integratori vengono connessi a massa, e pacchetti di carica di circa 13 pC vengono trasferiti alle capacità d'integrazione di entrambi gli ingressi. Questo trasferimento di carica fissa avviene durante il ciclo

d'integrazione, e può essere di un solo pacchetto da 13 pC oppure di pacchetti multipli: in quest'ultimo caso, ogni pacchetto da 13 pC si aggiunge alla carica già trasferita.
I diagrammi temporali della modalità "*test*" sono mostrati in fig. 4.13.

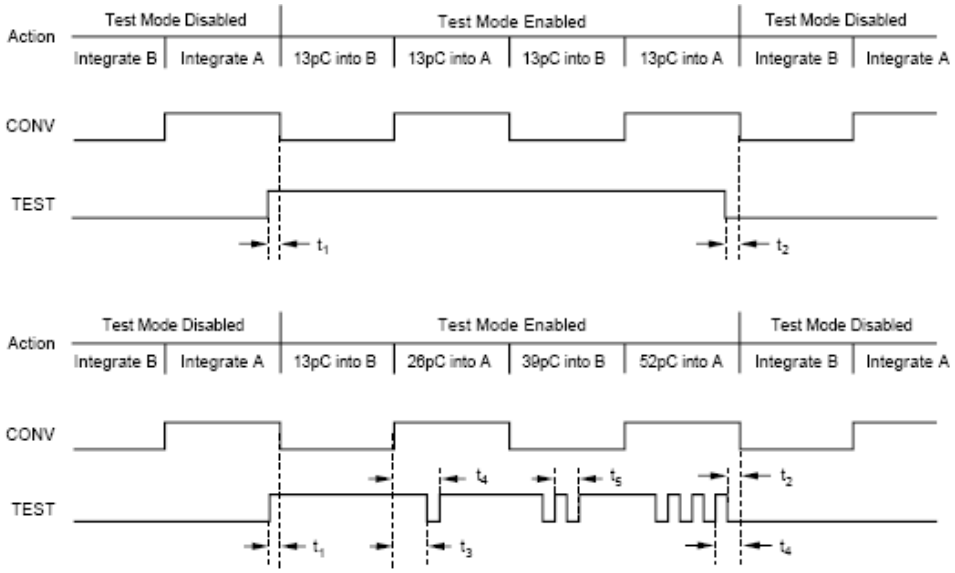


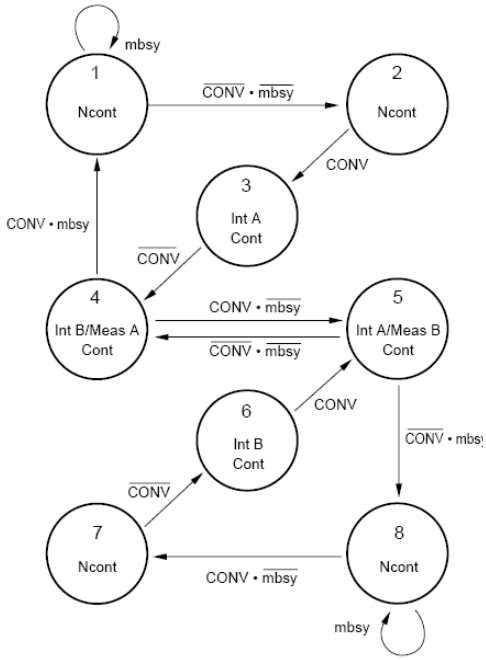
Fig. 4.13 *Diagramma temporale relativo al funzionamento in modalità test*

La parte superiore della fig. 4.13 si riferisce al trasferimento di un singolo pacchetto di carica da 13 pC, mentre la parte inferiore si riferisce al trasferimento di pacchetti di carica multipli: ogni volta che, all'interno di un singolo ciclo d'integrazione, si ha un fronte di salita di TEST, un pacchetto di carica addizionale di 13 pC si aggiunge alla carica già accumulata sulle capacità d'integrazione.
I segnali TEST e CONV lavorano insieme per implementare questa funzione. Si entra nella modalità "*test*" quando TEST diventa alto prima di un fronte di CONV: a questo punto, un fronte di CONV fa partire il collegamento a massa degli ingressi analogici e il trasferimento di un pacchetto di carica da 13 pC alle capacità d'integrazione. Se TEST è mantenuto alto durante almeno due integrazioni (cioè un fronte di salita e uno di discesa di CONV), tutti e quattro gli integratori saranno caricati con un pacchetto da 13 pC. Alla fine di ogni integrazione, la tensione in uscita agli integratori viene come sempre digitalizzata. Una volta che si entra in modalità "*test*", TEST può essere attivo per quanti cicli si desidera: si esce dalla modalità "*test*" quando TEST è basso e si ha un fronte di CONV.

4.3.5 Modalità di operazione "continua" e "non-continua"

Il *rate* di conversione del DDC112 è determinato da una combinazione fra il tempo d'integrazione (selezionato dall'utente) e la velocità di conversione del convertitore A/D

(che è principalmente una funzione della frequenza del *clock* di sistema, CLK). Un ciclo completo di conversione A/D comprende la conversione di due segnali (uno per ogni ingresso del DDC112) e il *reset* di tutti gli integratori coinvolti nelle due conversioni.
Nella maggior parte delle situazioni, il tempo di conversione A/D è minore del tempo d'integrazione: in questo caso, il DDC112 opera in modalità "continua", ovvero l'uscita del fotorivelatore è continuamente integrata da uno dei due lati di ogni ingresso.
Nel caso in cui il tempo di conversione A/D è maggiore del tempo d'integrazione, il DDC112 commuta nella modalità di operazione "non-continua": in questo caso, il convertitore A/D non riesce a tenere il passo degli integratori, e di conseguenza il processo d'integrazione è periodicamente mantenuto in *stand-by* fino al completamento della conversione A/D.
Il diagramma degli stati del DDC112 è riportato in fig. 4.14: in tutto sono previsti otto stati. Si usano quattro segnali per controllare la progressione degli stati: CONV, *mbsy* e i loro complementari. La macchina a stati controlla la progressione sui livelli, e non sui fronti di CONV. Il segnale *mbsy* è generato internamente e non è controllabile dall'utente: si attiva ogni volta che avviene un ciclo di misura/*reset*/auto-zero (m/r/az). Si definiscono stati in "in modalità continua" (*cont mode states*) quelli in cui si compiono integrazioni (3,4,5 e 6), mentre gli stati in cui non si compie alcuna integrazione vengono definiti stati "in modalità non-continua" (*ncont mode states*: 1,2,7 e 8). In modalità "continua", *mbsy* è sempre basso prima che CONV commuti, il che vuol dire che il lato che non sta integrando è sempre pronto a cominciare l'integrazione quando l'altro lato ha finito di integrare: di conseguenza, conoscere lo stato corrente di CONV è tutto quello che serve per sapere in quale stato si è. Negli stati 3 e 6 si compiono solo integrazioni (nessun ciclo m/r/az), mentre negli stati 4 e 5 si compiono sia integrazioni che digitalizzazioni (cicli m/r/az).



STATE	MODE	DESCRIPTION
1	Ncont	Complete m/r/az of side A, then side B (if previous state is state 4). Initial power-up state when CONV is initially held HIGH.
2	Ncont	Prepare side A for integration.
3	Cont	Integrate on side A.
4	Cont	Integrate on side B; m/r/az on side A.
5	Cont	Integrate on side A; m/r/az on side B.
6	Cont	Integrate on side B.
7	Ncont	Prepare side B for integration.
8	Ncont	Complete m/r/az of side B, then side A (if previous state is state 5). Initial power-up state when CONV is initially held LOW.

Fig. 4.14 *Diagramma degli stati del DDC112*

Il segnale *mbsy* diventa importante in modalità "non-continua" (stati 1,2,7 e 8): ogni volta che CONV commuta mentre *mbsy* è alto, il DDC112 entrerà (o rimarrà) nello stato 1, oppure 8. Dopo che *mbsy* si abbassa, si entra nello stato 2, oppure 7: in questo stato, il lato corrispondente si prepara per l'integrazione.
Un'interessante osservazione che si può fare osservando il diagramma degli stati è che le integrazioni si alternano sempre fra i lati A e i lati B dei due ingressi: questa relazione è valida per ogni andamento di CONV, ed è indipendente dalla modalità di operazione, in quanto gli stati 2 e 7 ne assicurano la validità anche in modalità "non-continua".
Non appena si accende il DDC112, lo stato iniziale può essere 1 o 8, a seconda del livello iniziale di CONV: se CONV è alto al *power-up* lo stato iniziale sarà 1, altrimenti sarà 8. In generale, nel diagramma c'è una corrispondenza simmetrica fra gli stati 1 e 8, 2 e 7, 3 e 6, 4 e 5: un'inversione di CONV si traduce in una progressione degli stati simmetrica a quella corrispondente al CONV non invertito.

4.3.5.1 Esempi di diagrammi di temporizzazione

La fig. 4.15 mostra alcuni cicli d'integrazione (comprensivi di *power-up* iniziale) per un esempio di funzionamento in modalità "continua".

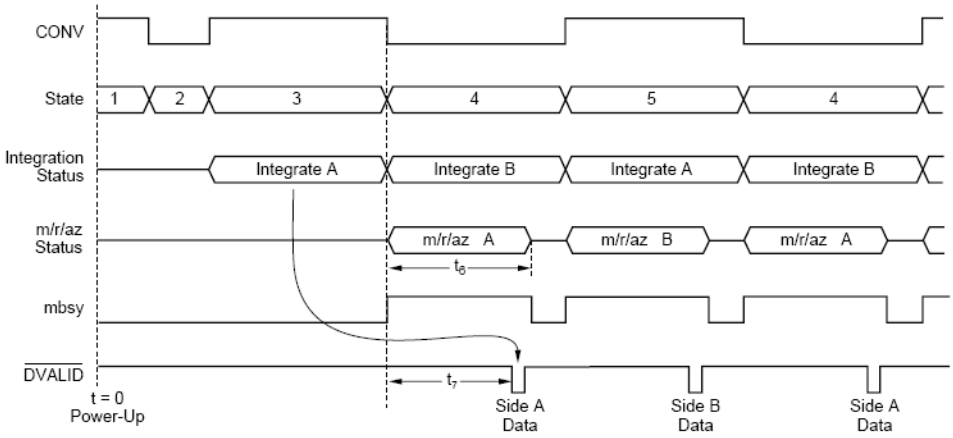


Fig. 4.15 *Diagramma di temporizzazione in modalità "continua"*

Nel diagramma compaiono nell'ordine: il segnale CONV fornito dall'utente, la progressione degli stati del diagramma, lo stato delle integrazioni, lo stato dei cicli di misura, il segnale interno *mbsy*, e il segnale DVALID (attivo basso quando i dati sono pronti per essere trasmessi). DVALID rimane attivo basso finché il segnale DXMIT non viene posto basso dall'utente.
Come si vede, all'accensione il DDC112 si mette nello stato 1, in quanto CONV parte alto: dopo lo stato 2 (in cui si prepara il lato A all'integrazione), si passa nello stato 3, in cui si integra sul lato A ma non avvengono cicli m/r/az. Dopo di ché, il DDC112 commuta continuamente fra lo stato 4 (in cui il lato B integra e la precedente integrazione su A viene digitalizzata) e lo stato 5 (in cui il lato A integra e la precedente integrazione su B viene digitalizzata). L'ingresso viene comunque integrato in maniera

continua, dal lato A oppure dal lato B. Il tempo t_6 necessario per la completa digitalizzazione (ciclo m/r/az) è lo stesso intervallo di tempo che fa da confine fra le due modalità di funzionamento “continua” e “non-continua”: per CLK = 10 MHz, t_6 = 479.4 μs. DVALID si attiva basso dopo un tempo t_7 successivo alla commutazione di CONV, indicando che i dati sono pronti per essere raccolti.
La fig. 4.16 illustra un generico diagramma di temporizzazione in modalità di funzionamento “non-continua”:

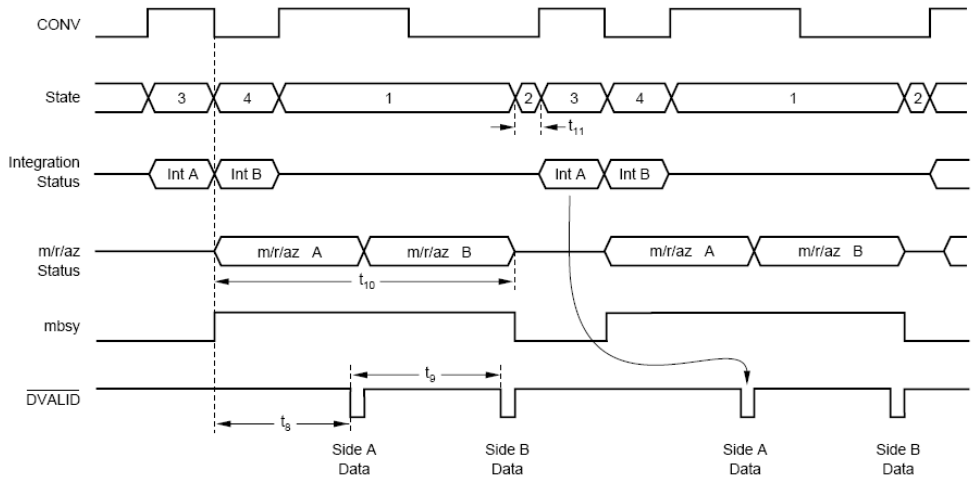


Fig. 4.16 *Diagramma di temporizzazione in modalità “non-continua”*

L’integrazione avviene a coppie (cioè lato A/lato B, oppure lato B/lato A, a seconda di come parte CONV), dopo di che c’è un certo intervallo di tempo in cui non avvengono integrazioni: durante questo intervallo di tempo, le integrazioni precedenti completano la loro digitalizzazione (ciclo di misura/*reset*/auto-zero per entrambi i lati). Prima che il DDC112 possa avanzare allo stato 3 (o 6) e ricominciare ad integrare, entrambi i lati A e B devono aver completato il processo di digitalizzazione delle integrazioni, il che richiede un tempo t_{10}. Quando i cicli m/r/az sono terminati, è necessario un tempo t_{11} per preparare il nuovo ciclo di integrazioni: è richiesto questo ritardo aggiuntivo perché i cicli m/r/az in modalità “non-continua” sono leggermente diversi da quelli in modalità “continua”. Dopo che la prima integrazione (es. su A) è terminata, DVALID si attiva basso nel tempo t_8 (un tempo uguale al tempo t_7 della modalità “continua”); il secondo dato (es. su B) è invece pronto dopo un tempo t_9 dalla prima attivazione di DVALID.
E’ importante notare che quando il DDC112 opera in modalità “non-continua”, passa sia attraverso stati *cont mode* che *ncont mode*: le integrazioni avvengono negli stati *cont mode*, mentre negli stati *ncont mode* vengono completate le digitalizzazioni delle precedenti integrazioni.

4.3.5.2 Cambiamento di modalità di operazione

Il passaggio dalla modalità di operazione "continua" a quella "non-continua" avviene ogniqualvolta il tempo d'integrazione è minore di t_6 (il tempo necessario alla completa digitalizzazione di un'integrazione). La fig. 4.17 mostra un esempio di questa transizione:

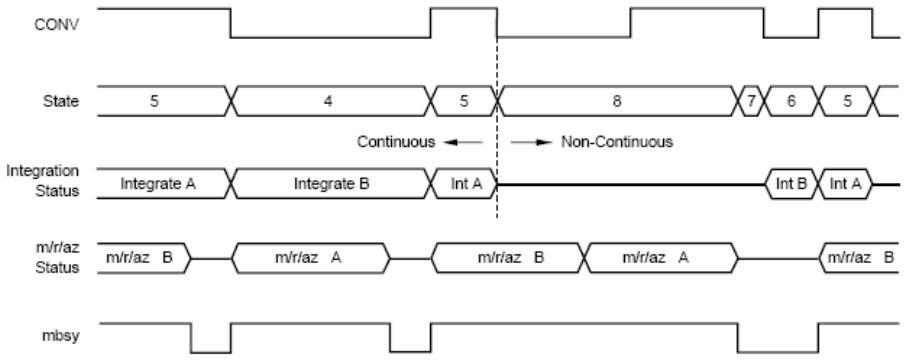


Fig. 4.17 *Passaggio dalla modalità "continua" alla modalità "non-continua"*

Come si vede in figura, si entra nella modalità "non-continua" (ad esempio) quando l'integrazione sul lato A viene completata prima della fine del ciclo m/r/az relativo alla precedente integrazione su B. Il DDC112 completa la digitalizzazione dei lati B ed A (durante gli stati 8 e 7) con gli ingressi analogici messi a massa, dopo di che si ha una nuova integrazione sul lato B nello stato 6.

Il passaggio dalla modalità di operazione "non-continua" a quella "continua" avviene ogniqualvolta il tempo d'integrazione viene aumentato in modo tale da essere sempre maggiore di t_6: con un tempo d'integrazione più lungo, il ciclo m/r/az relativo ad un'integrazione può essere completato prima dell'inizio dell'integrazione successiva, ed è dunque possibile un'integrazione continua del segnale in ingresso. Un esempio di questa transizione è riportato in fig. 4.18.

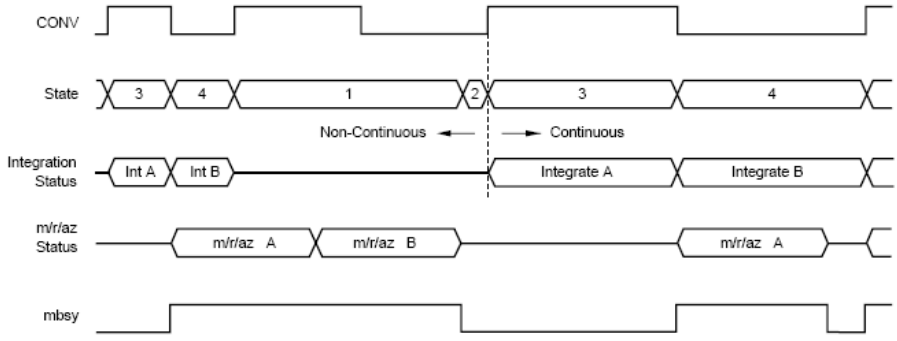


Fig. 4.18 *Passaggio dalla modalità "non-continua" alla modalità "continua"*

4.3.6 Utilizzo di più DDC112 in cascata

Un certo numero di convertitori DDC112 (dipendente dal tempo d'integrazione e dalla frequenza di trasmissione dei dati) può essere collegato in configurazioni serie o parallelo, come evidenziato nelle figure sottostanti. Il segnale DOUT può essere utilizzato insieme a DIN per collegare più DDC112 in configurazione *daisy-chain* (fig. 4.19), minimizzando le interconnessioni: in questa configurazione i dati escono in modo seriale, e vengono fatti scorrere nei vari componenti della catena. Le resistenze di *pull-up* evitano che DIN sia flottante quando il segnale di abilitazione alla trasmissione dei dati (DXMIT) non è attivo.

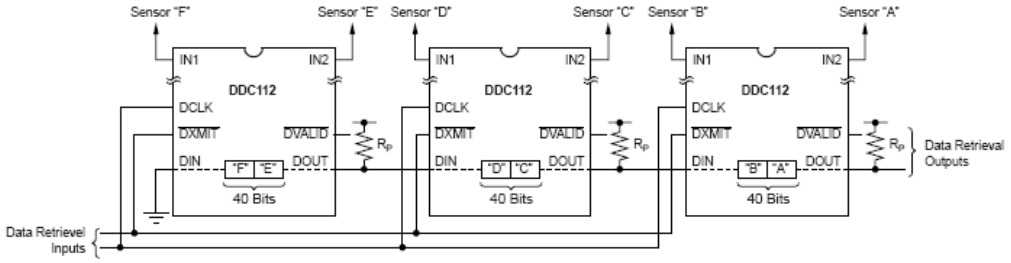


Fig. 4.19 *Connessione di più DDC112 in daisy-chain*

In caso di utilizzo di più DDC112 in parallelo (fig. 4.20), invece, il *pin* corrispondente a DIN deve essere messo a massa, come nel caso di utilizzo di un singolo DDC112.

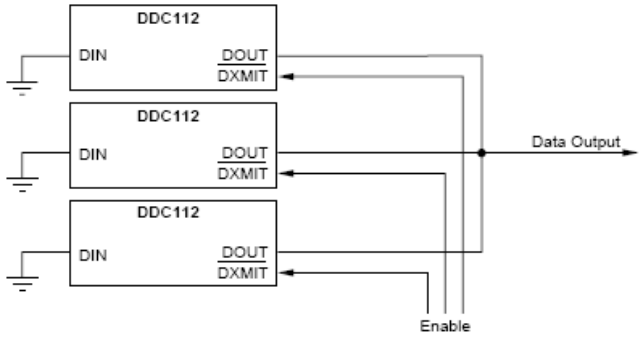


Fig. 4.20 *Connessione di più DDC112 in parallelo*

4.3.7 Trasferimento dei dati

In entrambe le modalità di operazione "continua" e "non-continua", i dati relativi all'ultima conversione sono disponibili per essere recuperati non appena si ha un fronte di discesa di DVALID (fig. 4.21). L'uscita seriale dei dati su DOUT è trasmessa in codice binario diretto.

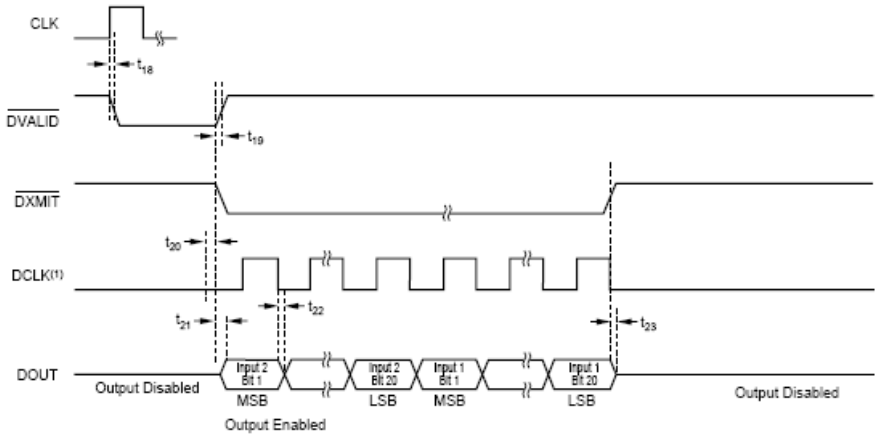


Fig. 4.21 *Diagramma di temporizzazione relativo al trasferimento di dati validi*

Il fronte di discesa di DXMIT in combinazione con DCLK darà il via alla trasmissione seriale dei dati in uscita dal DDC112. Tipicamente, i dati vengono recuperati dal DDC112 non appena si ha un fronte di discesa di DVALID, e il trasferimento si conclude prima della successiva commutazione alto-basso (o basso-alto) di CONV: nel caso in cui questo non si verifichi, occorre disabilitare DCLK (e conseguentemente DOUT) per almeno 10 μs a cavallo della successiva commutazione di CONV. Se non si adotta questa precauzione, è possibile un'introduzione di rumore nell'integrazione che parte al successivo fronte di CONV.
In modalità "continua", il metodo più semplice per il recupero dei dati è fare in modo che esso sia completato prima della commutazione di CONV: l'inizio della trasmissione avviene non appena DVALID diventa basso, mentre la fine si ha prima del successivo fronte di CONV (fig. 4.22).
Per ottenere migliori prestazioni, il recupero dei dati deve terminare un tempo t_{28} prima del successivo fronte di CONV. Il tempo utile al recupero dei dati è dato da $T_{INT} - t_{27} - t_{28}$: questo pone un limite al massimo numero di DDC112 che possono essere collegati in cascata in configurazione *daisy-chained*.

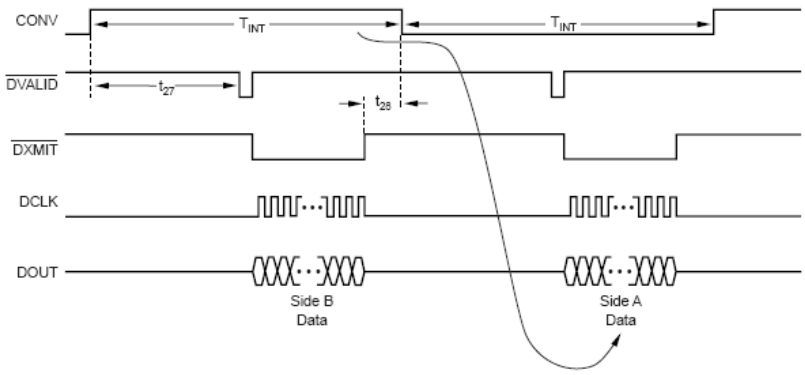


Fig. 4.22 *Recupero dati in modalità "continua"*

Ad esempio per CLK = 10 MHz, il numero massimo di convertitori utilizzabili in cascata è:

$$N_{MAX} = \frac{T_{INT} - 431.2\mu s}{40\tau_{DCLK}} \qquad \text{[IV. 2]}$$

Il recupero dei dati in modalità "non-continua" è leggermente diverso rispetto all'operazione in modalità "continua". Come mostrato in fig. 4.23, DVALID si attiva basso in un tempo t_{30} dal completamento della prima integrazione: se il tempo d'integrazione è minore di t_{30}, allora tutto il tempo t_{31} sarà disponibile per il recupero dei dati della prima integrazione, in quanto i dati (es. di A) sono pronti dopo il completamento della seconda integrazione (es. su B) . Se invece il tempo d'integrazione è maggiore di t_{30}, allora i dati della prima integrazione sono disponibili prima del completamento della seconda integrazione: in tal caso il recupero dei dati dev'essere ritardato, nell'attesa che la seconda integrazione si completi, e dunque il tempo disponibile al recupero dati non è più t_{31}, ma $t_{31} - (T_{INT} - t_{30})$. In entrambi i casi, comunque, i dati della seconda integrazione devono essere recuperati prima dell'inizio del nuovo ciclo d'integrazioni: questo tempo disponibile dipende molto dalla struttura del CONV scelto per implementare il funzionamento in modalità "non-continua".

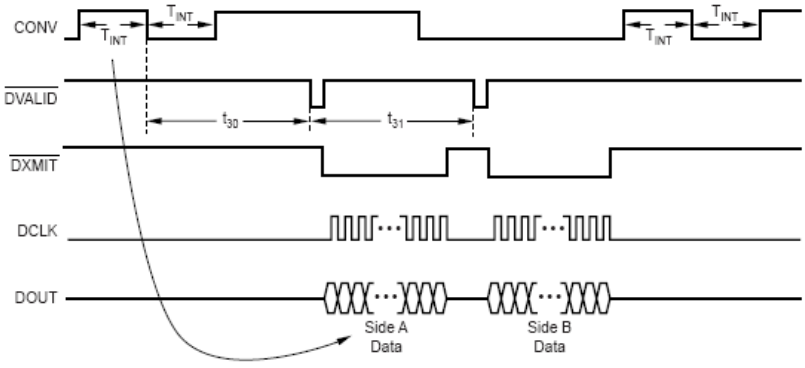


Fig. 4.23 *Recupero dati in modalità "non-continua"*

Come in modalità "continua", anche in modalità "non-continua" il recupero dei dati dev'essere sospeso nell'intorno di una commutazione di CONV, e deve completarsi prima che il DVALID segnali la disponibilità di un nuovo pacchetto di dati.

4.3.8 Sequenza di *power-up*

Prima dell'accensione (*power-up*) del dispositivo, tutti in *pin* d'ingresso analogici e digitali devono essere a livello logico basso; al momento dell'accensione, questi segnali d'ingresso possono essere portati ad una tensione diversa da 0 V, la quale in ogni caso non deve eccedere le tensioni di alimentazione analogica (AV_{DD}) e digitale (DV_{DD}). Il livello di CONV al *power-up* viene utilizzato per determinare quale lato (A o B) verrà

integrato per primo: in ogni caso, prima che le integrazioni abbiano inizio, CONV deve commutare (fig. 4.24).

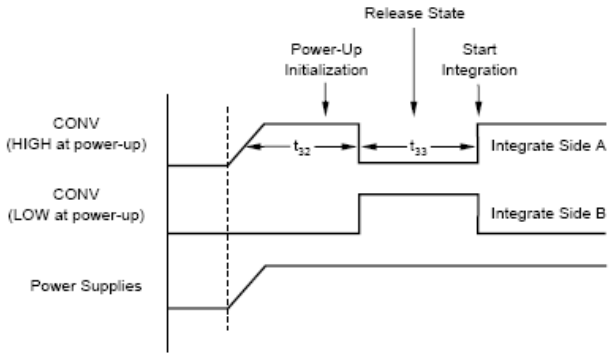


Fig. 4.24 *Sequenza di power-up del DDC112*

4.4 Microcontrollore LPC2103

Il microcontrollore LPC2103 è basato su una CPU (*Central Processing Unit*) ARM7TDMI-S a 32 *bit* capace di emulazioni in tempo reale; integra inoltre una memoria *flash* da 32 kB ad alta velocità. Un'interfaccia di memoria da 128 *bit*, combinata con un'architettura di accelerazione dedicata, consentono l'esecuzione di codici da 32 *bit* alla massima frequenza di *clock* (60 MHz): questo aumenta le prestazioni fino al 30% in più (rispetto alla modalità *Thumb*, vedi par. 4.4.2.5) in caso di esecuzione di operazioni critiche quali *routines* di *interrupt* o algoritmi DSP. Per applicazioni con codici di dimensioni critiche, la modalità *Thumb* alternativa a 16 *bit* riduce il codice di più del 30%, con un minimo calo delle prestazioni.
Grazie alle piccole dimensioni e al basso consumo di potenza, il μC LPC2103 è ideale per applicazioni in cui la miniaturizzazione è un fattore chiave, come il controllo di accessi o sistemi POS (*Point Of Sale*). La possibilità di scegliere fra diversi dispositivi di interfacciamento seriale (UART, SPI, SSP e I^2C) e una SRAM integrata da 8 kB rendono adatto il μC LPC2103 a porte di comunicazione e convertitori di protocollo. Infine, la presenza di una serie di *timer* a 16 o 32 *bit*, di un convertitore ADC a 10 *bit*, di 32 linee veloci GPIO e di 13 *pin* esterni di *interrupt*, rende questo microcontrollore particolarmente adatto ad applicazioni di controllo industriale e sistemi medicali.
Le caratteristiche del μC LPC2103 sono riassunte nel seguente elenco:

- Microprocessore ARM7TDMI-S a 32 *bit* incapsulato in un *package* LQFP48.
- SRAM integrata da 8 kB e memoria *flash* integrata programmabile da 32 kB.
- Interfaccia di memoria da 128 *bit* accelerata per operazioni ad alta velocità (60 MHz).
- *In-System/In-Application Programming* (ISP/IAP) attraverso un *software* integrato per il caricamento del *boot*: la cancellazione di un singolo settore *flash*

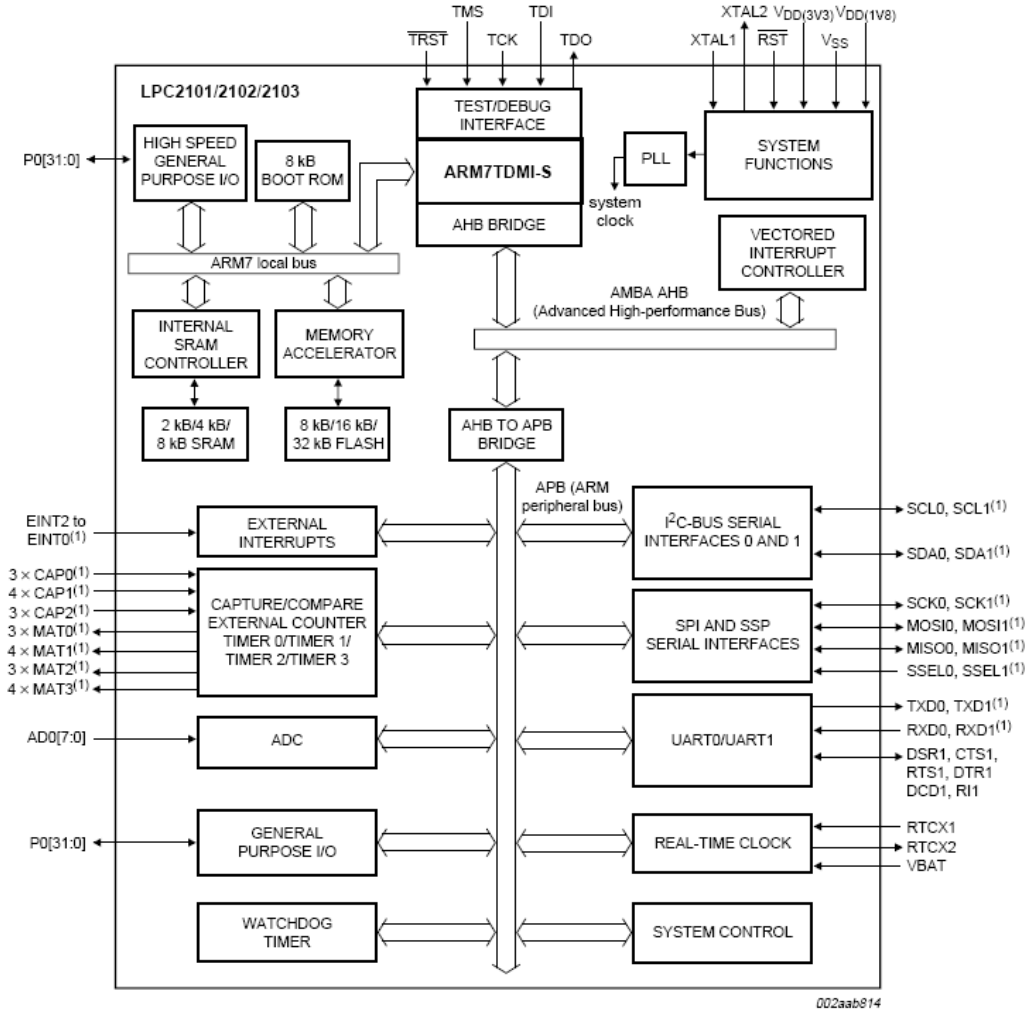
o dell'intero *chip* avviene in 100 ms, mentre la programmazione di 256 *bytes* avviene in 1 ms.

- *Debugging* in tempo reale.
- Convertitore A/D a 10 *bit* e 8 ingressi, con tempi di conversione fino a 2.44 μs per canale, e registri di uscita dedicati.
- Due *timer*/contatori di eventi esterni a 32 *bit* con sette canali di cattura e sette di comparazione.
- Due *timer*/contatori di eventi esterni a 16 *bit* con tre canali di cattura e sette di comparazione.
- *Real-Time Clock* (RTC) a basso consumo di potenza, alimentazione indipendente e un ingresso di *clock* dedicato a 32 kHz.
- Dispositivi d'interfacciamento seriale: due UART (16C550), due *bus* veloci I^2C (400 k*bit*/s), SPI e SSP con capacità di *buffering* e gestione di dati a lunghezza variabile.
- Controllore per *interrupt* vettorizzati (VIC) a priorità configurabile e indirizzo di vettore.
- 32 *pin* veloci I/O *general purpose* (GPIO), con una tensione massima applicabile di 5 V.
- 13 *pin* disponibili per ingressi esterni di *interrupt* (sensibili sul fronte o sul livello).
- *Clock* CPU (frequenza massima 60 MHz) disponibile tramite un PLL integrato programmabile con frequenza d'ingresso 10 ÷ 25 MHz e *settling time* di 100 μs.
- Oscillatore integrato su *chip* con quarzo esterno (frequenza 1 ÷ 25 MHz).
- Modalità di risparmio di potenza: *idle*, *power-down*, *power-down* con RTC attivo.
- Attivazione/disattivazione individuale di funzioni periferiche, e *clock* per le periferiche con frequenza programmabile per un'ottimizzazione addizionale del consumo di potenza.
- Riattivazione del processore dalla modalità *power-down* attraverso un *interrupt* esterno o RTC.

4.4.1 Architettura del microcontrollore LPC2103

Nella fig. 4.25, è possibile osservare lo schema a blocchi del microcontrollore LPC2103. Oltre alla CPU ARM7TDMI-S e ai vari dispositivi e periferiche elencati nel precedente paragrafo, si possono notare una serie di *bus* di comunicazione: il *bus* locale di ARM 7 per l'interfacciamento con i controllori di memoria *on-chip*, il *bus* AHB (*Advanced High-performance Bus*) per l'interfacciamento con il controllore degli *interrupt*, e il *bus* periferico APB (*Advanced Peripheral Bus*) per la connessione ai dispositivi periferici integrati.
Il μC LPC2103 configura il processore ARM7TDMI-S in ordine di *byte little-endian*. Le periferiche AHB sono allocate in un *range* di indirizzi da 2 MB in cima allo spazio totale di memoria da 4 GB del processore ARM: ogni periferica AHB è allocata in uno spazio da 16 kB all'interno dello spazio indirizzi AHB.

I dispositivi periferici del µC LPC2103 (oltre al controllore degli *interrupt*) sono connessi al *bus* APB. Il ponte AHB *to* APB interfaccia il *bus* APB con il *bus* AHB. Anche le periferiche APB sono allocate in un *range* d'indirizzi da 2 MB (che inizia nel punto d'indirizzo corrispondente a 3.5 GB), e ciascuna di esse è allocata in uno spazio da 16 kB all'interno dello spazio indirizzi APB.
La connessione dei dispositivi periferici integrati ai vari *pin* è controllata da un blocco di connessione *pin* (PCB), che dev'essere configurato tramite *software* per soddisfare i requisiti di applicazione specifici per l'utilizzo dei *pin* e delle funzioni periferiche.



(1) Pins shared with GPIO.

Fig. 4.25 *Architettura del microcontrollore LPC2103*

Nei successivi paragrafi verrà fatta una panoramica delle caratteristiche del μC LPC2103, con particolare riferimento alla descrizione della CPU ARM7 e delle periferiche, sia quelle di sistema che quelle di servizio.

4.4.2 Processore ARM7TDMI-S

L'ARM7TDMI-S è un processore a 32 *bit general purpose*, ad alte prestazioni e basso consumo di potenza [56]. L'architettura ARM è basata su principi RISC (*Reduced Instruction Set Computer*), con tutti i vantaggi che questo comporta rispetto ai sistemi CISC (*Complex Instruction Set Computer*) in termini di semplicità del *set* di istruzioni e del relativo meccanismo di decodifica, ovvero alto *throughput* di istruzioni e risposta in tempo reale agli *interrupt*.
La principale caratteristica del processore ARM7 sta nel fatto che è un'architettura *load-and-store* basata su registri, con un certo numero di modalità operative [57]. Nonostante ARM7 sia un processore a 32 *bit*, può eseguire anche istruzioni a 16 *bit*, denominate istruzioni *Thumb* (vedi par. 4.4.2.5): questo permette di avere un'alta densità di codice e ottenere un notevole risparmio di potenza. Il processore ARM7 integra inoltre una porta JTAG di *debug* e un emulatore integrato su *chip* (ICE, *In Circuit Emulator*) per permettere di scaricare i programmi ed effettuarne il *debug* internamente al sistema.

4.4.2.1 Pipeline

Per mantenere semplice l'architettura di ARM7, il codice e le zone riservate ai dati sono accessibili attraverso un singolo *data bus*: dunque, mentre tutte le istruzioni di elaborazione dati occupano un solo ciclo, altre istruzioni come il trasferimento di dati occupano più di un ciclo (almeno due accessi al *bus*, uno per l'istruzione e uno per i dati). Per ottimizzare le prestazioni delle istruzioni multiciclo, ARM7 è dotato di una *pipeline* a tre stadi, la più semplice forma di *pipeline* esistente, schematizzata in fig. 4.26: mentre un'istruzione viene eseguita (*execute*), quella successiva viene decodificata (*decode*), e una terza istruzione viene prelevata dalla memoria (*fetch*). In questo modo la *pipeline* aumenta il *throughput* delle istruzioni della CPU, cosicché la maggior parte di esse può essere eseguita in un unico ciclo: tutte le parti del processore e della memoria lavorano in maniera continua.

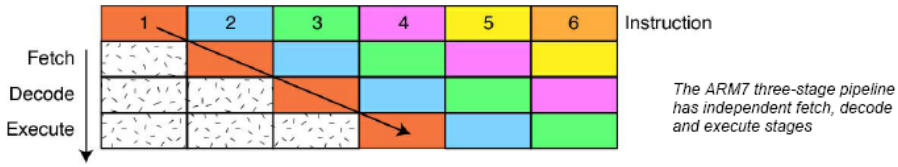


Fig. 4.26 *Pipeline a tre stadi*

4.4.2.2 Registri

L'ARM7 è un'architettura *load-and-store*: quando si devono eseguire istruzioni relative ad un dato, il dato viene dapprima trasferito dalla memoria ad un *set* centrale di registri, poi vengono eseguite le istruzioni, ed infine il dato viene ritrasferito nella memoria. Al

contrario di ciò che avviene in altri processori, non sono disponibili istruzioni da memoria a memoria.
Il *set* centrale di registri (fig. 4.27) è costituito da 16 registri R0 – R15, ognuno da 32 *bit*: i registri R0 – R12 sono di servizio, nel senso che non hanno una funzione specifica, mentre i registri R13 – R15 svolgono funzioni speciali per la CPU. R13 è usato come *stack pointer* (SP), mentre R14 è usato come *link register* (LR). Quando si chiama una funzione, l'indirizzo di ritorno è automaticamente memorizzato in LR ed è immediatamente disponibile al ritorno dalla chiamata della funzione: questo permette di entrare e uscire velocemente da una funzione "foglia" (ovvero una funzione che non chiamerà ulteriori funzioni); se la funzione è invece parte di un "ramo" (ovvero chiamerà altre funzioni), allora il contenuto di LR deve essere trasferito in SP. Infine, R15 è usato semplicemente come *program counter* (PC).

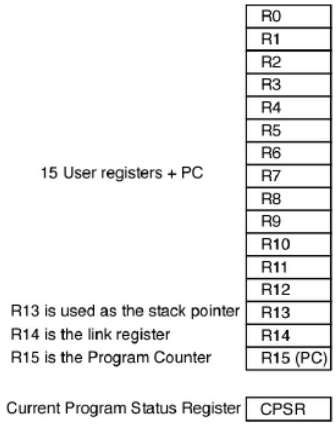


Fig. 4.27 *Set di registri del processore ARM 7*

Oltre al *set* centrale di registri, c'è un registro addizionale da 32 *bit* (CPSR, *Current Program Status Register* (CPSR), contenente un certo numero di *flags* (fig. 4.28) riguardanti le operazioni della CPU.

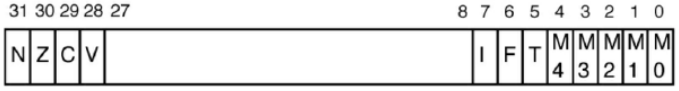


Fig. 4.28 *Flags del registro CPSR*

I primi 4 *bit* del registro CPSR (da 31 a 28) contengono informazioni sul risultato di un'operazione su un dato, ovvero se è stato generato un risultato negativo (N), nullo (Z), oppure un *carry* (C), o infine se si è verificato un *overflow* (V). Gli ultimi 8 *bit* contengono *flags* di servizio: i *bit* 7 e 6 (I e F) sono usati per abilitare o disabilitare i due generatori di *interrupt* esterni alla CPU, ai quali sono collegate tutte le periferiche del μC LPC2103. Il *bit* 5 è il *bit Thumb* (T), e indica quale *set* di istruzioni la CPU di ARM7

sta eseguendo, se il *set* di istruzioni ARM a 32 *bit* oppure il *set Thumb* da 16 *bit* (vedi par. 4.4.2.5).
Infine, gli ultimi 5 *bit* (da 4 a 0, M4 – M0) indicano la modalità operativa in cui sta lavorando il processore ARM7: oltre alla modalità *user*, ovvero alla modalità standard con accesso ai registri R0 – R15 e CPSR, il processore può infatti lavorare in altre 5 modalità (dette "privilegiate") alle quali passa in risposta ad un'eccezione, ad esempio un *interrupt*, un errore di memoria, o un istruzione di *interrupt* via *software*.
Le eccezioni per il processore ARM7 si possono dividere in tre categorie:

- Eccezioni causate dall'esecuzione di un'istruzione: includono *interrupt* via *software*, istruzioni indefinite o errori di memoria (*memory abort*).
- Eccezioni che rappresentano un effetto collaterale dell'esecuzione di un'istruzione, come un *data abort* causato dall'aver cercato di prelevare un dato da una locazione di memoria non valida.
- Eccezioni non correlate con l'esecuzione di un'istruzione, come il *reset* o un *interrupt* IRQ e FIQ.

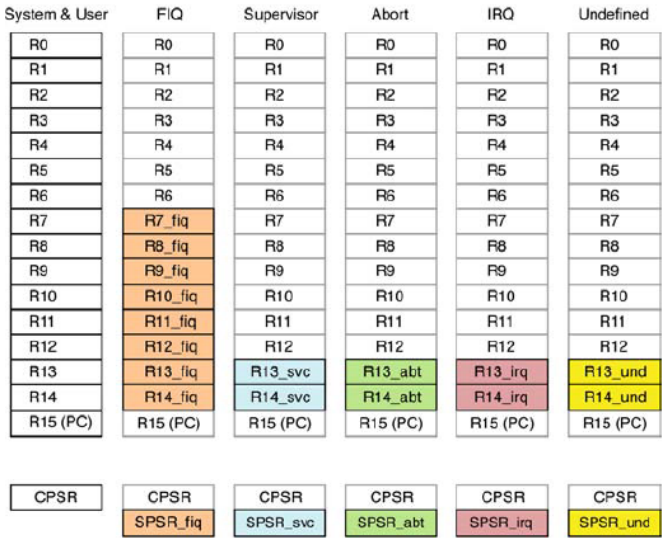


Fig. 4.29 *Configurazione nei registri nelle diverse modalità*

Al passaggio ad una modalità privilegiata, i registri R0 – R12 e R15 (PC) rimangono gli stessi, ma R13 (LR) e R14 (SP) sono sostituiti da una nuova coppia di registri caratteristici della nuova modalità (fig. 4.29). Inoltre la modalità FIQ (*Fast Interrupt Mode*) ha doppi registri R7 – R12, consentendo di entrare rapidamente in modalità FIQ senza necessità di mantenere i registri sullo *stack*.
Infine, ognuna delle 5 modalità alternative ha un registro addizionale, indicato con SPSR (*Saved Program Status Register*), nel quale, in caso di arrivo di un'eccezione e dunque di un cambio di modalità, verrà salvato il contenuto del registro CPSR: una volta

eseguito il codice relativo all'eccezione, il contenuto di SPRS verrà ritrasferito in CPSR. In fig. 4.29 sono evidenziate le diverse modalità operative di ARM7, con le rispettive configurazioni dei registri.

4.4.2.3 Modalità privilegiate

In caso di arrivo di un'eccezione, come anticipato nel precedente paragrafo, la CPU passa dalla modalità *user* ad una delle 5 modalità privilegiate:

- FIQ (*Fast Interrupt reQuest*): gestisce gli *interrupt* ad alta velocità. E' di solito usata per un singolo generatore critico di *interrupt* interno ad un sistema.
- IRQ (*Interrupt ReQuest*): gestisce tutte gli altri generatori di *interrupt* di un sistema.
- *Supervisor*: è una modalità protetta per l'esecuzione di codici di accesso all'*hardware*, oppure di funzioni del sistema operativo. ARM7 entra nella modalità *supervisor* dopo il *reset*.
- *Abort*: se un'istruzione (*prefetch abort*) o un dato (*data abort*) vengono prelevati da una locazione di memoria non valida, viene generata un'eccezione "*abort*".
- *Undefined instruction*: se un *opcode* (gruppo di *bit* indicativi dell'operazione da compiere) prelevato dalla memoria non corrisponde a nessuna istruzione ARM, viene generata un'eccezione di "istruzione non definita".

Ogni tipologia di eccezione ha associato un vettore (detto vettore di eccezione), prontamente memorizzato nel registro PC all'arrivo dell'eccezione. La tavola dei vettori di eccezione inizia dall'indirizzo 0 (che corrisponde al vettore di *reset*), dopo di che si ha un vettore di eccezione ogni 4 *bytes*. Se si verifica l'arrivo di più eccezioni in contemporanea, c'è una priorità fissa che viene rispettata, come evidenziato in fig. 4.30.

Exception	Mode	Address
Reset	Supervisor	0x00000000
Undefined instruction	Undefined	0x00000004
Software interrupt (SWI)	Supervisor	0x00000008
Prefetch Abort (instruction fetch memory abort)	Abort	0x0000000C
Data Abort (data access memory abort)	Abort	0x00000010
IRQ (interrupt)	IRQ	0x00000018
FIQ (fast interrupt)	FIQ	0x0000001C

Priority	Exception
Highest 1	Reset
2	Data Abort
3	FIQ
4	IRQ
5	Prefetch Abort
Lowest 6	Undefined instruction SWI

Fig. 4.30 *Tavola dei vettori di eccezione e priorità delle eccezioni*

Si può notare l'assenza del vettore d'indirizzo 0x00000014: questa locazione veniva utilizzata nei primi processori ARM, ma è stata mantenuta in ARM7 per assicurare compatibilità *software* fra le diverse versioni (nel μC LP2103 questi 4 *bytes* sono comunque usati per altri scopi).
Quando avviene un'eccezione (es. IRQ), prima di tutto l'indirizzo della successiva istruzione da eseguire viene salvato nel *link register*. Poi il registro CPSR viene copiato

nel registro SPSR della modalità privilegiata in cui si sta per entrare (es. SPSR_irq). Dopo di che nel registro PC viene memorizzato l'indirizzo del vettore associato all'eccezione (che per IRQ è 0x00000018). Allo stesso tempo avviene il passaggio alla modalità IRQ, con i registri R13 e R14 sostituiti dai registri R13_irq e R14_irq. A questo punto viene eseguita la *routine* di servizio (ISR) relativa all'eccezione IRQ: la prima cosa che il codice deve prevedere è preservare ognuno dei registri R0 – R12 che la ISR userà, ponendoli in cima allo *stack* IRQ (fatto questo, si può cominciare a processare l'eccezione).
Una volta terminata la ISR, si deve tornare alla modalità *user*, e il codice *user* deve continuare da dove era stato lasciato all'arrivo dell'eccezione. Il *set* di istruzioni ARM, però, non prevede un'istruzione specifica di "ritorno dall'*interrupt*", cosicché il problema dev'essere affrontato in maniera diversa a seconda dell'eccezione dalla quale si deve tornare: ognuna delle modalità privilegiate ha la sua ben precisa istruzione di ritorno (es. `SUBS R15, R14, #4` per le modalità IRQ e FIQ).
Una volta eseguita l'istruzione di ritorno, i contenuti modificati del *link register* vengono ritrasferiti nel registro PC, mentre il contenuto del registro SPSR viene ritrasferito nel registro CPSR (in caso di eccezioni IRQ e FIQ vengono anche riabilitate le rispettive linee di *interrupt*): questo conclude la procedura di uscita dalla modalità privilegiata e il ritorno alla modalità *user*.

4.4.2.4 Set di istruzioni ARM

La CPU di ARM7 ha due *set* di istruzioni: ARM (a 32 *bit*) e *Thumb* (a 16 *bit*). In questo paragrafo verrà presentato il solo *set* di istruzioni ARM (per le istruzioni *Thumb* vedi par. 4.4.2.5).
Sebbene il processore ARM7 sia stato progettato per lavorare sia in modalità *little endian* che *big endian* (fig. 4.31), ovvero il *bit* più significativo (MSB) può essere quello di ordine maggiore (*little*) o quello di ordine minore (*big*), la famiglia di μC LPC2000 (di cui ARM7 è il cuore) utilizza la modalità *little endian.*

Fig. 4.31 *Modalità little endian e big endian*

Una delle caratteristiche più interessanti delle istruzioni ARM è che, a differenza di quasi tutti i microprocessori tradizionali, ogni istruzione può essere eseguita sotto condizione [58]: i primi 4 *bit* dell'operando sono confrontati con i primi 4 *bit* del registro CPSR (le *condition code flags*), e se il confronto non va a buon fine l'istruzione non viene eseguita, e passa attraverso la *pipeline* come NOP (*no operation*).
Lo scopo di questa esecuzione condizionata delle istruzioni è di mantenere un flusso omogeneo di istruzioni attraverso la *pipeline*. Nei processori tradizionali, ogni volta che nel codice c'è un salto, la *pipeline* viene svuotata per poi essere riempita di nuovo, e questo causa un calo nella prestazione complessiva; nel processore ARM, invece, la

presenza di un salto nel codice non prevede lo svuotamento-riempimento della *pipeline*, ma una serie di NOP successive dal punto di partenza al punto di arrivo del salto. Ovviamente c'è un punto di equilibrio fra i due metodi in termini di prestazione, che nel caso di *pipeline* a 3 stadi è proprio di 3 istruzioni: per piccole ramificazioni (meno di 3 istruzioni), come ad esempio:

```
if( x<100)
{
x++;
}
```

è più efficiente l'utilizzo delle esecuzioni condizionate tipiche dell'ARM, mentre per *branch* che prevedono più di 3 istruzioni è più conveniente svuotare e poi riempire la *pipeline* (altrimenti si avrebbe un numero eccessivo di NOP forzate che diminuirebbe ancora di più le prestazioni).
In fig. 4.32 sono riportati i possibili codici condizionali e le relative *flags*:

Suffix	Flags	Meaning
EQ	Z set	equal
NE	Z clear	not equal
CS	C set	unsigned higher or same
CC	C clear	unsigned lower
MI	N set	negative
PL	N clear	positive or zero
VS	V set	overflow
VC	V clear	no overflow
HI	C set and Z clear	unsigned higher
LS	C clear and Z set	unsigned lower or same
GE	N equals V	greater or equal
LT	N not equal to V	less than
GT	Z clear AND (N equals V)	greater than
LE	Z set OR (N not equal to V)	less than or equal
AL	(ignored)	always

Fig. 4.32 *Codici condizionali e relative flags*

Ad esempio, l'istruzione ADD può essere preceduta da un codice condizionale, come nell'esempio qui di seguito:

```
EQADD R1, R2,R3 ; If ( Zero flag = 1) then R1 = R2+R3
```

Dal commento si può notare come l'istruzione venga eseguita solo se il *bit* Z (*Zero flag*) è uguale a 1, condizione corrispondente al prefisso EQ.
Inoltre, il programmatore può decidere se una particolare istruzione può settare la *flag* del CPSR corrispondente al codice condizionale, come in questo secondo esempio:

```
EQADDS R1,R2 R3, LSL #2 ;  If (Zero flag = 1) then R1 = R2+(R3 x 4)
                                  and set condition code flags
```

I gruppi principali di istruzioni del *set* ARM appartengono a 6 differenti categorie: salto (*branching*), elaborazione dati (*data processing*), trasferimento dati (*data transfer*), trasferimento blocchi di dati (*block transfer*), moltiplicazione (*multiply*) e interruzione *software* (*software interrupt*).

4.4.2.4.1 Branching

Le istruzioni di *branch* sono di 4 diverse tipologie: l'istruzione "*branch*" di base (fig. 4.33a) permette un salto ad un indirizzo di destinazione che si trova avanti o indietro al massimo di 32 MB. Una versione modificata dell'istruzione di base, detta *branch link* (fig. 4.33b) consente il salto e inoltre memorizza nel *link register* R14 l'indirizzo di ritorno del PC corrente più 4 *bytes*, in modo tale da permettere anche il ritorno dal salto.

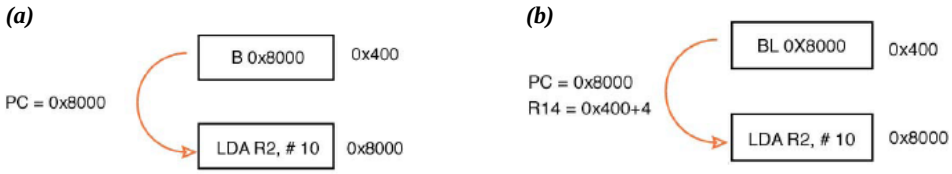


Fig. 4.33 *Istruzioni di branch (a) e branch link (b)*

Con l'utilizzo dei codici condizionali è possibile anche operare un *branching*, o chiamare una funzione, in modo condizionale.
Le istruzioni di *branch* hanno altre due varianti, chiamate *branch exchange* (fig. 4.34a) e *branch link exchange* (fig. 4.34b), con le quali si eseguono le stesse operazioni di *branch* illustrate in precedenza, e inoltre si effettua un passaggio dal *set* d'istruzioni ARM al *set* d'istruzioni *Thumb* (e viceversa): questo è l'unico modo per scambiare i due *set* d'istruzioni, perché agire manualmente sul *bit* "T" del registro CPSR può portare a risultati imprevedibili.

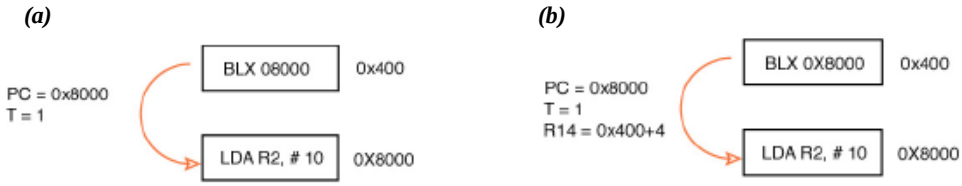


Fig. 4.34 *Istruzioni di branch exchange (a) e branch link exchange (b)*

4.4.2.4.2 Elaborazione dati

La forma generica di tutte le istruzioni di elaborazione dati è riportata in fig. 4.35. Ogni istruzione ha un registro di risultato e due operandi: il primo operando dev'essere un registro, il secondo può essere un registro o un valore istantaneo. Inoltre, ARM7

contiene uno *shifter* che permette al secondo operando di essere traslato di 32 *bit* all'interno del ciclo d'istruzione.

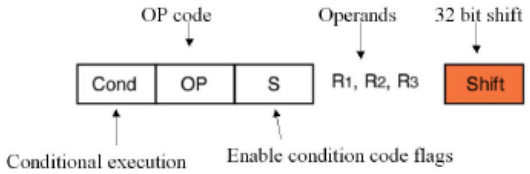


Fig. 4.35 *Forma generale delle istruzioni di elaborazione dati*

Il *bit* "S" è usato per il controllo dei codici condizionali: se S = 1 i codici condizionali vengono modificati in base al risultato dell'operazione, altrimenti (S = 0) non avviene alcuna modifica. Se, tuttavia, si specifica R15 (registro *Program Counter*) come registro di risultato e S = 1, allora il registro SPSR relativo alla modalità privilegiata corrente verrà copiato nel registro CPSR: questa operazione viene compiuta alla fine di un'eccezione, per ripristinare il *Program Counter* e tornare alla modalità *user*.
Qui di seguito è elencato il *set* di istruzioni per l'elaborazione dei dati:

```
AND             Logical bitwise AND
EOR             Logical bitwise exclusive OR
SUB             Subtract
RSB             Reverse Subtract
ADD             Add
ADC             Add with carry
SBC             Subtract with carry
RSC             Reverse Subtract with carry
TST             Test
TEQ             Test Equivalence
CMP             Compare
CMN             Compare negated
ORR             Logical bitwise OR
MOV             Move
BIC             Bit clear
MVN             Move negated
```

4.4.2.4.3 Copia di registri

Il gruppo successivo di istruzioni ARM è relativo al trasferimento dati: la CPU di ARM7 è dotata di registri *load-and-store* che possono trasferire parole (con o senza segno), metà parole, oppure *bytes* di dati a (o da) un registro selezionato. Poiché il *set* di registri è completamente ortogonale, è possibile caricare un valore da 32 *bit* nel *Program Counter*, forzando un salto ovunque all'interno dello spazio indirizzi del microprocessore. Oltre a caricare e immagazzinare singoli valori di registro, l'ARM è dotato di istruzioni per il *load-and-store* di registri multipli: in tal modo è possibile copiare nella memoria, con una singola istruzione, l'intero blocco di registri (o una sua parte), e ripristinarlo con una seconda istruzione. Qui di seguito è elencato il *set* di istruzioni per il *load-and-store*:

```
LDR             Load Word
LDRH            Load Half Word
LDRSH           Load Signed Half Word
LDRB            Load Byte
LRDSB           Load Signed Byte

STR             Store Word
STRH            Store Half Word
STRSH           Store Signed Half Word
STRB            Store Byte
STRSB           Store Signed Half Word
```

4.4.2.4.4 Istruzioni di scambio (swap)

L'istruzione di scambio (*swap*) effettua in tempo reale lo scambio di una parola fra due registri (fig. 4.36): questa operazione richiede due cicli, ma viene trattata come una singola istruzione, in modo tale che (specialmente nel caso di scambio di dati importanti) lo scambio non possa essere messo a repentaglio da eventuali eccezioni (come un *interrupt*).

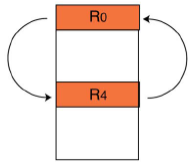


Fig. 4.36 *Istruzione di swap*

L'istruzione di *swap* non è impostabile attraverso il linguaggio C usato dall'utente per la programmazione: è supportata invece da funzioni intrinseche speciali all'interno delle librerie di compilazione.

4.4.2.4.5 Modifica dei registri di stato

Come descritto nel par. 4.4.2.2, CPSR e SPSR sono registri di CPU, e non fanno parte del *set* centrale di registri. Le uniche due istruzioni ARM che possono operare direttamente su questi due registri sono MSR (fig. 4.37a) e MRS (fig. 4.37b), che permettono di trasferire il contenuto dei registri CPSR o SPSR a (o da) un registro selezionato: per esempio, per disabilitare un *interrupt* IRQ il contenuto di CPSR dev'essere trasferito ad un registro, il *bit* "I" dev'essere posto uguale a 1 (facendo un AND fra il contenuto del registro e 0x00000080) per disabilitare l'*interrupt*, e infine CPSR dev'essere riprogrammato con il nuovo valore.

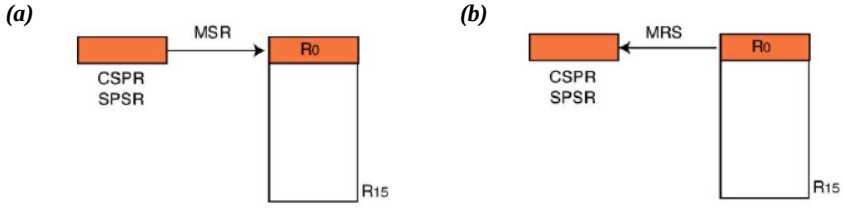


Fig. 4.37 *Istruzioni MSR (a) e MRS (b)*

Le istruzioni MSR e MRS sono operative solo nelle modalità privilegiate (quindi tutte tranne la modalità *user*), e dunque solo in questi casi è possibile cambiare modalità, e abilitare (o disabilitare*) interrupt*. Pertanto non si può abbandonare la modalità *user* se non attraverso un'eccezione (*reset*, FIQ, IRQ o SWI).

4.4.2.4.6 Software interrupt

L'istruzione di *software interrupt* (SWI) genera un'eccezione durante un'esecuzione, forza il processore ad entrare nella modalità *supervisor*, e fa saltare il *Program Counter* a 0x00000008. Come tutte le altre istruzioni ARM, anche l'istruzione SWI (vedi fig. 4.38) contiene nei primi 4 *bit* i codici di esecuzione condizionale, seguiti dal codice OP (indicativo dell'operazione da eseguire), anch'esso di 4 *bit* . Il resto dell'istruzione (24 *bit*) è vuoto, ma è possibile sfruttare i *bit* inutilizzati per la codifica di un numero.

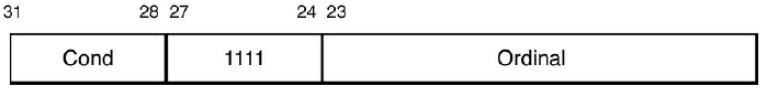


Fig. 4.38 *Struttura dell'istruzione di software interrupt*

Eseguendo un'istruzione SWI, questo numero verrà esaminato per determinare quale procedura SWI seguire (fra 16 milioni di possibili funzioni SWI): in tal modo, è possibile usare l'istruzione SWI nel codice *user* per chiamare funzioni da eseguire, invece che in modalità *user* , in modalità protetta (*supervisor*).

4.4.2.4.7 Unità MAC

Oltre allo *shifter* da 32 *bit* (vedi par. 4.4.2.4.2), il processore ARM7 integra un unità MAC (*Multiply Accumulate Unit*), che permette di eseguire moltiplicazioni fra interi (in formato *integer* o *long integer*). L'istruzione di moltiplicazione in formato *integer* permette di moltiplicare due registri da 32 *bit* e porre il risultato in un terzo registro, anch'esso da 32 *bit*: dopo di che un'istruzione di accumulazione (*multiply-accumulate*) può prendere il prodotto ottenuto e aggiungerlo ad un totale.
L'istruzione di moltiplicazione in formato *long integer* permette di moltiplicare due registri da 32 *bit* e porre il risultato, che stavolta è di 64 *bit*, in due registri da 32 *bit* ciascuno: anche in questo caso è disponibile l'istruzione *multiply-accumulate*.
Qui di seguito sono elencate tutte le istruzioni relative all'unità MAC:

MUL	Multiply	32 bit result
MULA	Multiply accumulate	32 bit result
UMULL	Unsigned multiply	64 bit result
UMLAL	Unsigned multiply accumulate	64 bit result
SMULL	Signed multiply	64 bit result
SMLAL	Signed multiply accumulate	64 bit result

4.4.2.5 Set di istruzioni Thumb

Sebbene ARM7 sia un processore a 32 *bit*, prevede un secondo *set* di istruzioni a 16 *bit* chiamato *Thumb*, che in realtà è una forma compressa del *set* di istruzioni ARM: questo permette alle istruzioni di essere immagazzinate in formato 16 *bit*, convertite in istruzioni ARM a 32 *bit*, e infine eseguite (fig. 4.39).

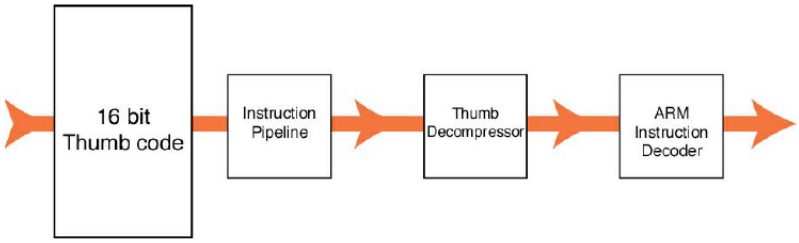


Fig. 4.39 *Passaggio da un'istruzione Thumb a un'istruzione ARM*

Nonostante le istruzioni *Thumb* determinino un calo nelle prestazioni del codice rispetto alle istruzioni ARM, permettono comunque di raggiungere una densità di codice molto più alta (e un notevole risparmio di potenza): un codice compilato con istruzioni *Thumb* permette di risparmiare fino al 30% di spazio, mentre lo stesso codice compilato con istruzioni ARM viene eseguito il 40% più velocemente. Ad ogni modo, al fine di realizzare un'applicazione di dimensioni ragionevoli, e che soprattutto sia possibile implementare con un piccolo microprocessore a singolo *chip*, è di vitale importanza compilare il codice con un *mix* di istruzioni ARM e *Thumb*. Questo processo è denominato *interworking*, ed è supportato da tutti i compilatori ARM.
Il *set* di istruzioni *Thumb* è molto simile al *set* di istruzioni di un microprocessore tradizionale: a differenza delle istruzioni ARM, le istruzioni *Thumb* non sono eseguite sotto condizione (a parte le istruzioni di *branch*). Le istruzioni di elaborazione dati hanno un formato a due indirizzi, e il registro di destinazione è uno dei registri sorgente, ad esempio:

ARM Instruction	**THUMB Instruction**	
`ADD R0, R0,R1`	`ADD R0,R1`	`R0 = R0+R1`

Il *set* di istruzioni *Thumb* non ha accesso a tutti i registri del *set* centrale: tutte le istruzioni di elaborazione dati hanno accesso ai soli registri R0 – R7 (i cosiddetti "registri bassi"). Solo alcune particolari istruzioni (MOV, ADD e CMP) hanno accesso ai "registri alti" (R8 – R12).
Il *set* di istruzioni *Thumb* non contiene istruzioni di modifica dei registri di stato (MSR e MRS), e dunque i registri CPSR e SPSR si possono gestire solo indirettamente: se si vuole modificare uno qualsiasi dei *bit* di servizio del registro CPSR, occorre passare ad istruzioni ARM. Si può cambiare *set* di istruzioni attraverso le istruzioni BX (passaggio da ARM a *Thumb*) e BLX (passaggio da *Thumb* a ARM). Anche quando si esce da un *reset*, o si entra in una modalità privilegiata, si passa automaticamente alle istruzioni ARM.
Il *set* di istruzioni *Thumb* comprende anche le tradizionali istruzioni PUSH e POP per la manipolazione dello *stack* (fig. 4.40): viene implementato uno *stack* totalmente discendente, utilizzando R13 come *stack pointer* (SP):

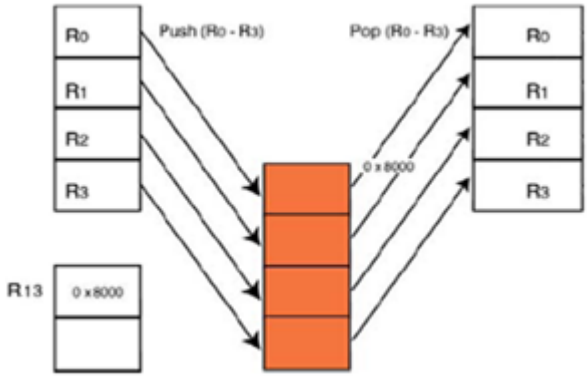


Fig. 4.40 *Istruzioni Thumb PUSH e POP*

Infine, il *set* di istruzioni *Thumb*, contiene un istruzione di *software interrupt* (SWI) che lavora allo stesso modo del caso ARM, ma contiene solo 8 *bit* inutilizzati (anziché 24), consentendo dunque solo 255 possibili funzioni SWI.

4.4.3 Periferiche di sistema

In questa sezione verranno presentate le periferiche di sistema del μC LP2103, ovvero le parti utilizzate per il controllo delle prestazioni e delle caratteristiche funzionali del dispositivo: queste includono le memorie integrate *flash* e SRAM, il circuito ad aggancio di fase (PLL, *Phase Locked Loop*) utilizzato per moltiplicare la frequenza dell'oscillatore esterno al fine di ottenere un frequenza di *clock* massima di 60 MHz, e infine le periferiche per il controllo della potenza. Infine, verrà data una descrizione del sistema di *interrupt* del μC LP2103, con particolare riferimento ai *pin* riservati agli *interrupt* esterni.

4.4.3.1 Struttura dei bus

Il *core* del processore ARM7 è connesso al *bus* AHB (*Advanced High performance Bus*), che rappresenta la via più veloce per connettere il processore con le periferiche di sistema. Connessi al *bus* AHB (fig. 4.41) ci sono anche il controllore per *interrupt* vettorizzati (VIC), responsabile della gestione di tutti i generatori di *interrupt*, e un ponte per un secondo *bus*, l'APB (*Advanced Peripheral Bus*), anche detto VPB (*VLSI Peripheral Bus*), al quale sono connesse tutte le restanti periferiche di sistema.
Il ponte AHB/APB contiene un divisore di *clock*, e dunque il *bus* APB è più lento rispetto al processore ARM7 e al *bus* AHB (più precisamente il *bus* APB lavora di *default* a ¼ della frequenza di *clock* della CPU). Questo è utile per due motivi: 1) le periferiche di sistema lavorano più lentamente rispetto ad ARM7 per risparmiare potenza; 2) c'è la possibilità di aggiungere periferiche lente senza il rischio di creare "colli di bottiglia" sul *bus* veloce AHB. In ogni caso, per periferiche di sistema ad alta

velocità, è sempre possibile fare in modo che la frequenza di *clock* del *bus* APB sia uguale a quella del *bus* AHB.

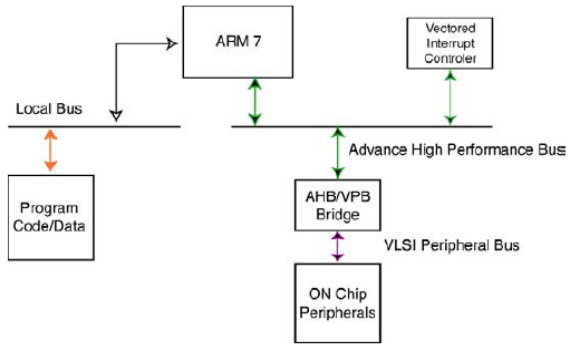


Fig. 4.41 *Struttura dei bus del μC LPC2103*

Infine, un terzo *bus* locale connette alla CPU le memorie integrate *flash* e SRAM. Sarebbe possibile in teoria usare per questa connessione il *bus* veloce AHB, ma questo introdurrebbe stalli di esecuzione a causa di contese su *bus*: per questo è meglio comunque usare, per l'invio alla CPU del codice di programma e dei dati in memoria, un *bus* locale separato.

4.4.3.2 Mappa di memoria

Nonostante la presenza di un certo numero di *bus* interni, il μC LPC2103 ha una mappa di memoria, mostrata in fig. 4.42, completamente lineare.

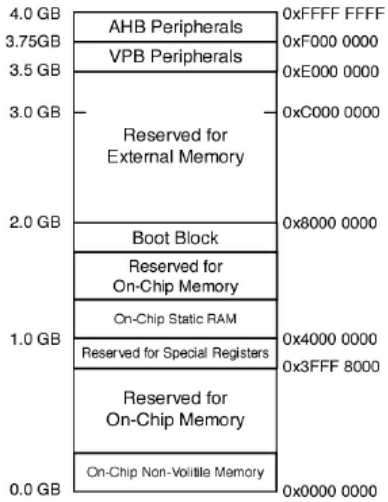


Fig. 4.42 *Mappa di memoria del μC LPC2103*

La memoria integrata *flash* è allocata all'indirizzo 0x00000000, mentre la SRAM è fissata all'indirizzo 0x40000000. Il μC LPC2103 è pre-programmato con un caricatore

di *boot* FLASH e con il programma ARM per il *debug* in tempo reale: a questi programmi sono destinate le locazioni nell'intervallo 0x7FFFFFFF ÷ 0x80000000. La regione fra 0x80000000 e 0xE0000000 è riservata alla memoria esterna.
Le periferiche di servizio (*user*) connesse al *bus* APB sono tutte mappate nella regione fra 0xE0000000 e 0xE0200000, e ad ogni periferica è riservata una pagina di memoria da 16 kB. Infine, il controllore degli *interrupt* vettorizzati VIC è allocato in cima al *range* degli indirizzi (0xFFFFF000).
Se il codice utente cerca di accedere alla memoria al di fuori di queste regioni, oppure viene specificato un indirizzo di memoria inesistente, verrà generata dalla CPU un'eccezione di tipo *abort*: è una funzione cablata a livello *hardware* nel processore, e dunque non può essere né modificata né interdetta.

4.4.3.3 Programmazione dei registri

In questo paragrafo verrà descritto brevemente come sono programmati sui *chip* ARM7 i registri per funzioni speciali (SFR, *Special Function Register*). Come regola generale, ogni SFR è controllato da tre registri di servizio (*user*): un registro *Set* è usato per settare i *bit* selezionati, un registro *Clear* è usato per resettare i *bit* voluti (N.B. per resettare un *bit* occorre scrivere un 1 logico), e infine un registro *Status* è usato per leggere il contenuto corrente del registro SFR. In fig. 4.43 è riportato un esempio:

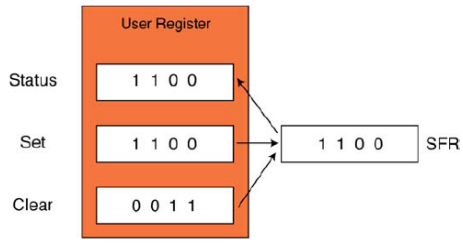


Fig. 4.43 *Struttura di un registro per funzioni speciali*

4.4.3.4 Modulo di accelerazione di memoria

Il MAM (*Memory Acceleration Module*) consente di aumentare la frequenza di esecuzione delle istruzioni del processore ARM7. Il MAM è connesso al *bus* locale, ed è situato fra la memoria *flash* e la CPU.
Uno dei limiti che possono determinare un calo delle prestazioni del μC è il tempo di accesso alla memoria *flash*. La CPU di ARM7 può lavorare fino a 60 MHz, mentre il tempo di accesso alla memoria *flash* è di 50 ns: dunque l'accesso diretto alla *flash* limiterebbe la velocità di esecuzione della CPU a 20 MHz.
Ci sono una serie di possibili soluzioni al problema: la più semplice è caricare sezioni critiche del programma dentro la RAM, che ha un tempo di accesso molto minore rispetto alla *flash*. Ad ogni modo, usare la RAM per memorizzare istruzioni di programma non conviene, visto che sarebbe inevitabilmente minore lo spazio riservato al codice che vogliamo eseguire.
Un altro approccio potrebbe essere quello di integrare una memoria *cache*, ovvero una piccola memoria veloce (posta fra il processore e la memoria principale) che carica le informazioni più recenti scambiate con la memoria principale. La memoria *cache* è però

una periferica complessa, che richiede un gran numero di porte, e di conseguenza occuperebbe molto spazio sul *chip*; un altro problema è che il tempo di esecuzione di un codice che fa uso di una memoria *cache* non è più un valore deterministico e prevedibile, e questo è un problema per tutte le applicazioni che richiedono ripetibilità.
Il MAM rappresenta un compromesso fra la complessità di una memoria *cache* e la semplicità di un accesso diretto del processore alla memoria *flash*. Come una *cache*, il MAM cerca di immagazzinare nella sua memoria locale la prossima istruzione ARM in tempo utile per la sua esecuzione da parte della CPU.

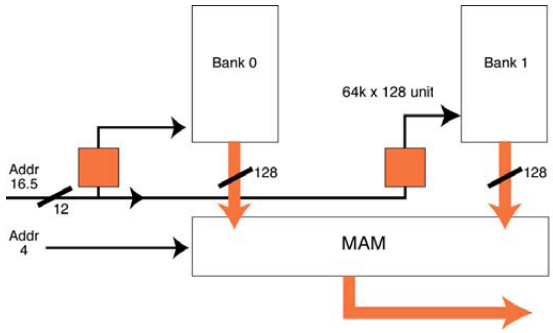


Fig. 4.44 *Modulo di accelerazione di memoria*

Prima di tutto la memoria *flash* è divisa in due banche dati da 128 *bit* (vedi fig. 4.44) e accessibili in modo indipendente: questo vuol dire che un solo accesso *flash* può caricare 4 istruzioni ARM, oppure 8 istruzioni *Thumb*.
Il codice dell'utente è suddiviso fra le due banche dati (*interleaving*), in modo tale che, mentre viene eseguito il codice corrispondente ai 128 *bit* prelevati dalla prima banca dati, i successivi 128 *bit* prelevati dalla seconda banca dati vengono "preparati" alla successiva esecuzione. Inoltre, in caso di piccoli *loop* o salti nel codice, il MAM è dotato di una serie di *buffer* che mantengono in memoria istruzioni caricate di recente, e che possono dunque essere rieseguite su richiesta.
Il MAM è configurato da due registri: temporizzazione e controllo. Il registro di temporizzazione è usato per la gestione del rapporto fra il *clock* della CPU e il tempo di accesso alla *flash*: i primi 3 *bit* del registro di temporizzazione specificano il numero di colpi di *clock* della CPU richiesti dal MAM per accedere alla *flash*. Poiché la frequenza di accesso alla *flash* è di 20 MHz e il *clock* della CPU ha una frequenza massima di 60 MHz, il massimo numero di cicli richiesti per accedere alla *flash* è 3: dunque il MAM permette di caricare 4 istruzioni ARM (o 8 istruzioni *Thumb*) in soli 3 cicli della CPU. Il registro di controllo è usato invece per definire la modalità operativa del MAM.
Al *reset*, il MAM è sempre disabilitato (vedi fig. 4.45), e l'accesso all'intero codice e ai dati avviene direttamente nella *flash*. E' però possibile abilitare il MAM parzialmente, ovvero il codice sequenziale è prelevato dal MAM, mentre le istruzioni di *branch* e i dati sono prelevati dalla *flash*. In caso di MAM totalmente abilitato, come già descritto in precedenza, è infine possibile accedere alla *flash* solamente attraverso il MAM.

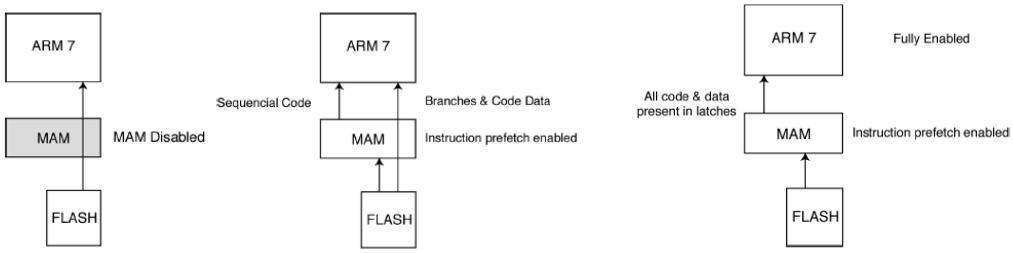


Fig. 4.45 *Disabilitazione e abilitazione (totale o parziale) del MAM*

Il motivo di questa possibilità di disabilitare (o abilitare solo parzialmente) il MAM sta nel fatto che, come nel caso di una memoria *cache*, il tempo di esecuzione del codice non è un valore deterministico (anche se molto più facilmente prevedibile che nel caso di una *cache*): per questo è conveniente disabilitare il MAM, o abilitarlo parzialmente, quando si ha l'assoluta necessità di prevedere con esattezza il tempo di esecuzione del codice.
Oltre ai due registri descritti in precedenza, il MAM è dotato di un gruppo di registri statistici che possono essere usati per verificare le prestazioni del MAM: sono essenzialmente basati su due contatori che registrano gli accessi diretti alla *flash* e quelli effettuati attraverso il MAM, fornendo dunque uno strumento valido per valutare l'efficacia del MAM nell'esecuzione di un codice.

4.4.3.5 Programmazione della memoria flash

Sebbene la memoria *flash* sia suddivisa in due banche dati, può essere gestita dall'utente come un'unica memoria: in termini di programmazione, più precisamente, essa appare come una serie di settori da 8 kB che possono essere programmati e cancellati in modo individuale. Ci sono diversi possibili modi per la programmazione della memoria *flash*: il più semplice è utilizzare il caricatore di *boot* integrato, che permette al codice di essere caricato nella RAM attraverso il dispositivo UART0, per poi essere programmato nella *flash*.
Il secondo modo per programmare la *flash* è utilizzare lo strumento di sviluppo JTAG, utile in fase di sviluppo perché la programmazione della *flash* può essere fatta direttamente dall'ambiente di *debug*, senza necessità di commutare fra il *debugger* e il caricatore di *boot*. Inoltre, la connessione JTAG può essere molto veloce, consentendo di scaricare fino a 400 KB/s: dunque per le grandi applicazioni, in particolare quelle che usano una *flash* esterna, è il miglior modo per la programmazione della *flash*.
Infine, è possibile riprogrammare le sezioni della memoria *flash* attraverso dei comandi impartiti direttamente dal programma applicativo scritto dall'utente: questa soluzione è meglio nota come *In-Application Programming* (IAP).

4.4.3.6 Circuito ad aggancio di fase

Il circuito ad aggancio di fase (PLL, *Phase Locked Loop*) raccoglie la frequenza di un oscillatore esterno al quarzo (nell'intervallo 10 ÷ 25 MHz) e la moltiplica fino ad ottenere un valore massimo di 60 MHz, ovvero la frequenza di *clock* utilizzata dalla CPU e dalle periferiche veloci (AHB). Il PLL consente dunque di far lavorare il μC alla massima velocità utilizzando un oscillatore esterno di frequenza più bassa,

minimizzando in tal modo le emissioni elettromagnetiche. La frequenza di uscita del PLL può essere anche fatta variare in modo dinamico, permettendo al dispositivo di rallentare la velocità di esecuzione nelle fasi di inattività (*idle*), consentendo di risparmiare potenza.

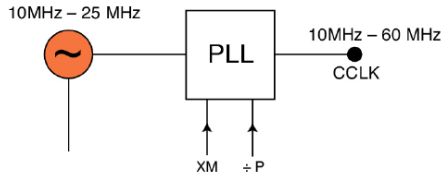


Fig. 4.46 *Controllo del circuito ad aggancio di fase*

Il PLL è controllato dalla due costanti M e P (fig. 4.46), che devono essere programmate per determinare la frequenza di uscita (CCLK). La costante M è un semplice fattore di moltiplicazione della frequenza del quarzo esterno. La frequenza di uscita sarà semplicemente data da:

```
Cclk = M x Osc
```

Nel circuito di reazione del PLL c'è un oscillatore controllato in corrente (CCO) che deve operare nel *range* 156 MHz ÷ 320 MHz: la seconda costante P agisce come un divisore programmabile per assicurare che la frequenza del CCO si mantenga sempre interna al *range* specificato. La frequenza operativa del CCO è definita in questo modo:

```
Fcco = Cclk x 2 x P
```

La frequenza dell'oscillatore al quarzo presente sulla scheda del µC LPC2103 è di 12 MHz, pertanto per ottenere i 60 MHz necessari per il *clock* della CPU dev'essere:

```
M = Cclk/Osc = 60/12 = 5
156< Fcco <320 = 60 x 2 x P
```

Si può osservare per ispezione che P = 2.
L'interfaccia di programmazione del PLL è mostrata in fig. 4.47:

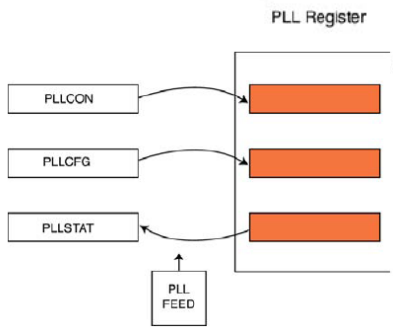


Fig. 4.47 *Interfaccia di programmazione del PLL*

Per inizializzare il PLL, occorre scrivere i valori di M e P nel registro PLLCFG; il registro PLLCON servirà invece per far partire il PLL e abilitarlo poi in uscita. E'importante notare che i valori di M e P sono trasferiti ai registri interni del PLL solo dopo che una sequenza iniziale (*feed*) viene scritta nel registro di *feed* del PLL: più precisamente, una volta aggiornati i registri PLLCON e PLLCFG, occorre scrivere 0x000000AA seguito da 0x00000055 (il registro PLLFEED).
Una volta partito il PLL, c'è sempre un intervallo di attesa prima che la frequenza di uscita sia stabile (ovvero avvenga l'aggancio) e utilizzabile come *clock*: questo transitorio può essere monitorato leggendo il *bit* LOCK presente nel registro PLLSTAT. Non appena il *bit* LOCK va a 1, il PLL può essere abilitato in uscita e lavorare come generatore di CCLK al posto oscillatore esterno: la sostituzione diviene effettiva ponendo a 1 il *bit* PLLC del registro PLLCON.
Se il *chip* entra nella modalità risparmio di potenza, il PLL si spegne e viene disconnesso: un eventuale ritorno alla modalità normale non ripristina il PLL, e dunque la procedura d'inizializzazione deve essere eseguita ogni volta che il *chip* esce dalla modalità risparmio di potenza.

4.4.3.7 Gestione del consumo di potenza

Il μC LPC2103 possiede delle funzionalità specifiche per la gestione del consumo di potenza. La CPU di ARM7 può infatti lavorare in due diverse modalità per il risparmio di potenza, controllate dai primi due *bit* del registro PCON:
1) *Idle mode*: la CPU è resa inattiva, mentre le periferiche sono comunque operative; in caso di *interrupt* proveniente da una qualsiasi periferica, la CPU si riattiva immediatamente.
2) *Power-down mode*: sia la CPU che le periferiche sono rese inattive. In questa modalità l'oscillatore è disattivato, e solo un *reset* o un *interrupt* esterno possono riattivare l'intero *chip*. Tutti gli *status* dei registri del processore e della RAM vengono mantenuti, così come i livelli logici dei *pin* I/O. Tornando alla modalità normale, il generatore di *clock* è l'oscillatore esterno, e il PLL dev'essere riconfigurato. E' possibile porre in *power-down* anche una sola periferica, se non utilizzata, attraverso il suo specifico *bit* di controllo di potenza nel registro PCONP. Ad ogni modo, alcune periferiche non possono essere mai disattivate, più precisamente: *watchdog*, GPIO, blocco di connessione *pin* e blocco di controllo del sistema.

4.4.3.8 Sistema di gestione degli interrupt IRQ e FIQ

In questa sezione verranno esaminate le due eccezioni più importanti per il μC LPC2103: l'*interrupt* generico IRQ e l'*interrupt* veloce FIQ. Queste due eccezioni vengono utilizzate per gestire tutti i generatori di *interrupt* esterni alla CPU di ARM7, che nel caso del μC LPC2103 sono le periferiche di servizio (*user*).
Per esaminare la struttura degli *interrupt* del μC LPC2103, è utile prendere come esempio il caso di un *interrupt* generato da un *pin* riservato a *interrupt* esterni (es. EINT1), che rappresenta la più semplice periferica da configurare.

4.4.3.8.1 Blocco di connessione pin

Tutti i *pin* I/O del μC LPC2103 sono connessi a un certo numero di funzioni interne attraverso un *multiplexer*, chiamato *blocco di selezione pin* (vedi fig. 4.48) che permette

all'utente di configurare un *pin* scegliendone la funzionalità, ad esempio GPIO (*General Purpose Input/Output*). Al *reset*, tutti i *pin* I/O sono configurati come GPIO: le funzionalità secondarie vengono selezionate attraverso il registro PINSEL. Ad esempio, per poter utilizzare il *pin* P0.14 come *pin* EINT1, occorre configurare opportunamente il registro PINSEL, permettendo il passaggio dalla funzionalità GPIO alla funzionalità EINT1.

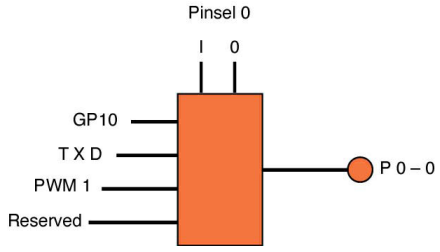


Fig. 4.48 *Blocco di selezione pin*

4.4.3.8.2 Pin riservati ad interrupt esterni

Gli *interrupt* esterni sono gestiti da tre registri. Il registro EXMODE verifica se l'*interrupt* è attivato sul livello o sul fronte: in quest'ultimo caso, il registro EXPOL è utilizzato per notificare se l'attivazione dell'*interrupt* avviene sul fronte di salita o su quello di discesa (in caso di attivazione sul livello, gli *interrupt* esterni sono sempre attivi a livello logico basso). Se si è in modalità *power-down*, il registro EXWAKE può abilitare un *interrupt* a riattivare la CPU.

4.4.3.8.3 Controllore per interrupt vettorizzati (VIC)

Come detto in precedenza, la CPU di ARM7 ha due linee dedicate agli *interrupt* esterni: IRQ e FIQ. Generalmente, in un sistema semplice, ci dovrebbe essere un solo generatore di *interrupt* veloci FIQ, in modo tale che il processore possa soddisfare la richiesta FIQ nel più breve tempo possibile: questo significa che tutti gli altri generatori di *interrupt* devono essere connessi alla linea IRQ. Si potrebbe utilizzare una porta OR, connessa alla linea IRQ, con tanti ingressi quanti sono i generatori di *interrupt*: questo però significa che, ad ogni richiesta IRQ, la CPU dovrebbe interrogare ogni periferica per scoprire chi ha generato l'*interrupt*, e questo richiederebbe molti cicli. Per questo motivo il µC LPC2103 integra un VIC (*Vector Interrupt Controller*) per una più efficiente gestione degli *interrupt* esterni [59] (vedi fig. 4.49).
Ognuno dei generatori di *interrupt* integrati su *chip* è connesso al VIC attraverso un canale prefissato. L'applicazione *software* può connettere ognuno di questi canali alle linee di *interrupt* in tre modi possibili: FIQ, IRQ vettorizzato e IRQ non vettorizzato. Il tempo di risposta all'*interrupt* varia per ognuna delle richieste: FIQ è ovviamente la più veloce, seguita dalla richiesta IRQ vettorizzata, mentre la richiesta IRQ non vettorizzata è la più lenta.
Nei prossimi paragrafi verrà presentato ognuno di questi metodi di gestione degli *interrupt* esterni.

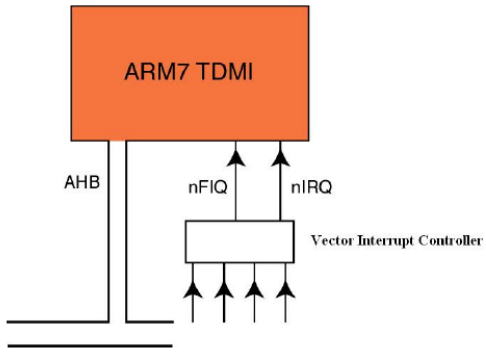


Fig. 4.49 *Impiego del VIC per la gestione di interrupt esterni*

4.4.3.8.4 Interrupt FIQ

Ogni generatore di *interrupt* può essere connesso alla linea FIQ: il registro di selezione *interrupt* (ISR, *Interrupt Select Register*) del VIC ha un *bit* riservato per ognuno di essi, e il settaggio di questo *bit* connette automaticamente il canale selezionato alla linea FIQ. E' possibile, come avviene nei sistemi più complessi, settare più *bit* dell'ISR, abilitando in questo modo più di un generatore di *interrupt* FIQ: in questo caso, per determinare quale sorgente ha generato l'*interrupt*, si devono esaminare il registro di *status* FIQ del VIC e il codice eseguito. Ovviamente, avere più generatori di *interrupt* FIQ rallenta l'ingresso nel codice ISR.

Una volta selezionata la sorgente FIQ, l'*interrupt* può essere abilitato nel registro di abilitazione *interrupt* (IER, *Interrupt Enable Register*) del VIC. Naturalmente, come il VIC, anche la periferica che ha generato l'*interrupt* dev'essere configurata, e devono essere abilitati i suoi registri interni di *interrupt*.

Dopo che è stata inoltrata una richiesta FIQ, il processore comincerà a lavorare in modalità FIQ, e nel registro PC verrà memorizzato il vettore associato all'eccezione FIQ, ovvero 0x0000001C: per poter eseguire la *subroutine* ISR dell'*interrupt*, nel codice utente occorre sempre mettere un salto a questa locazione. Prima di uscire dalla *subroutine* ISR, inoltre, occorre sempre verificare di aver resettato tutte le *flags* di *status* dell'*interrupt* nel PIR (*Peripheral Interrupt Register*) associato alla periferica che ha fatto la richiesta FIQ, altrimenti si avranno *interrupt* FIQ in continuazione.

4.4.3.8.5 Interrupt IRQ vettorizzati

Nel caso di un solo generatore di *interrupt* FIQ, tutte le altre periferiche in grado di generare *interrupt* devono essere connesse alla linea IRQ. Per assicurare un'efficiente e tempestiva esecuzione delle richieste IRQ, il VIC è dotato di una tabella *hardware* programmabile che riporta l'indirizzo della funzione da eseguire (codice ISR) per un dato generatore di *interrupt*.

Il VIC contiene 16 *slot* disponibili per l'indirizzamento vettorizzato (vedi fig. 4.50), e ognuno di questi *slot* contiene un registro di controllo (VCR, *Vector Control Register*) e un registro di indirizzo (VAR, *Vector Andress Register*) . Il VCR è costituito da un gruppo di *bit* di canale e da un *bit* di *enable*: programmando i *bit* di canale, una qualsiasi delle periferiche IRQ può essere connessa ad uno qualsiasi degli *slot*, e infine abilitata

utilizzando il *bit* di *enable*. La priorità di un *interrupt* vettorizzato è indicata nel numero dello *slot*: più basso è e più importante è l'*interrupt*.

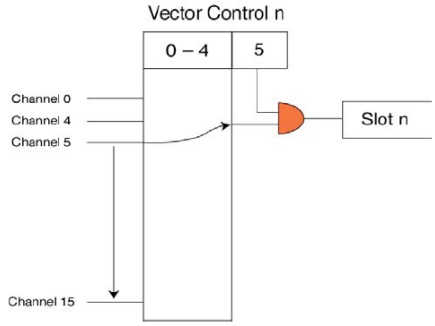


Fig. 4.50 *Struttura interna del VIC*

Il VAR è un registro in cui viene memorizzato l'indirizzo della *subroutine* ISR da eseguire quando l'*interrupt* associato a un certo *slot* del VIC viene abilitato: in pratica, quando viene generato un *interrupt* vettorizzato, il canale corrispondente viene deviato su uno *slot* specifico, e l'indirizzo della *subroutine* ISR da eseguire (memorizzato nello *slot*) viene caricato in una locazione di memoria fissa, il VAR. Mentre sta accadendo questo nell'unità VIC, la CPU completa il passaggio alla modalità IRQ, e nel registro PC viene memorizzato il vettore corrispondente all'IRQ (0x00000018) al fine di poter eseguire la specifica *subroutine* ISR (il cui indirizzo è contenuto nel VAR).
Come per gli *interrupt* FIQ, anche nel caso IRQ occorre assicurarsi, prima di uscire dalla *subroutine* ISR, che siano resettate le *flags* di *status* dell'*interrupt* nel PIR della periferica che ha fatto la richiesta IRQ. Inoltre, alla fine dell'*interrupt* occorre scrivere necessariamente un indirizzo fittizio (*dummy*) nel registro VAR: questo segnala al VIC la fine dell'*interrupt*, e ogni altra richiesta IRQ in sospeso potrà essere accettata.

4.4.3.8.6 Interrupt IRQ non vettorizzati

Il VIC è in grado di gestire 16 periferiche come IRQ vettorizzati, e almeno un'altra periferica come FIQ. Dunque, nel caso in cui ci siano più di 17 generatori di *interrupt* sul *chip*, ogni periferica addizionale viene gestita come *interrupt* IRQ non vettorizzato. I generatori di *interrupt* non vettorizzati sono serviti da un'unica *subroutine* ISR, il cui indirizzo è memorizzato in un registro VAR addizionale detto DVAR (*Default Vector Address Register*). Quando viene accettata la richiesta di un *interrupt* non vettorizzato, l'indirizzo specificato nel DVAR viene caricato nel VAR, causando il salto del processore alla *subroutine* ISR predefinita. Ovviamente, la CPU deve leggere preventivamente il registro di status IRQ per capire quale delle periferiche associate a un *interrupt* non vettorizzato ha generato l'eccezione.
Come nel caso di IRQ vettorizzato, anche in questo caso occorre resettare il registro PIR prima di uscire dalla *subroutine* ISR, e inoltre specificare un indirizzo *dummy* nel VAR.

4.4.3.8.7 Simulazione di un interrupt

Con il VIC è possibile generare un *interrupt* su un canale qualsiasi direttamente attraverso il *software* dell'applicazione, grazie ai registri VIC SWI (*Software Interrupt*): questi registri non hanno nulla a che vedere con le istruzioni SWI, ma servono solo a testare i generatori di *interrupt*, ad esempio durante le operazioni di *test* o di simulazione in fase di progetto.

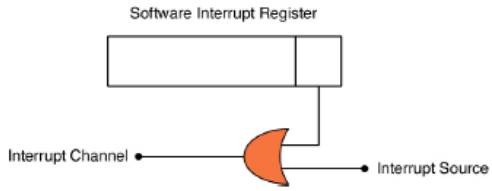


Fig. 4.51 *Registro SWI per la simulazione di un interrupt*

Inoltre il VIC ha una modalità protetta che impedisce di accedere a qualunque suo registro quando si è in modalità *user*. Se il codice dell'applicazione deve accedere al VIC, occorre entrare in una modalità privilegiata (FIQ, IRQ o SWI).

4.4.3.8.8 Interrupt annidati

La struttura degli *interrupt* interna alla CPU di ARM7 e al VIC non supporta *interrupt* annidati. Se l'applicazione richiede la possibilità di interrompere con un *interrupt* l'esecuzione di una *subroutine* ISR, occorre provvedere via *software* attraverso una coppia di macroistruzioni: IENABLE e IDISABLE.

Una volta che il processore è entrato nella *subroutine* ISR in modalità IRQ, occorre eseguire, per abilitare un *interrupt* annidato, l'istruzione IENABLE, che prevede una serie di sottoistruzioni: 1) il registro SPSR_irq dev'essere copiato in LR e salvato, per permettere di ripristinare correttamente il registro CPSR al ritorno nella modalità *user*; 2) viene abilitata nuovamente la linea IRQ per permettere ulteriori *interrupt* e si passa in modalità *system* (uguale alla modalità *user*, ma con la possibilità di modificare i registri CPSR e SPSR attraverso le istruzioni MRS e MSR); 3) una volta entrati in modalità *system*, il nuovo registro LR dev'essere preservato ponendolo sullo *stack*, perché potrebbe contenere dei valori utilizzati anche dal codice in modalità *user*.

Una volta completata la macroistruzione IENABLE, viene eseguita la *subroutine* ISR dell'*interrupt* annidato, dopo di che si esegue la macroistruzione IDISABLE per invertire il processo: si ripristina lo stato del registro LR, si disabilita la linea IRQ, si ritorna alla modalità IRQ, e infine si procede al ripristino del registro SPSR_irq. Questo conclude la sequenza di *interrupt* annidati.

Qui di seguito sono riportate nel dettaglio le due macroistruzioni IENABLE e IDISABLE:

```
#define IENABLE /* Nested Interrupts Entry */
__asm { MRS LR, SPSR } /* Copy SPSR_irq to LR */
__asm { STMFD SP!, {LR} } /* Save SPSR_irq */
__asm { MSR CPSR_c, #0x1F } /* Enable IRQ (Sys Mode) */
__asm { STMFD SP!, {LR} } /* Save LR */

#define IDISABLE /* Nested Interrupts Exit */
```

```
__asm { LDMFD SP!, {LR} } /* Restore LR */
__asm { MSR CPSR_c, #0x92 } /* Disable IRQ (IRQ Mode) */
__asm { LDMFD SP!, {LR} } /* Restore SPSR_irq to LR */
__asm { MSR SPSR_cxsf, LR } /* Copy LR to SPSR_irq */
```

Si tratta, come si può notare, di un codice composto complessivamente da 8 istruzioni (32 *bytes* di codice ARM), e l'esecuzione di entrambe le macroistruzioni richiede un totale di 230 ns.
Questo schema permette ad un *interrupt* di interrompere un qualsiasi altro *interrupt*: nel caso in cui ci fosse bisogno di dare una priorità agli *interrupt* annidati, le macroistruzioni possono bloccare gli *interrupt* a priorità bassa disabilitando le rispettive periferiche attraverso il VIC.

4.4.4 Periferiche di servizio (*user*)

In questa sezione verranno brevemente descritte le periferiche di servizio (*user*) del μC LPC2103, che comprendono fra le altre: *pin* I/O *general purpose* (GPIO), *timer general purpose*, modulatore PWM, *Real Time Clock* (RTC), sistema *watchdog*, UART, interfacce I^2C e SPI, convertitore analogico/digitale.

4.4.4.1 Pin I/O general purpose (GPIO)

Al *reset*, il blocco di connessione *pin* configura tutti i *pin* periferici come *pin* d'ingresso *general purpose* I/O (GPIO). I *pin* GPIO sono controllati da quattro registri, come mostrato in fig. 4.52:

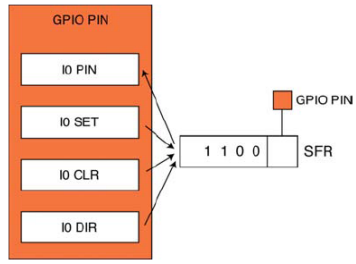


Fig. 4.52 *Registri di controllo dei pin GPIO*

Il registro IODIR permette ad ogni *pin* di essere configurato individualmente come *pin* d'ingresso (0) o *pin* di uscita (1). Se il *pin* è stato configurato come *pin* di uscita, i registri IOSET e IOCLR permettono di gestire lo stato del *pin*: più precisamente, scrivendo un 1 all'interno di questi registri si setta (IOSET) o si resetta (IOCLR) il corrispondente *pin*. Infine, lo stato del *pin* GPIO può essere verificato ad ogni istante andando a leggere il contenuto del registro IOPIN.

4.4.4.2 Timer general purpose

Il μC LPC2103 è dotato di quattro *timer general purpose*, strutturalmente identici ma leggermente diversi in termini di quantità di funzioni supportate. I *timer* sono basati su

due contatori (a 16 o 32 *bit*): un *timer counter* e un *prescaler*. Il *clock* comune a tutti i *timer* è PCLK, il *clock* utilizzato dalle periferiche integrate.
La frequenza di conteggio del *timer* è controllata dal valore memorizzato nel registro del *prescaler*, più precisamente: 1) il *prescaler* viene incrementato di 1 ad ogni colpo di PCLK finché non raggiunge il valore memorizzato nel suo registro; 2) viene incrementato di 1 il *timer counter*; 3) il *prescaler* viene resettato e ricomincia il suo conteggio. Il registro *Timer Control* contiene solo due *bit*, utilizzati per abilitare (o disabilitare) il *timer* e resettare il suo conteggio.

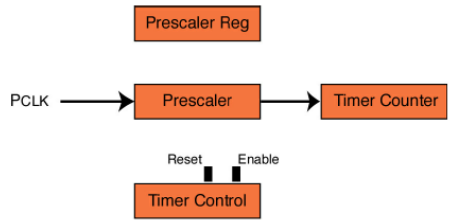


Fig. 4.53 *Struttura di un timer general purpose*

Ogni *timer* è dotato di un certo numero di canali di cattura (vedi fig. 4.54), che permettono all'utente di registrare il valore attuale del *timer counter*. Ogni canale di cattura è infatti associato ad un *pin* di cattura (CAP) che può essere abilitato dal blocco di connessione *pin*. Un registro di controllo (*capture control register*) può essere configurato per abilitare la cattura sul fronte di salita o di discesa (o entrambi) del segnale presente sul *pin* di cattura: quando si verifica un evento di questo tipo, il valore attuale del *timer counter* viene trasferito nel registro di cattura (*capture register*), e può essere generato un *interrupt*.

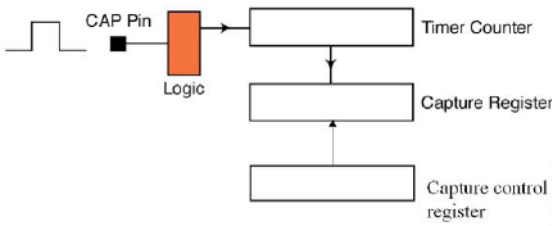


Fig. 4.54 *Canale di cattura di un timer*

Ogni *timer* è dotato anche di un certo numero di canali di comparazione (*match channel*), ad ognuno dei quali è associato un registro (*match register*) che può memorizzare un numero a 32 *bit*. Il valore attuale del *timer counter* può essere confrontato col numero memorizzato nel *match register*, e se i due valori sono uguali viene generato un evento, che può essere un'azione da compiere sul *timer* (*reset*, *stop* o generazione di un *timer interrupt*) oppure su un *pin* esterno (*set*, *reset* o *toggle*).

4.4.4.3 Modulatore PWM

Il modulatore PWM (*Pulse Width Modulation*) è capace di produrre 6 canali di modulazione PWM "*single edge*" e 3 canali di modulazione PWM "*dual edge*".
E' sostanzialmente un *timer general purpose* con delle funzionalità *hardware* aggiuntive. In un *timer* semplice, quando un nuovo valore viene scritto in un *match register* diviene subito effettivo (a meno che questo non venga impedito via *software*): all'interno di un ciclo PWM, in fase di aggiornamento dei canali di modulazione, l'attivazione in istanti diversi dei nuovi valori PWM impostati nei *match register* può portare a risultati imprevedibili. Per evitare questo, il modulatore PWM integra un meccanismo (*shadow latch*) che permette comunque di impostare i nuovi valori PWM, ma di renderli effettivi simultaneamente solo all'inizio di un nuovo ciclo.

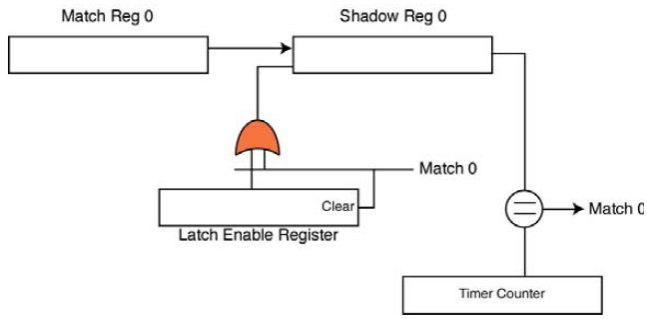


Fig. 4.55 *Shadow latch per un modulatore PWM*

Come si vede in fig. 4.55, il valore di un dato *match register* può essere aggiornato in ogni istante, ma non diventa effettivo finché il *bit* corrispondente al canale di comparazione non viene settato all'interno del *Latch Enable Register* (LER): una volta settato il LER, il valore del *match register* verrà trasferito allo *shadow register* all'inizio di un nuovo ciclo.
La seconda differenza *hardware* del modulatore PWM rispetto ad un semplice *timer* sta nel controllo dei *pin* di uscita dei canali di comparazione: invece di una connessione diretta con le linee *match*, ci sono una serie di *flip-flop* SR e *multiplexer*, per permettere al modulatore di produrre canali di modulazione PWM sia "*single edge*" che "*dual edge*". I *multiplexer* sono gestiti dal registro PWMSEL per configurare l'uscita del canale di modulazione in una delle due modalità.

4.4.4.4 Real Time Clock (RTC)

Il *Real Time Clock* (RTC) del μC LPC2103 è un orologio-calendario che integra un *set* di registri di allarme, impostabili su una particolare data e ora (fino al 2099) oppure su un dato valore di conteggio specificato in un altro registro: in entrambi i casi, un esito positivo della comparazione dà luogo alla generazione di un *interrupt*. C'è anche la possibilità di impostare la generazione di un *interrupt* ad ogni incremento di un certo registro temporale (ad esempio ogni secondo, ogni minuto, ogni ora, etc.).
L'RTC lavora ad una frequenza di *clock* standard 32.768 kHz, che viene ottenuta dalla frequenza di PCLK (il *clock* periferico) attraverso un *prescaler* programmabile attraverso due registri: PREINT per i numeri interi e PREFRAC per le cifre decimali.

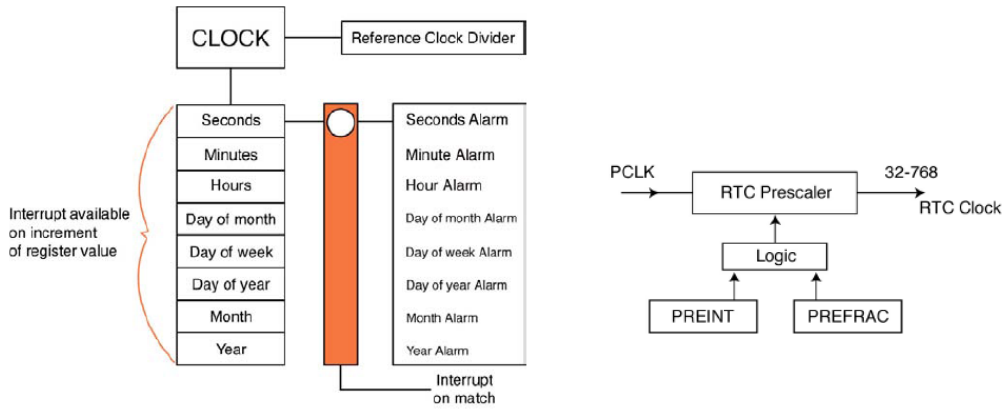


Fig. 4.56 *Set di registri dell'RTC (a) e generazione del clock RTC (b)*

Le equazioni utilizzate per calcolare i valori da impostare nei due registri PREINT e PREFRAC sono:

```
PREINT = (Ina)(pclk/32768)-1
PREFRAC = pclk – ((PREINT+1) x 32768)
```

Nel caso in cui PCLK = 30 MHz, si ottiene PREINT = 914 e PREFRAC = 17280: questi valori possono essere programmati direttamente nei registri del *prescaler*. E'sufficiente poi abilitare il *clock* nel CCR (*Clock Control Register*) per far partire il conteggio:

```
PREINT = 0x00000392; //Set RTC prescaler for 30.000 MHz Pclk
PREFRAC = 0x00004380;
CCR = 0x00000001;    //Start the RTC
```

4.4.4.5 Watchdog

Come molti altri microcontrollori, anche l'LPC2103 è dotato di un sistema *watchdog* per recuperare il controllo di un programma andato in blocco, il che può avvenire attraverso un *reset* del processore oppure generando un *interrupt*. Il sistema *watchdog* è costituito da quattro registri, come si può vedere in fig. 4.57:
Il periodo di *timeout* del sistema *watchdog* (ovvero il periodo alla fine del quale, se non viene re-inizializzato dal processore, genererà un *reset* o un *interrupt*) è settato da un valore programmato nel registro WDTCR (*Watchdog Constant Register*), ed è determinato dalla formula:

```
Wdperiod = Pclk x WDTC x 4
```

Il valore minimo per la costante WDTC è 256 (che corrisponde a un periodo di 17.066 μs a 60 MHz), il valore massimo è 2^{32} (appena sotto i 5 minuti a 60 MHz).

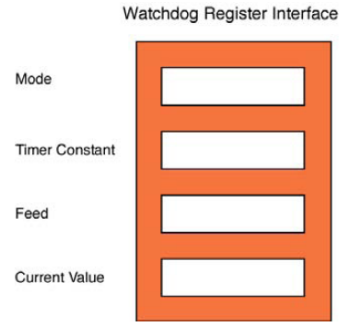


Fig. 4.57 *Registri del sistema watchdog*

Una volta programmata la costante WDTC, si può configurare la modalità operativa nel *watchdog mode register* (WDTMR), che contiene 3 *bit* di controllo: il primo nel caso di generazione di un *interrupt*, il secondo nel caso di generazione di un *reset*, mentre il terzo è un *bit* di abilitazione del *watchdog*.
Il *mode register* contiene anche due *flags*: 1) WDTOF è settata alla fine del periodo di *timeout* e viene rimessa a 0 solo dopo un *reset* esterno di tipo *hardware*, consentendo al codice iniziale di capire se il *reset* è stato generato esternamente (*power-on reset*) oppure da un errore di programma; 2) *Watchdog Interrupt* è di sola lettura, e dev'essere letta con lo scopo di resettare l'*interrupt* generato dal *watchdog*.
Dopo la configurazione dei registri WDTCR e WDTMR, il *watchdog* viene attivato scrivendo nel registro *feed* la sequenza 0xAA, seguita da 0x55: se tale sequenza non viene correttamente scritta, si considera questo un errore di *feed* e viene generato un evento di *timeout* (*interrupt* o *reset*). E'importante notare che, sebbene il *watchdog* possa essere abilitato con il registro WDTMR, sarà operativo soltanto quando riceverà la prima corretta sequenza di *feed*: una volta operativo, il *watchdog* dovrà poi regolarmente ricevere dal processore sequenze di *feed* per poter essere re-inizializzato e ricominciare a contare.
Infine, l'ultimo registro è WDTVR (*Watchdog Timer Value Register*), che permette di leggere il valore attuale del contatore del sistema *watchdog*.

4.4.4.6 UART

Il µC LPC2103 integra due dispositivi UART strutturalmente identici: UART0 e UART1, con l'unica differenza che UART1 supporta anche un *modem*. Entrambe le periferiche sono conformi allo standard industriale 550, hanno un generatore interno di *baud rate* e registri FIFO da 16 *bytes* in trasmissione e ricezione. L'inizializzazione della periferica UART0 è mostrata qui di seguito:

```
void init_serial (void) /* Initialize Serial Interface */
{
       PINSEL0 = 0x00050000; /* Enable RxD0 and TxD0 */
       U0LCR = 0x00000083; /* 8 bits, no Parity, 1 Stop bit */
       U0DLL = 0x000000C2; /* 9600 Baud Rate @ 30MHz APB Clock */
       U0LCR = 0x00000003; /* DLAB = 0 */
}
```

Innanzitutto, occorre programmare il blocco connessione *pin* per far passare i *pin* dalla funzionalità generica GPIO alla funzionalità UART (RXD0 e TXD0). Dopo di che, il registro di controllo della linea UART (U0LCR) viene usato per configurare il formato del carattere da trasmettere: nell'esempio mostrato, il formato del carattere è impostato a 8 *bit*, nessun *bit* di parità e 1 *bit* di stop. Nel registro U0LCR c'è un *bit* addizionale (DLAB, *Divisor Latch Access Bit*), come mostrato in fig. 4.58, che dev'essere settato per poter programmare il generatore di *baud rate*.

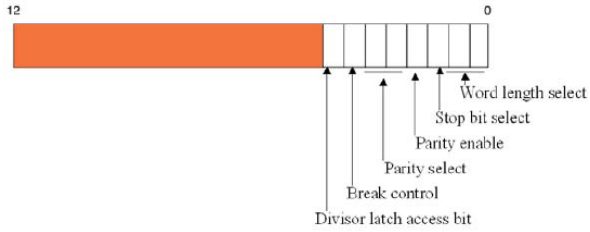


Fig. 4.58 *Registro di controllo della linea UART*

Il generatore di *baud rate* è un *prescaler* a 16 *bit* che divide la frequenza del *clock* periferico PCLK per generare il *clock* UART, che deve avere una frequenza pari a 16 volte il *baud rate*. Dunque la formula utilizzata per il calcolo del *baud rate* UART è:

```
Divisor = Pclk/16 x BAUD
```

Nel caso in cui PCLK = 30 MHz, per ottenere un *baud rate* di 9665, si deve utilizzare la formula:

```
Divisor = 30,000,000/16 x 9600 = (approx) 194 or 0xC2
```

Il valore del divisore è mantenuto in due registri: *Divisor Latch MSB* (DLM) e *Divisor Latch LSB* (DLL). I primi 8 *bit* di entrambi i registri contengono le due metà del divisore a 16 *bit*, come mostrato in fig. 4.59:

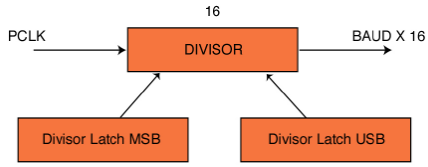


Fig. 4.59 *Registri DLM e DLL*

L'ultima istruzione di inizializzazione prevede il ritorno a 0 del *bit* DLAB, per proteggere il contenuto dei registri del divisore.
Una volta inizializzata la periferica UART, si possono cominciare a trasmettere caratteri scrivendoli nel THR (*Transmit Holding Register*) o a ricevere caratteri leggendoli dall'RBR (*Receive Buffer Register*). In effetti, entrambi i registri occupano la stessa locazione di memoria: la scrittura trasferisce il carattere nel FIFO in trasmissione,

mentre la lettura dalla suddetta locazione di memoria carica il carattere nel registro FIFO in ricezione. Qui di seguito sono riportate le *routine* di scrittura e lettura di caratteri:

```
int putchar (int ch) /* Write character to Serial Port */
{
   if (ch == '\n')
       {
         while (!(U0LSR & 0x20));
         U0THR = CR; /* output CR */
       }
   while (!(U0LSR & 0x20));
   return (U0THR = ch);
}

int getchar (void) /* Read character from Serial Port */
{
   while (!(U0LSR & 0x01));
   return (U0RBR);
}
```

Le funzioni *putchar*() e *getchar*() sono usate per la scrittura/lettura di un carattere nella UART, ed inoltre sono in grado di leggere il *Link Status Register* (LSR) per controllare la presenza di errori UART (*framing*, *parity*, etc.) e lo stato dei registri FIFO in trasmissione e ricezione (fig. 4.60).

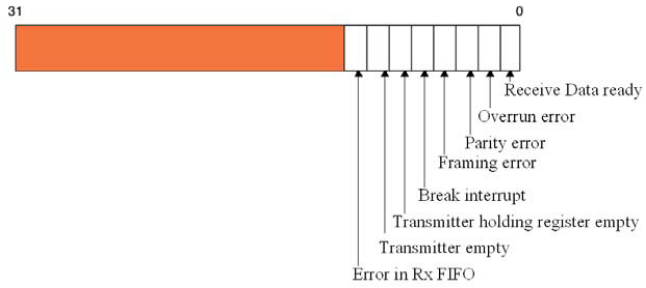


Fig. 4.60 *Struttura del registro LSR*

L'UART ha un singolo canale di *interrupt* connesso al VIC, ma ben tre generatori di *interrupt*. Un primo *interrupt* UART può essere generato in seguito a un cambiamento nello *status* della linea di ricezione: in questo caso, se si verifica una condizione di errore, viene generato un *interrupt* e il registro LSR può essere letto per scoprire qual è stata la causa dell'errore.
Gli altri due tipi di *interrupt* UART sono gli *interrupt* di ricezione e di trasmissione. L'*interrupt* di ricezione è innescato da caratteri che si stanno ricevendo nel registro RX FIFO, e il punto di generazione dell'*interrupt* può essere configurato nel registro di controllo UART FIFO. L'*interrupt* di ricezione può essere programmato per essere generato dopo la ricezione di 1, 4, 8 o 14 caratteri: ad esempio, se si programma un *interrupt* dopo la ricezione di 8 caratteri nel *buffer*, nel caso di un totale di 34 caratteri inviati verranno generati 4 *interrupt* di ricezione, con 2 caratteri lasciati indietro nel

FIFO. La presenza di caratteri residui nel FIFO determina un *interrupt* CTI (*Character Timeout Indication*), che consentirà di terminarne la ricezione.
Infine, anche il registro FIFO di trasmissione può generare *interrupt* quando il registro THR (*Transmit Holding Register*), oppure il registro TSR (*Transmit Shift Register*), sono vuoti.

4.4.4.7 Interfaccia I²C

I²C (*Inter-Integrated Circuit*) è un sistema di comunicazione seriale sincrona tra circuiti integrati, ed ha dunque il vantaggio di prevedere due sole linee: SDA (*Serial Data Line*) e SCL (*Serial Clock Line*). Il classico sistema I²C è composto da almeno un *master* ed uno *slave*, ma possono essere usate architetture *multimaster* e *multislave* in sistemi più complessi.
Nel caso del μC LPC2103, ad esempio, l'interfaccia I²C può operare sia in modalità *master* che in modalità *slave* (a 400 k*bit*/s), e in modalità *master* farà automaticamente da arbitro in un sistema *multimaster*. Un tipico sistema I²C è mostrato in fig. 4.61, con il μC LPC2100 (della stessa famiglia dell'LPC2103) connesso a due dispositivi esterni: da notare le resistenze di *pull-up* su entrambe le linee SDA e SCL.

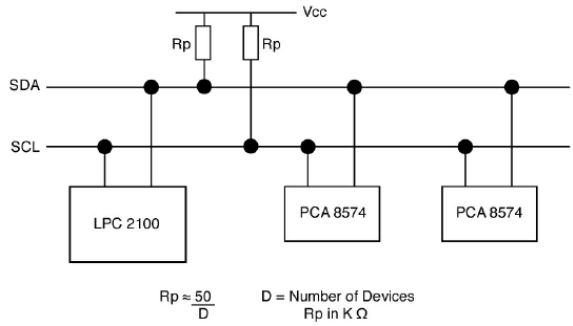


Fig. 4.61 *Sistema d'interfacciamento I²C*

L'interfaccia I²C è composta da 7 registri. Ci sono 2 registri separati (I2CONSET e I2CONCLR) utilizzati per settare e resettare *bit* nel registro di controllo; altri 2 registri (I2SCLH e I2SCLL) servono per determinare il *bit rate*; il registro di stato (I2STAT) restituisce codici di controllo per mettere in relazione i diversi eventi sul *bus*; il registro dati (I2DAT) è usato per raccogliere i dati in ricezione o per preparare i dati da trasmettere; infine, nel caso in cui il μC LPC2103 sia configurato in modalità *slave*, il suo indirizzo di rete viene settato programmando il registro I2ADR.
Il codice di inizializzazione dell'interfaccia I²C è mostrato qui di seguito:

```
VICVectCntl1 = 0x00000029;       //select a priority slot for a
                                   given interrupt
VICVectAddr1 = (unsigned)I2CISR; //pass the address of the IRQ
                                   into the VIC slot
VICIntEnable = 0x00000200;       //enable interrupt

PINSEL0 = 0x50; //Switch GPIO to I2C pins
I2SCLH = 0x08;  //Set bit rate to 57.6KHz
```

```
I2SCLL = 0x08;
```

L'interfaccia I^2C dev'essere programmata per rispondere ad ogni evento che avviene sul *bus*, e dunque è una vera e propria periferica governata dagli *interrupt*: per questo motivo, la prima cosa da fare in fase di inizializzazione è configurare il VIC per rispondere ad un *interrupt* I^2C; poi si configura il blocco di connessione *pin* per connettere le linee SDA e SCL ai *pin* esterni; infine, si setta il *bit rate* programmando i registri I2SCLH e I2SCLL.
Una volta configurato, il μC LPC2103 può iniziare la comunicazione con gli altri dispositivi connessi al *bus*, per leggere o scrivere dati (se *master*), oppure (se *slave*) ricevere e rispondere a richieste del *master*.

4.4.4.8 Interfaccia SPI

L'interfaccia SPI (*Serial Peripheral Interface*) è molto simile all'interfaccia I^2C: è anch'essa un sistema di comunicazione seriale sincrona tra circuiti integrati del tipo *master/slave*, ma è meno "intelligente", nel senso che la gestione della comunicazione sul *bus* è lasciata al codice di programmazione.
La periferica SPI ha 4 *pin* esterni: un *pin* per il *clock* seriale, un *pin* per la selezione dello *slave* e due *pin* per i dati (*master in /slave out* oppure *master out/slave in*). Il *pin* di *clock* seriale fornisce un *clock* a 400 k*bit*/s in modalità *master*, oppure accetta un *clock* esterno in modalità *slave*. Il *bus* SPI è puramente una connessione per il trasferimento seriale di dati ad alta velocità, e a differenza dell'interfaccia I^2C non ha alcuno schema d'indirizzamento: per questo c'è un *pin* separato che permette di selezionare una periferica esterna come *slave*. Infine, i due *pin* di dati MISO e MOSI sono connessi alla periferica esterna SPI, e il loro orientamento è ovviamente diverso a seconda che il μC LPC2103 sia *master* o *slave*.

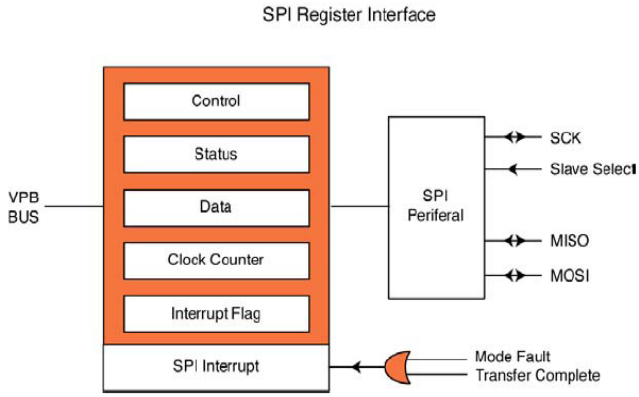


Fig. 4.62 *Interfaccia SPI*

L'interfaccia SPI è composta (vedi fig. 4.62) da 5 registri. Il registro *Clock Counter* determina il *baud rate* (PCLK è semplicemente diviso per il valore presente in questo registro). Il registro di controllo è utilizzato per configurare la modalità operativa del *bus* SPI: si può ad esempio scegliere la polarità del *clock* (se attivo basso o attivo alto), oppure la fase del *clock* rispetto al *bus* dati, o ancora l'orientamento dei dati (se

trasmettere per primo l'LSB o il MSB), e ognuna di queste configurazioni ha il suo *bit* specifico nel registro di controllo. Una volta programmato il *bit rate* e configurato il registro di controllo, può iniziare la comunicazione.

4.4.4.9 Convertitore analogico/digitale (ADC)

L'ADC integrato nel μC LPC2103 è un convertitore ad approssimazioni successive (SAR) a 10 *bit*, con tempo di conversione di 2.44 μs (410 kSPS) e 8 ingressi multiplexati. L'interfaccia di programmazione è mostrata in fig. 4.63:

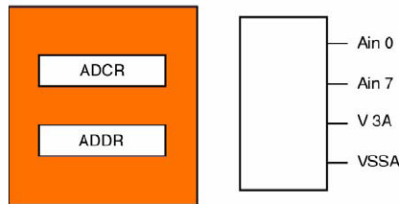


Fig. 4.63 *Interfaccia di programmazione dell'ADC interno al μC LPC2103*

Il registro di controllo A/D (ADCR), la cui struttura è riportata in fig. 4.64, stabilisce la configurazione del convertitore e controlla l'inizio della conversione.

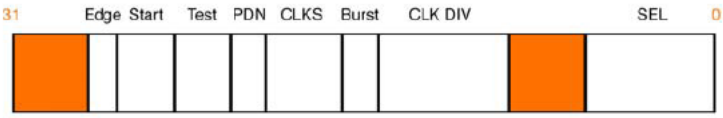


Fig. 4.64 *Struttura del registro di controllo dell'ADC*

Il primo passo della configurazione dell'ADC consiste nel settare il *clock*: come per le altre periferiche, anche il *clock* dell'ADC (`Adclk`) deriva da PCLK, la cui frequenza dev'essere suddivisa fino ad ottenere 4.5 MHz. Più precisamente, PCLK viene diviso per il valore memorizzato in CLKDIV più uno:

```
CLKDIV = ( PCLK/Adclk) – 1
```

L'ADC è in grado di lavorare in modalità *power-down*: questo riduce il consumo di potenza e il rumore generato dall'ADC all'interno del *chip*. Al *reset*, l'ADC è in modalità *power-down*, e per abilitarlo occorre settare il *bit* PDN nel registro ADCR.
Diversamente da altre periferiche, l'ADC può effettuare misure sui *pin* esterni configurati come GPIO: ad ogni modo, per aumentare l'accuratezza della conversione, è possibile agire sul blocco connessione *pin* per cambiare la funzionalità dei *pin* esterni, e farli passare da GPIO a *pin* dedicati espressamente all'ADC.
Prima di una conversione, la risoluzione dev'essere definita programmando il campo CLKS: la massima risoluzione è 10 *bit*, ma è possibile diminuirla fino a 3 *bit*. Una volta definita la risoluzione, si può passare a configurare le conversione.
L'ADC presenta due modalità di conversione: *hardware* e *software*. La modalità *hardware* consente di selezionare un certo numero di canali d'ingresso, e ognuno di essi

viene convertito separatamente, uno alla volta, finché l'ADC non viene stoppato. Alla fine di ogni conversione, il risultato è disponibile nel registro dati dell'ADC (ADDR), la cui struttura è presentata in fig. 4.65:

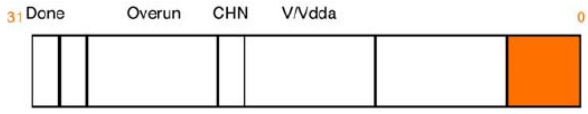


Fig. 4.65 *Struttura del registro dati dell'ADC*

Alla fine di una conversione, il *bit Done* viene settato, ed è possibile generare un *interrupt*. Il risultato della conversione è memorizzato nel campo V/V_{DDA} come rapporto fra la tensione sul canale analogico e la tensione sul *pin* di alimentazione analogica. Nel campo CHN è immagazzinato il numero di canali per i quali è stata effettuata la conversione. Infine, se il risultato di una conversione non viene letto prima dell'arrivo del risultato della conversione successiva, verrà sovrascritto da quest'ultimo, e verrà settato il *bit Overun*.
Nella modalità di conversione *software*, viene selezionato un solo canale per la conversione usando i *bit* SEL del registro di controllo ADCR, e la conversione inizia scrivendo tramite *software* il valore 0x01 nel campo *Start*. Il risultato della singola conversione viene poi immagazzinato, nella stessa maniera della modalità *hardware*, nel registro ADDR. La fine della conversione può essere segnalata da un *interrupt*, oppure interrogando il *bit Done* del registro ADDR.
Infine, nella modalità di conversione *software*, è possibile far partire una conversione anche quando si presenta un fronte sui *pin* P0.16 o P0.22: se il fronte dev'essere di salita o di discesa viene specificato nel campo *Edge* del registro ADCR.

4.5 Progetto delle CPLD per la gestione del banco di DDC112

La logica di controllo di ECSIC32 prevede l'utilizzo di due CPLD (*Complex Programmable Logic Device*) ispMACHM4A5-64/32 della Lattice Semiconductor [60] da affiancare al microcontrollore allo scopo di:

- generare i segnali necessari al corretto funzionamento dei DDC112 nelle diverse modalità d'integrazione, ovvero: CONV, DXMIT, DCLK e TEST
- fornire ai DDC112 tutte le informazioni (impostate da remoto e gestite dal microcontrollore) riguardanti: tempo d'integrazione, funzionamento normale o in modalità "*test*", numero di impulsi di *test*, integrazione a 32 o 64 canali.

L'utilizzo del solo microcontrollore LPC2103 per la gestione dei DDC112 non avrebbe consentito di soddisfare le stringenti specifiche temporali dei convertitori (che richiedono segnali sincronizzati col *clock* con un ritardo massimo di 10 ns), in quanto dotato di CPU con una frequenza massima di *clock* di 60 MHz. Inoltre, un microcontrollore è in pratica una macchina sequenziale, mentre una CPLD è in grado di

generare più segnali in parallelo: questo velocizza senza dubbio il processo di lettura ed elaborazione dati.

4.5.1 Generazione del segnale CONV

Il DDC112 prevede la possibilità di impostare un tempo d'integrazione T_{INT} variabile nel *range* 50 μs ÷ 1s: utilizzando un *clock* a 10 MHz, per T_{INT} < 479.4 μs il DDC112 lavora in modalità non-continua, mentre per T_{INT} > 479.4 μs, si avrà la modalità continua.
L'idea di base per la generazione del segnale CONV è quella di progettare un contatore sincrono a 24 *bit* (GEN_CONV) che utilizzi come *clock* il *master clock* a 10 MHz: le forme d'onda corrispondenti ai vari *bit* del contatore avranno un semiperiodo corrispondente ad un dato tempo d'integrazione.
Per selezionare il tempo d'integrazione voluto, si hanno a disposizione 4 *bit* di selezione, memorizzati in ingresso da altrettanti *flip-flop* D, che commutano al fronte di salita del segnale COUT, generato alla fine di un ciclo completo di conteggio del contatore a 24 *bit*: l'abilitazione del nuovo T_{INT} in sincronismo con COUT consente di lasciare il tempo al DDC112 di completare il suo ciclo d'integrazione/conversione con il vecchio T_{INT}.
Nella tabella 4.1 sono riportati i valori delle frequenze delle onde quadre in uscita dal contatore, i relativi semiperiodi (ovvero i vari T_{INT}), il *bit* corrispondente, la relativa modalità di funzionamento del DDC112 e i *bit* di selezione per il T_{INT} desiderato: il primo segnale utile si ha per T_{INT} = 51.2 μs, fino ad arrivare ad un tempo d'integrazione massimo T_{INT} = 0.83 s in corrispondenza dell'ultimo *bit*.

Frequenza	**Semiperiodo (T_{INT})**	***Bit* Contatore**	**Modalità**	***Bit* selezione**
9.76 kHz	51.2 μs	q_9	NCONT	0000
4.88 kHz	102.4 μs	q_{10}	NCONT	0001
2.44 kHz	204.8 μs	q_{11}	NCONT	0010
1.22 kHz	409.6 μs	q_{12}	NCONT	0011
610.35 Hz	819.2 μs	q_{13}	CONT	0100
305.17 Hz	1.63 ms	q_{14}	CONT	0101
152.68 Hz	3.27 ms	q_{15}	CONT	0110
76.29 Hz	6.55 ms	q_{16}	CONT	0111
38.14 Hz	13.1 ms	q_{17}	CONT	1000
19.07 Hz	26.2 ms	q_{18}	CONT	1001
9.53 Hz	52.4 ms	q_{19}	CONT	1010
4.76 Hz	104.8 ms	q_{20}	CONT	1011
2.38 Hz	209.7 ms	q_{21}	CONT	1100
1.19 Hz	419.4 ms	q_{22}	CONT	1101
0.59 Hz	0.83 s	q_{23}	CONT	1110

Tab. 4.1 *Parametri relativi al blocco di generazione del segnale CONV*

I *bit* di selezione (da i3 a i0) sono i segnali di controllo di un *multiplexer* a 15 ingressi (MUX_CONV), ognuno dei quali corrisponde ad un possibile tempo d'integrazione. In uscita dal *multiplexer* si ha un segnale Y, che è direttamente il segnale CONV nel caso

di modalità continua (a meno della successiva sincronizzazione col *master clock*), mentre in modalità non-continua è la parte del segnale di CONV relativa alla sola fase di integrazione: in questo caso si ha infatti il problema di permettere al DDC112 di avere il tempo necessario a completare i cicli di conversione (cicli m/r/az) per entrambi i canali, un tempo che non è mai inferiore a 934.8 µs. Per risolvere il problema è stato progettato un sistema di controllo che in modalità non-continua seleziona Y per la fase d'integrazione, mentre in fase di conversione "maschera" Y e seleziona l'uscita Q_{14} del contatore GEN_CONV, la prima forma d'onda utile per rispettare la tempistica dei cicli m/r/az. Un *multiplexer* a 2 ingressi (MUX_MASK) permette di avere alternativamente in uscita Y per la fase d'integrazione e Q_{14} per la fase di conversione, "componendo" l'andamento del segnale CONV richiesto per il corretto funzionamento del DDC112 in modalità non-continua. Il controllo di MUX_MASK avviene attraverso il segnale MASK, generato da un circuito logico in grado di attivare il mascheramento di Y in fase di conversione del DDC112: tale circuito attiva MASK nel caso in cui il DDC112 lavori in modalità non-continua e stia effettuando un ciclo di conversione (vedi fig. 4.66).

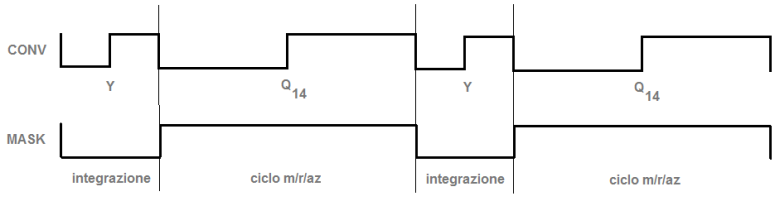


Fig. 4.66 *Sistema di mascheramento azionato in modalità non-continua*

In fig. 4.67 è riportato il blocco circuitale relativo alla generazione del segnale CONV:

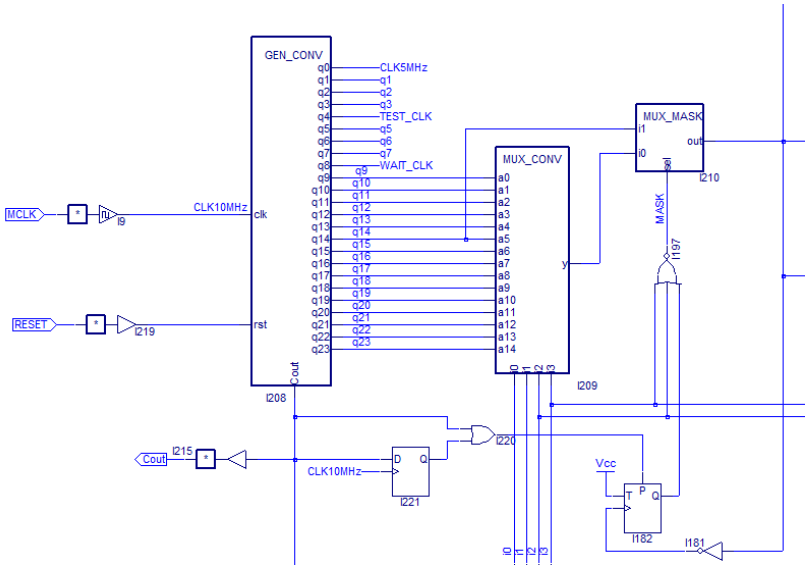


Fig. 4.67 *Schema circuitale per la generazione del segnale CONV*

Per completezza vengono inoltre riportati qui di seguito i codici ABEL dei diversi blocchi utilizzati per la generazione del segnale CONV (nell'ordine: GEN_CONV, MUX_CONV e MUX_MASK):

1)

```
MODULE gen_conv

TITLE 'contatore per la generazione di CONV a diversi tempi di
       integrazione'


"inputs"
       clk,rst                    pin;

"outputs"
       q0..q23                    pin istype 'reg';
       Cout                       pin istype 'com';

"sets"
       count=[q23..q0];           "dal MSB al LSB"

equations

       count.clk=clk;
       count.ar=rst;
       count:=count+1;
       when (count==16777215) then Cout=1 else Cout=0;

END
```

2)

```
MODULE mux_conv

TITLE 'multiplexer per la gestione dei CONV in base al tempo di
       integrazione scelto'

"inputs"
       i0..i3, a0..a14            pin;

"output"
       y                          pin istype 'com';

"sets"
       sel=[i3..i0];

equations

       when sel==0 then y=a0;
       when sel==1 then y=a1;
       when sel==2 then y=a2;
       when sel==3 then y=a3;
       when sel==4 then y=a4;
       when sel==5 then y=a5;
       when sel==6 then y=a6;
       when sel==7 then y=a7;
```

```
        when sel==8 then y=a8;
        when sel==9 then y=a9;
        when sel==10 then y=a10;
        when sel==11 then y=a11;
        when sel==12 then y=a12;
        when sel==13 then y=a13;
        when sel==14 then y=a14;

END
```

3)

```
MODULE mux_mask

TITLE 'multiplexer per la gestione del CONV in modalità ncont'

"inputs"
        sel, i0..i1         pin;

"output"
        out                 pin istype 'com';

equations

        when !sel then out=i0 else out=i1;

END
```

4.5.2 Generazione di segnali necessari al prelievo dei dati (*data retrieval*)

Il metodo più semplice per gestire il prelievo dei dati dal DDC112, è la modalità di prelievo prima della commutazione di CONV (*before CONV toggles*) [61].

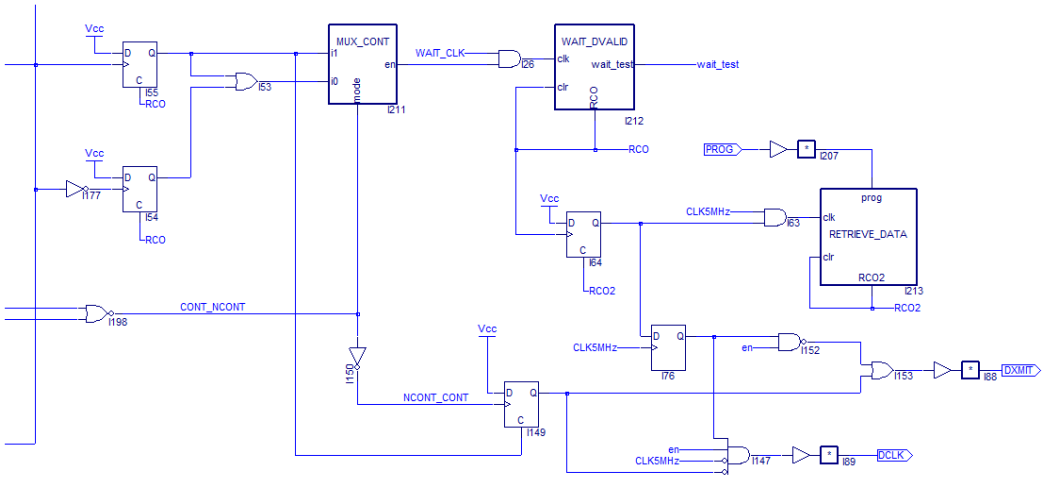


Fig. 4.68 *Schema circuitale relativo alla gestione del prelievo dei dati*

Il prelievo dei dati relativi ad una singola operazione di integrazione/conversione ha inizio ad ogni fronte di CONV (salita o discesa) nel caso di modalità di funzionamento continua del DDC112, mentre nella modalità non-continua ha inizio ad ogni fronte di salita del CONV: a tale scopo due rilevatori di fronte di CONV (il primo per il fronte di salita, il secondo per il fronte di discesa) segnalano alla logica circuitale successiva l'inizio del processo di *data retrieval*.
Il blocco MUX_CONT (di cui qui di seguito è riportato il codice ABEL) è un *multiplexer* a due ingressi e un *bit* di selezione, e si occupa di far partire il processo di prelievo dei dati nelle due modalità continua e non-continua, fornendo in uscita un segnale di *enable* per la logica circuitale successiva: più precisamente, il segnale di *enable* verrà generato nel caso continuo ad ogni fronte di CONV, mentre nel caso non-continuo verrà generato sui soli fronti di salita di CONV.

```
MODULE mux_cont

TITLE 'multiplexer per la gestione del data retrieval
       nelle due modalità cont e ncont'

"inputs"
       mode, i0..i1        pin;

"output"
       en                          pin istype 'com';

equations

       when !mode then en=i0 else en=i1;

END
```

Il blocco WAIT_DVALID è un contatore a 4 *bit* dedicato a generare un intervallo di attesa, necessario per evitare il prelievo dei dati prima del completamento delle operazioni di integrazione/conversione da parte del DDC112. Da specifiche, tale tempo di attesa è di circa 421 μs, ed è indipendente sia dal tempo d'integrazione che dalla modalità di funzionamento: a tale scopo è stato utilizzato come *clock* del contatore il segnale q8 (poi denominato WAIT_CLOCK) uscente dal blocco principale GEN_CONV. Il segnale WAIT_CLOCK ha periodo pari a 51.2 μs: il conteggio arriva fino a 9, introducendo un tempo di attesa di 435.2 μs (che rispetta dunque le specifiche). La scelta di un *clock* diverso dal *master clock* a 10 MHz ha consentito di ridurre il numero di *bit* del contatore WAIT_DVALID, ottimizzando l'occupazione di macrocelle della CPLD e semplificando la logica. Alla fine di ogni conteggio del tempo di attesa, WAIT_DVALID viene resettato, ed il segnale di *reset* attiva la parte circuitale successiva, dando il via alla generazione dei segnali DXMIT e DCLK necessari al prelievo dei dati. Contemporaneamente, la fine del conteggio di attesa va a resettare i rilevatori di fronte di CONV, rendendoli pronti ad un nuovo ciclo di integrazione/conversione/prelievo dati. Il codice ABEL del blocco WAIT_DVALID è il seguente:

```
MODULE wait_dvalid

TITLE 'contatore per attesa del DVALID'

"inputs"
       clk,clr                    pin;

"outputs"
       q0..q3                     pin istype 'reg';
       RCO,wait_test              pin istype 'com';

"sets"
       count=[q3..q0];            "dal MSB al LSB"

equations

       count.clk=clk;
       count.ar=clr;
       count:=count+1;
       when count==9 then count:=0;
       when count==9 then RCO=1 else RCO=0;
       when count==1 then wait_test=1 else wait_test=0;

END
```

L'impulso di *reset* del blocco WAIT_DVALID, attraverso un semplice circuito logico di controllo, attiva DXMIT (che è attivo basso) e DCLK, un *clock* di frequenza pari a 5 MHz che deve essere abilitato (ovvero messo in uscita) soltanto per il tempo strettamente necessario al trasferimento dei dati dal blocco dei DDC112 alla memoria, e dunque soltanto per il tempo in cui DXMIT è basso: per gestire la tempistica di questo delicato processo è stato progettato il blocco RETRIEVE_DATA.
Il blocco RETRIEVE_DATA è un contatore programmabile gestito da un *clock* a 5 MHz (la stessa frequenza di DCLK), ed ha il compito di contare il numero di *bit* che il DDC112 sta trasferendo alla memoria. E' possibile indicare da remoto, attraverso il segnale PROG, se l'elettronica di lettura che si sta utilizzando è a 8, 16, 32 o 64 canali: visto che il pacchetto di dati in uscita da un singolo DDC112 è composto da 20 *bit*, a seconda del numero di canali il blocco RETRIEVE_DATA si occuperà di contare fino a 160, 320, 640 o 1280, generando alla fine del conteggio un segnale di *reset* indicatore della fine del processo di prelievo e trasferimento in memoria dei dati. Tale segnale di *reset* disattiva DXMIT (riportandolo alto), disabilita DCLK e resetta la logica di controllo che si occupava di abilitare RETRIEVE_DATA al conteggio dei *bit* prelevati. Qui di seguito viene riportato il codice ABEL del blocco RETRIEVE_DATA programmato per misure a 32 o 64 canali:

```
MODULE retrieve_data

TITLE 'contatore per la gestione di pacchetti da 20 bit
       provenienti da 32 o 64 canali'

"inputs"
       clk,clr,prog        pin;
```

```
"outputs"
      q0..q10                    pin istype 'reg';
      RCO2                       pin istype 'com';

"sets"
      count=[q10..q0]; "dal MSB al LSB"

equations

      count.clk=clk;
      count.ar=clr;
      count:=count+1;
      when !prog "32 canali" then
            {when count==640 then count:=0;
            when count==640 then RCO2=1 else RCO2=0;}
      else "64 canali"
            {when count==1280 then count:=0;
             when count==1280 then RCO2=1 else RCO2=0;}

END
```

4.5.3 Gestione della modalità "*test*"

In modalità "*test*", in base alle specifiche del componente DDC112 [62], è possibile inviare al suo ingresso, in caso di segnale RANGE di valore massimo (RANGE0 = 1, RANGE1 = 1 e RANGE2 = 1), fino a 350 pC di carica di *test*. A tale scopo, una parte della CPLD si occuperà di generare un segnale TEST corrispondente alla quantità di carica di *test* voluta dall'utente: essendo la carica di *test* inviata al DDC112 attraverso una serie di pacchetti successivi da 13 pC, sono stati riservati per la modalità "*test*" 5 *bit* di selezione, per un *range* complessivo di carica di *test* 13 ÷ 338 pC. Nella tabella 4.2 sono riportate le diverse quantità di carica di *test* e il corrispondente valore dei *bit* di selezione da remoto.

Carica di *test*	***Bit* selezione**	**Carica di *test***	***Bit* selezione**
13 pC	00000	182 pC	01101
26 pC	00001	195 pC	01110
39 pC	00010	208 pC	01111
52 pC	00011	221 pC	10000
65 pC	00100	234 pC	10001
78 pC	00101	247 pC	10010
91 pC	00110	260 pC	10011
104 pC	00111	273 pC	10100
117 pC	01000	286 pC	10101
130 pC	01001	299 pC	10110
143 pC	01010	312 pC	10111
156 pC	01011	325 pC	11000
169 pC	01100	338 pC	11001

Tab. 4.2 *Parametri relativi alla gestione della modalità "test"*

Per ottimizzare l'occupazione di spazio della CPLD, 4 dei 5 *pin* destinati ai *bit* di selezione per la carica di *test* (SEL0, SEL1, SEL2 e SEL3) sono condivisi con i *bit* di selezione del tempo d'integrazione, mentre il *pin* SEL4 viene riservato al solo funzionamento in modalità "*test*" (in fase di misura vera e propria SEL4 = 0, essendo previsto l'utilizzo di soli 4 *bit* di selezione).
Per destinare i *bit* di selezione all'una o all'altra modalità di funzionamento del DDC112 ("*test*" o misura standard), è stato attribuito un *pin* d'ingresso della CPLD ad un segnale di controllo TEST_OFF, basso se si vuole lavorare in modalità "*test*" oppure alto se si vuole lavorare in condizioni di normale integrazione.
Il blocco SELECT_MODE è un *demultiplexer* a 5 ingressi (i 5 *bit* di selezione) e 1 segnale di controllo (TEST_OFF):

- se TEST_OFF = 1 il DDC112 lavorerà in condizioni di normale integrazione, ed i 4 *bit* di selezione corrispondenti al tempo d'integrazione scelto verranno inviati al blocco GEN_CONV.
- se TEST_OFF = 0 il DDC112 lavorerà in modalità "*test*", e i 5 *bit* di selezione corrispondenti alla quantità di carica di *test* voluta verranno inviati al blocco TEST.

Si è scelto di utilizzare un solo segnale CONV per il funzionamento in modalità "*test*", più precisamente il CONV corrispondente ad un tempo d'integrazione T_{INT} = 1.63 ms (modalità d'integrazione continua): pertanto, in caso di modalità "*test*" (TEST_OFF = 0), i *bit* inviati dal blocco SELECT_MODE al blocco GEN_CONV saranno sempre quelli corrispondenti ad un tempo d'integrazione di 1.63 ms.
Qui di seguito viene per completezza riportato il codice ABEL relativo al blocco SELECT_MODE:

```
MODULE select_mode

TITLE 'demultiplexer selezione modalità test o integrazione'

"inputs"
       sel0..sel4,TEST_OFF pin;

"outputs"
       i0..i4,t0..t4       pin istype 'com';

equations

       when TEST_OFF then [i0..i4]=[sel0..sel4];
       when TEST_OFF then [t0..t4]=[0,0,0,0,0];
       when !TEST_OFF then [i0..i4]=[1,0,1,0,0];
       when !TEST_OFF then [t0..t4]=[sel0..sel4];

END
```

Il blocco TEST è un particolare contatore a 6 *bit*, dunque uno in più dei 5 *bit* selezione: è stato infatti programmato in modo tale da arrestarsi quando il conteggio è arrivato esattamente al doppio del numero specificato nei 5 *bit* di selezione. In questo modo si

può utilizzare come segnale TEST (dopo un'opportuna negazione prevista nel *datasheet* del DDC112 [48]) il primo *bit* di uscita del contatore, che è un segnale di frequenza pari alla metà di quella del *clock* utilizzato, ed è dunque formato da un numero d'impulsi esattamente pari al numero specificato nei 5 *bit* di selezione. In fig. 4.69 è riportato l'intero schema circuitale relativo alla gestione della modalità "*test*" e della sequenza di *power-up* (vedi par. 4.5.4).

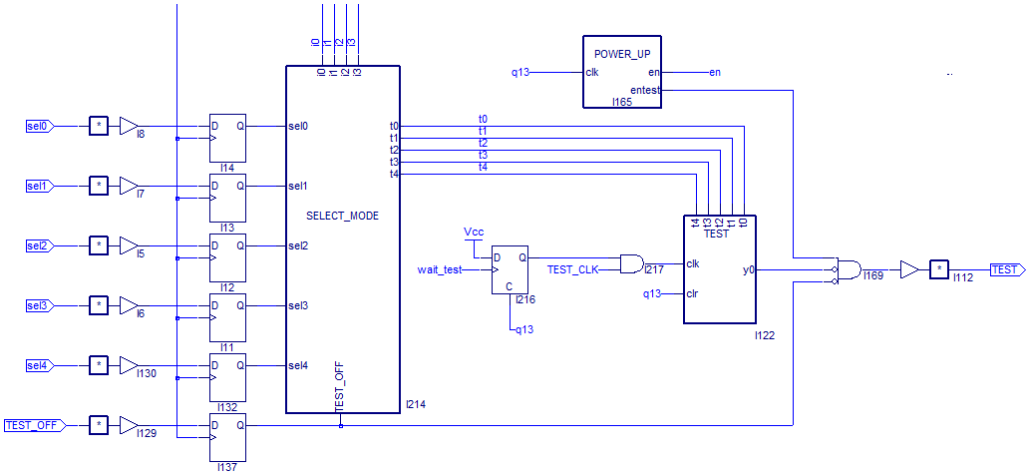


Fig. 4.69 *Schema circuitale relativo alla gestione della modalità "test"*

La durata minima di un impulso di *test* dev'essere, in base alle specifiche del DDC112, di 5.4 µs: pertanto è stato scelto per il blocco TEST un *clock* (TEST_CLK) di periodo 3.2 µs, che corrisponde ad una durata degli impulsi di *test* in uscita pari a 6.4 µs. Gli impulsi del segnale TEST non devono sovrapporsi con gli impulsi del segnale DCLK, ed inoltre non devono partire in coincidenza di un fronte di CONV: per questo TEST_CLK è abilitato da un segnale WAIT_TEST (generato dal blocco WAIT_DVALID) che consente di evitare sovrapposizioni e rispettare le specifiche.
Il codice ABEL per il blocco TEST è il seguente:

```
MODULE test

TITLE 'contatore impulsi di test'

"inputs"
      clk,clr,t0..t4            pin;

"outputs"
      y0..y5                    pin istype 'reg';

"sets"
      count=[y5..y0];
      test=[t4..t0,0];

equations

      count.clk=clk;
```

```
        count.ar=clr;
        when count==test then count:=count else count:=count+1;

END
```

4.5.4 Gestione della sequenza di *power-up*

Il blocco POWER_UP si occupa di gestire l'integrato DDC112 all'inizio di una sessione di misura, ovvero all'accensione di ECSIC32. Visto che la scelta della modalità di funzionamento ("*test*" o integrazione standard) e del tempo d'integrazione (o dei pacchetti di carica) diventa operativa solo dopo un intero ciclo di conteggio del blocco GEN_CONV, si è scelto di far partire ECSIC32 con una configurazione di *default*, più precisamente in modalità "*test*" a 13 pC di carica (nessun impulso di *test*).
Per rispettare le specifiche sulla sequenza di *power-up* previste nel datasheet del DDC112 in caso di partenza di CONV a livello basso (prima integrazione sul canale B), occorre fare in modo di avere un evoluzione temporale dei segnali in uscita come in fig. 4.70:

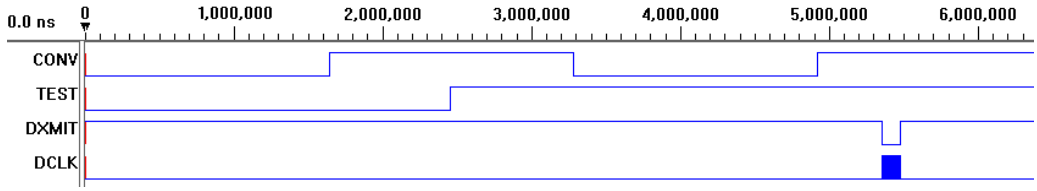


Fig. 4.70 *Diagramma temporale di una corretta sequenza di power-up*

La prima specifica è relativa alla prima integrazione eseguita dal DDC112, che in questo caso avviene sul canale B (CONV basso, terzo semiperiodo): il tempo che intercorre dall'accensione del sistema alla prima integrazione dev'essere superiore a 100 μs. Proprio per questo motivo si è scelto di utilizzare come CONV di *default* (e per semplicità di usarlo anche in ogni successiva operazione in modalità "*test*") il segnale q14 generato dal blocco GEN_CONV: il periodo di tale segnale è 1.6 ms, e questo garantisce l'inizio delle integrazioni in una situazione robusta e soprattutto stabile, lontana da eventuali transitori del sistema di alimentazione.
Il segnale TEST da 13 pC deve attivarsi (ovvero andare alto) nel secondo semiperiodo di CONV, in anticipo rispetto all'inizio della prima integrazione (terzo semiperiodo), e ben lontano dai fronti di CONV.
I segnali DXMIT e DCLK devono invece essere abilitati in uscita solo a partire dal quarto semiperiodo di CONV, in quanto la prima integrazione/conversione avviene nel terzo semiperiodo di CONV, e dunque solo dal quarto in poi si hanno a disposizione dati validi.
Il blocco POWER_UP è un semplice contatore a 2 *bit* avente come *clock* il segnale q13 del blocco GEN_CONV, scelto perché i suoi fronti di salita si hanno esattamente in mezzo a un semiperiodo del CONV di *default*, in modo tale che il segnale TEST si attivi ben lontano da un fronte di CONV.

Il blocco POWER_UP genera due segnali di *enable*:

- il segnale ENTEST viene generato quando il conteggio arriva a 2 (dunque nel secondo semiperiodo di CONV) e, unitamente alla condizione TEST_OFF=0, mette alto il segnale TEST.
- Il segnale EN viene generato quando il conteggio arriva a 3 (dunque nel terzo semiperiodo di CONV) e abilita in uscita DXMIT e DCLK.

Ovviamente, una volta esaurita la sequenza di *power-up*, entrambi i segnali di *enable* devono rimanere attivati per tutta la durata della sessione di misura: per questo motivo è stato previsto che il conteggio del blocco POWER_UP rimanesse ancorato a 3 e non venisse mai resettato (vedi codice ABEL qui di seguito).

```
MODULE power_up

TITLE 'sequenza di power-up per DXMIT e TEST'

"inputs"
       clk                pin;

"outputs"
       q0,q1              pin istype 'reg';
       en,entest          pin istype 'com';

"sets"
       count=[q1..q0];

equations

       count.clk=clk;
       when count==3 then count:=3 else count:=count+1;
       when count==3 then en=1 else en=0;
       when (count==3)#(count==2) then entest=1 else entest=0;

END
```

4.5.5 Soluzione dei problemi relativi al cambiamento di modalità

In caso di passaggio da modalità continua a modalità non-continua, il segnale CONV generato dalla logica descritta nei precedenti paragrafi presenta dei problemi. L'evoluzione temporale del segnale CONV a cavallo del passaggio di modalità è infatti la seguente:

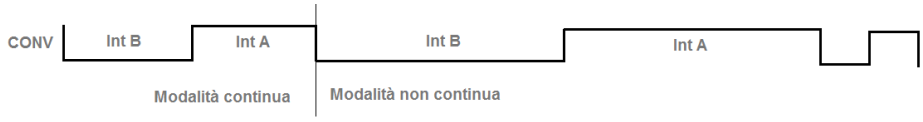


Fig. 4.71 *Errato andamento del segnale CONV in occasione di un passaggio cont – ncont*

Questa configurazione non va bene, in quanto il DDC112 interpreta i primi due semiperiodi di CONV dopo il passaggio di modalità come periodi d'integrazione su B e su A, mentre dovrebbero essere interpretati come stati di attesa.
In base alle specifiche del DDC112, il segnale CONV, in caso di transizione da modalità continua a modalità non-continua (e prima integrazione su B), dovrebbe avere un andamento di questo tipo [63]:

Fig. 4.72 *Corretto andamento del segnale CONV in occasione di un passaggio cont – ncont*

In tal modo il DDC112 interpreta correttamente il segnale CONV una volta avvenuto il passaggio in modalità non-continua: avviene una prima integrazione su B, seguita da due stati di attesa (il primo relativo all'ultima integrazione continua sul canale A, il secondo alla prima integrazione non-continua sul canale B), dopo di che si ha il normale ciclo di integrazioni/conversioni su A e B in modalità non-continua.
Per ottenere questo andamento del segnale di CONV è stata progettata una parte apposita della logica di controllo, evidenziata in fig. 4.73:

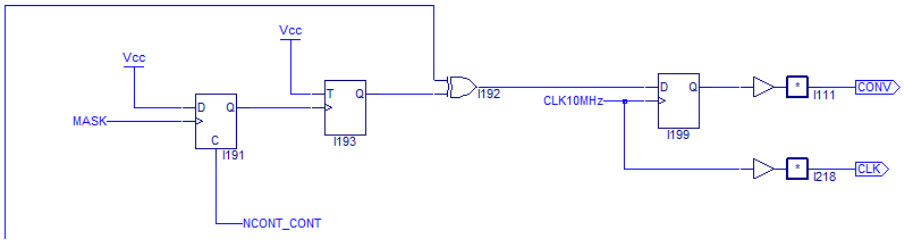


Fig. 4.73 *Estensione del blocco di generazione del segnale CONV per la gestione del cambio di modalità*

In caso di transizione da modalità continua a modalità non-continua, il segnale CONV viene invertito al primo fronte di MASK dopo la transizione (a tal proposito, il segnale MASK si attiva con un certo ritardo rispetto al cambio di modalità, in modo da consentire senza problemi la prima integrazione non-continua su B).
Supponiamo di lavorare in modalità continua: in tal caso la porta XOR agisce da *buffer* per il segnale CONV. In caso di passaggio alla modalità non-continua, al primo fronte di MASK si ha la commutazione del *flip-flop* T, e questo fa agire la porta XOR da *inverter* per il segnale CONV. Un eventuale successivo ritorno alla modalità continua rimette a 0 l'uscita del *flip-flop* D, rendendo il sistema pronto ad una nuova inversione di CONV in caso di transizione dalla modalità continua alla modalità non-continua.
L'ultimo *flip-flop* D prima dell'uscita sincronizza il segnale CONV con il *master clock* a 10 MHz.

In caso invece di transizione da modalità non-continua a modalità continua, occorre fare attenzione alla generazione dei segnali DXMIT e DCLK relativi al prelievo dei dati. L'evoluzione temporale dei segnali CONV, DXMIT e DCLK in una situazione di questo tipo è rappresentata in fig. 4.74:

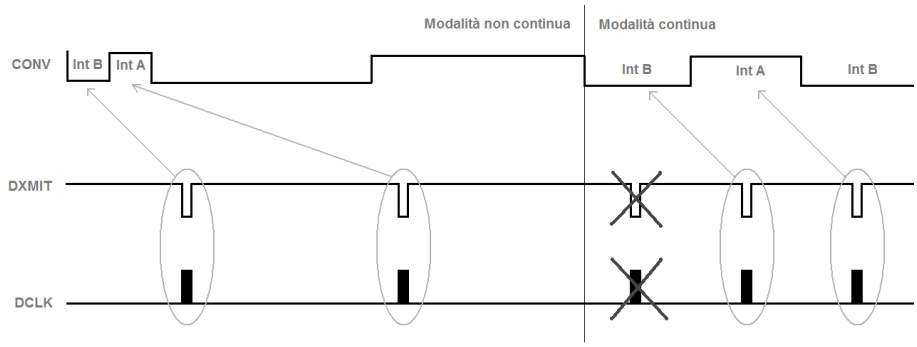


Fig. 4.74 *Gestione dei segnali DXMIT e DCLK in caso di passaggio ncont – cont*

Occorre eliminare il primo impulso DXMIT (e i relativi impulsi di DCLK) dopo il passaggio alla modalità continua: i dati prelevati in questa situazione, come si evince dalla figura, non sono riferiti ad alcun periodo d'integrazione, e dunque sono completamente inattendibili. A tale scopo, non appena avviene una transizione da modalità non-continua a modalità continua, si attiva una semplice logica che disabilita l'uscita di DXMIT e DCLK; il successivo fronte di CONV farà poi ripartire il regolare prelievo dei dati, riattivando DXMIT e DCLK.

4.5.6 Verifica del funzionamento della CPLD per la gestione del banco di DDC112

In questo paragrafo vengono riportati e commentati alcuni diagrammi temporali di simulazione (*timing simulation*) ottenuti con ispLEVER Project Navigator. Per effettuare la simulazione in tempi ragionevoli, il numero di *bit* del contatore dei tempi d'integrazione (che è il contatore GEN_CONV a 24 *bit*) è stato ridotto a 19. E' stato posto PROG = 0, ovvero si prevede la simulazione di un'elettronica a 32 canali.
La simulazione prevede la verifica di tutte le modalità di funzionamento del DDC112, con particolare riferimento ai cambiamenti di modalità che possono essere imposti dall'utente nel corso di operazioni di misura.

1° ciclo: sequenza di POWER-UP
La sequenza di *power-up* all'accensione del sistema avviene correttamente (fig. 4.75): vengono rispettati i tempi t_{32} di inizializzazione e t_{33} di *release* (entrambi pari a 1.6 ms, dunque ben superiori rispetto ai 50 µs imposti come durata minima dalle specifiche). Come da progetto, il sistema parte di *default* effettuando un ciclo di *test*, con l'invio di un singolo pacchetto di carica da 13 pC (nessun impulso di *test*): il segnale TEST va alto nel secondo semiperiodo di CONV, rispettando la specifica su t_1 (*setup time for test mode enable*, valore minimo 100 ns). La prima integrazione avviene correttamente nel terzo semiperiodo di CONV, e il prelievo dei dati relativi alla prima integrazione avviene (con DXMIT e DCLK) nel semiperiodo di CONV successivo. Lo schema

d'integrazione è dunque BABABA, etc. DXMIT e DCLK seguono in modo corretto l'evolversi delle integrazioni.

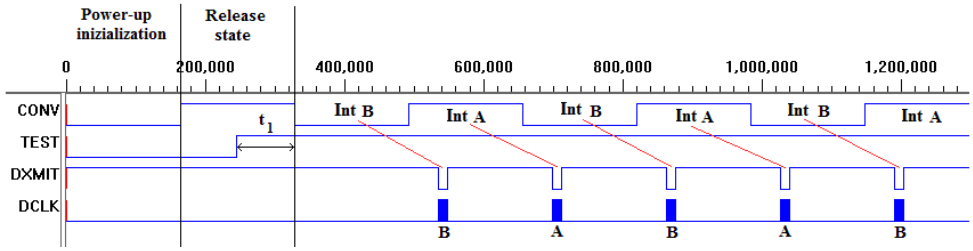


Fig. 4.75 *Simulazione del 1° ciclo*

Nel diagramma qui di seguito (fig. 4.76) è riportato un particolare dei segnali DXMIT e DCLK: si noti come il primo dei 640 impulsi di DCLK (20 x 32 = 640 *bit*) si presenti solo dopo l'attivazione (bassa) di DXMIT.

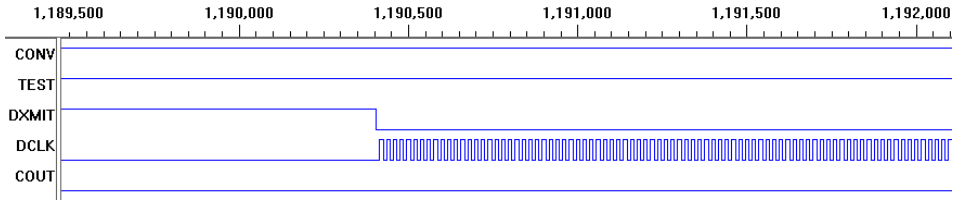


Fig. 4.76 *Particolare della simulazione del 1° ciclo*

2° ciclo: passaggio TEST – modalità NON-CONTINUA

All'arrivo del primo COUT (vedi fig. 4.77) si disattiva correttamente il segnale TEST, ed avviene il passaggio alla modalità d'integrazione non-continua (T_{INT} = 51.2 µs). Il segnale CONV s'inverte (vedi par. 4.5.5), in quanto il segnale CONV in fase di *test* ha il tipico andamento del CONV in modalità d'integrazione continua. L'ultima integrazione di *test* è sul canale A, seguita da un'integrazione sul canale B, che è la prima integrazione in modalità non-continua: a questa seguono due stati di attesa (per completare i cicli di conversione), dopo di che parte lo schema d'integrazione ABABAB, etc.

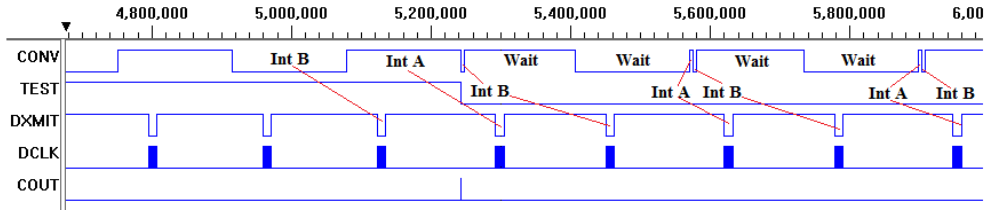


Fig. 4.77 *Simulazione del 2° ciclo*

Effettuando uno zoom sull'arrivo del COUT e la fine della modalità "*test*", si può notare (fig. 4.78) come il fronte di discesa del segnale TEST avvenga circa 200 ns prima del fronte di CONV, rispettando la specifica su t_2 (*setup time for test mode disable*, valore minimo 100 ns).

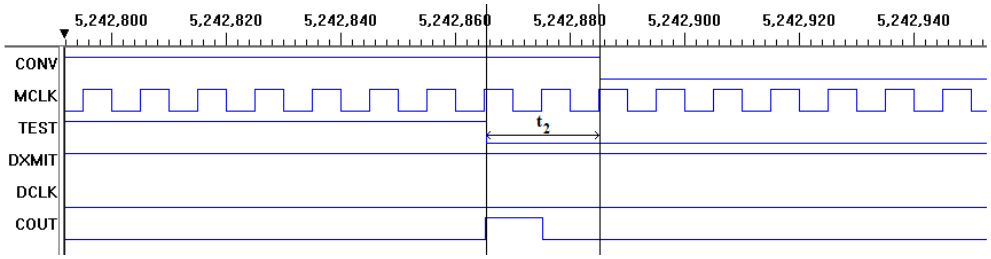


Fig. 4.78 *Particolare della simulazione del 2° ciclo*

3° ciclo: passaggio modalità NON-CONTINUA – modalità NON-CONTINUA

Si passa dalla modalità non-continua con T_{INT} = 51.2 µs, ad un'altra modalità non-continua, ma con T_{INT} = 409.6 µs: si è scelto di simulare questo passaggio (fig. 4.79) perché il tempo d'integrazione T_{INT} = 409.6 µs è l'ultimo tempo utile per lavorare in modalità non-continua, e rappresenta una situazione critica per il prelievo dei dati. Le specifiche su DXMIT sono però correttamente rispettate: t_{29} (*data retrieval start up after edge of* CONV) = 25.6 µs > 10 µs (min.), t_{30} (*1st ncont mode data ready*) = 435 µs > 421.1 µs (min.), ed infine t_{31} (*2nd ncont mode data ready*) = 1.23 ms > 454.8 µs (min.).

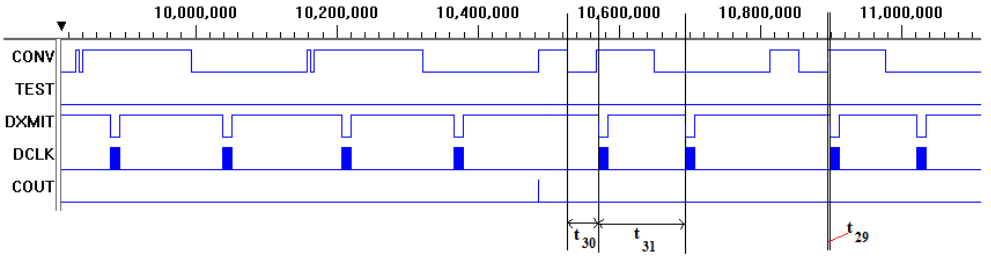


Fig. 4.79 *Simulazione del 3° ciclo*

4° ciclo: passaggio modalità NON-CONTINUA – modalità CONTINUA

Si passa dall'ultimo T_{INT} utile in modalità non-continua al primo T_{INT} utile in modalità continua, ovvero T_{INT} = 819.2 µs (fig. 4.80). Non si ha alcuna inversione del segnale CONV, e inoltre si ha correttamente la soppressione del primo DXMIT (e conseguentemente di DCLK) dopo l'arrivo di COUT, come dev'essere ad ogni passaggio di modalità non-continua – continua.

Anche T_{INT} = 819.2 µs prevede un CONV critico per il prelievo dei dati, ma le specifiche temporali su DXMIT sono comunque rispettate: t_{27} (*cont mode data ready*) = 256 µs > 10 µs (min.) e t_{28} (*data retrieval shutdown before edge of* CONV) = 435 µs > 421.2 µs (min.).

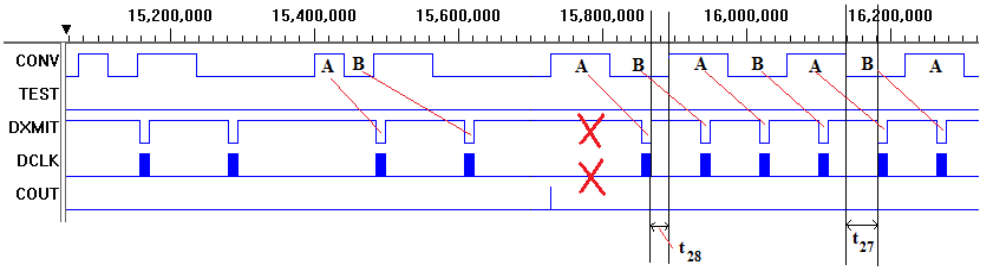


Fig. 4.80 *Simulazione del 4° ciclo*

5° ciclo: passaggio modalità CONTINUA – modalità CONTINUA

In questo ciclo viene simulato il passaggio fra due modalità d'integrazione continua, ma caratterizzate da due diversi tempi d'integrazione: si passa da T_{INT} = 819.2 μs a T_{INT} = 1.63 ms (fig. 4.81). Il CONV, così come dev'essere, non subisce inversioni. E' una situazione che non presenta alcuna criticità dal punto di vista delle specifiche temporali.

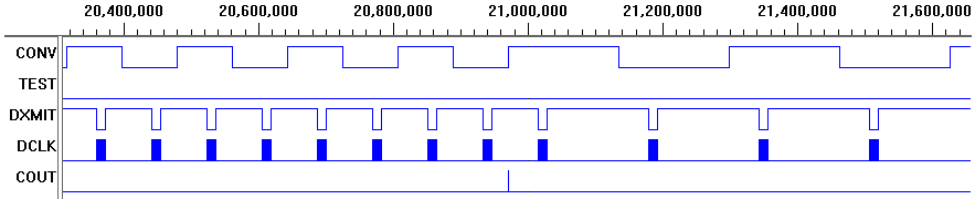


Fig. 4.81 *Simulazione del 5° ciclo*

6° ciclo: passaggio modalità CONTINUA – modalità NON-CONTINUA

In seguito al passaggio dalla modalità d'integrazione continua a quella non-continua (si è scelto, senza perdita di generalità, T_{INT} = 102.4 μs) si ha come previsto un'inversione del segnale CONV dopo l'arrivo dell'impulso COUT (fig. 4.82): l'ultima integrazione in modalità continua è sul canale B, dopo di che si ha la prima integrazione in modalità non-continua sul canale A, seguita da due stati di attesa (per il completamento dei cicli di conversione su entrambi i canali). Infine, parte lo schema d'integrazione non-continua BABABA, etc.

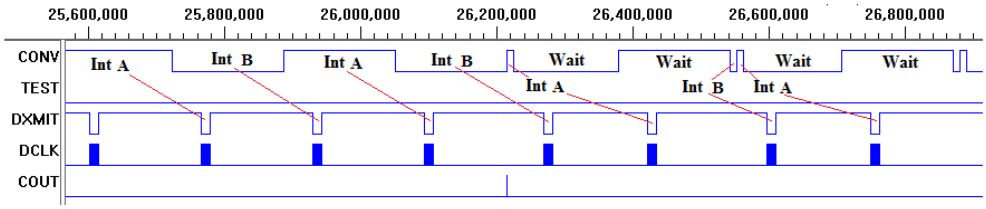


Fig. 4.82 *Simulazione del 6° ciclo*

7° ciclo: passaggio modalità NON-CONTINUA – modalità CONTINUA

Questo ciclo serve per ottenere (nel successivo 8° ciclo) il passaggio dalla modalità d'integrazione continua alla modalità "*test*", che ancora non si è verificato nel corso della simulazione. Anche in questo ciclo, come nel 4°, non si ha inversione di CONV, e si ha correttamente la soppressione dei primi DXMIT e DCLK dopo l'arrivo dell'impulso COUT che segna la fine del ciclo (fig. 4.83).

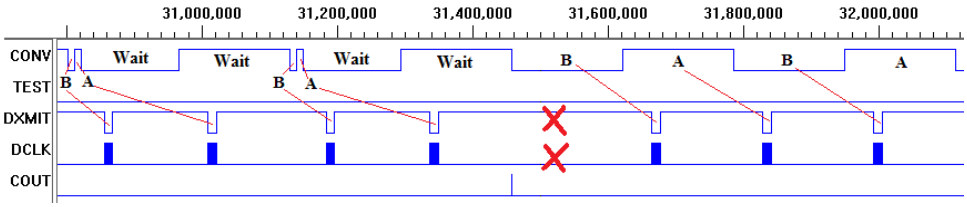


Fig. 4.83 *Simulazione del 7° ciclo*

8° ciclo: passaggio modalità CONTINUA – modalità TEST

In questo ciclo si può osservare il passaggio da un'integrazione continua standard ad una fase di *test* in cui si inietta una carica complessiva di 338 pC, la massima possibile (25 impulsi di *test*). All'arrivo dell'impulso COUT, il segnale TEST ha un fronte di salita ed inizia la fase di *test* (fig. 4.84).

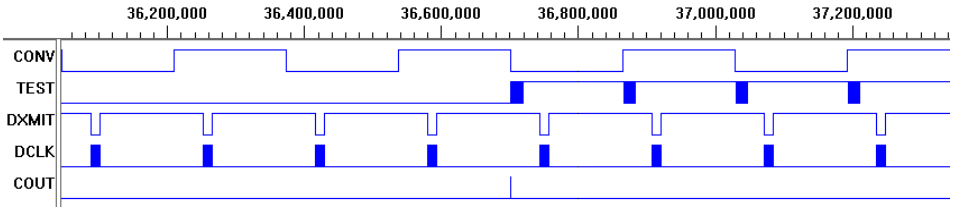


Fig. 4.84 *Simulazione dell'8° ciclo*

Si può notare dalla fig. 4.85 come si eviti ogni sovrapposizione fra gli impulsi di *test* e quelli di DCLK. Inoltre, sono rispettate le stringenti specifiche temporali sul segnale TEST: t_3 (*hold time for test mode enable*) = 27 µs > 100 ns (min.), t_4 (*from rising edge of TEST to the edge of CONV while test mode enabled*) = 30.2 µs > 5.4 µs (min.), e t_5 (*rising edge to rising edge of TEST*) = 6.4 µs > 5.4 µs (min.).

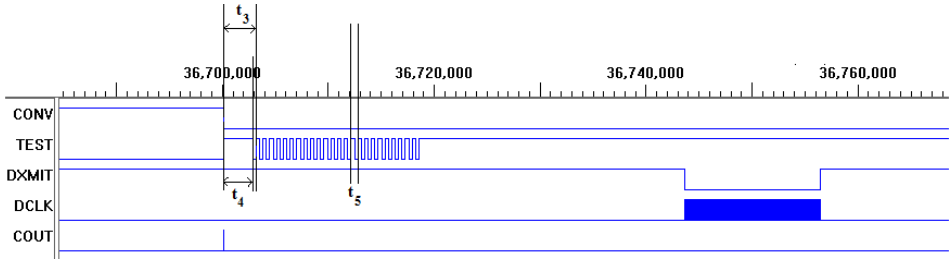


Fig. 4.85 *Particolare della simulazione dell'8° ciclo*

Quest'ultimo diagramma (fig. 4.86) fa infine vedere in dettaglio che cosa succede a cavallo dell'impulso COUT: il segnale TEST sale in anticipo rispetto al fronte di CONV di due periodi di MCLK (il *master clock* a 10 MHz), rispettando così la specifica su t_1 (*setup time for test mode enable*) = 197 ns > 100 ns (min.).

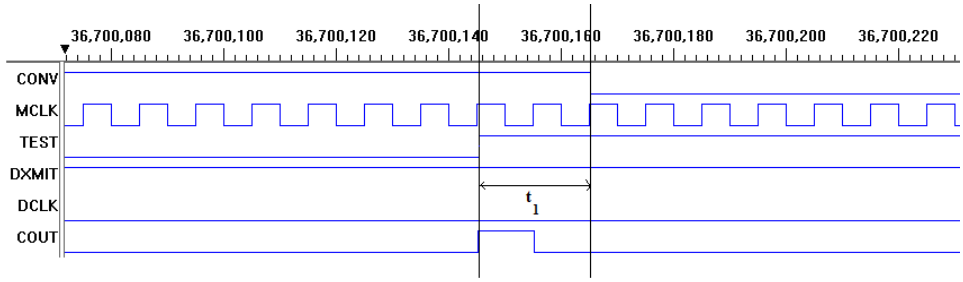


Fig. 4.86 *Particolare della simulazione dell'8° ciclo*

9° ciclo: passaggio modalità CONTINUA – modalità TEST

L'ultimo ciclo prevede la simulazione del passaggio da una fase di *test* ad una fase d'integrazione standard in modalità continua, l'unico passaggio non ancora simulato nei cicli precedenti. L'evoluzione temporale dei segnali utili avviene in maniera corretta (fig. 4.87): non si ha alcuna inversione di CONV, in quanto nella fase di *test* si utilizza un CONV valido anche per le integrazioni standard in modalità continua.

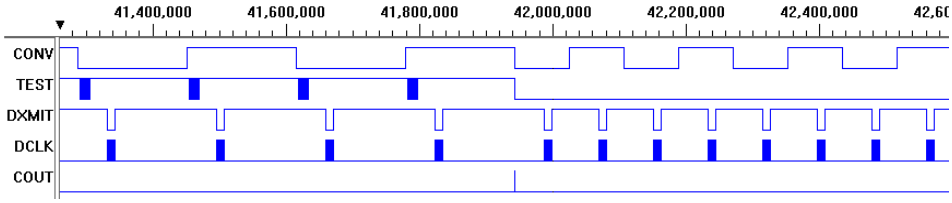


Fig. 4.87 *Simulazione del 9° ciclo*

Il segnale di TEST viene disattivato, come da specifiche, prima del fronte di CONV (fig. 4.88): si ottiene t_2 (*setup time for test mode disable*) = 197 ns > 100 ns (min.).

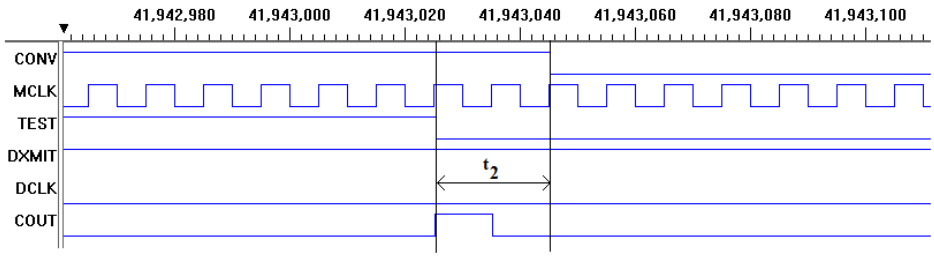


Fig. 4.88 *Particolare della simulazione del 9° ciclo*

4.6 Banco di memoria

Il banco di memoria è costituito da 5 *chip* Winbond W24257A, una RAM statica CMOS ad alta velocità (tempo di accesso 10 ÷ 20 ns) e basso consumo di potenza (400 mW in fase attiva). Ogni *chip* di memoria ha una capacità di 32 kB, è dunque è possibile memorizzare nell'intero banco 2048 *frames* (quadri) in caso di sensore a 32 canali (20 *bit* per canale). In caso di sensore a 64 canali si possono memorizzare 1024 *frames*, oppure utilizzare due banchi di memoria per mantenerne 2048 come nel caso a 32 canali.

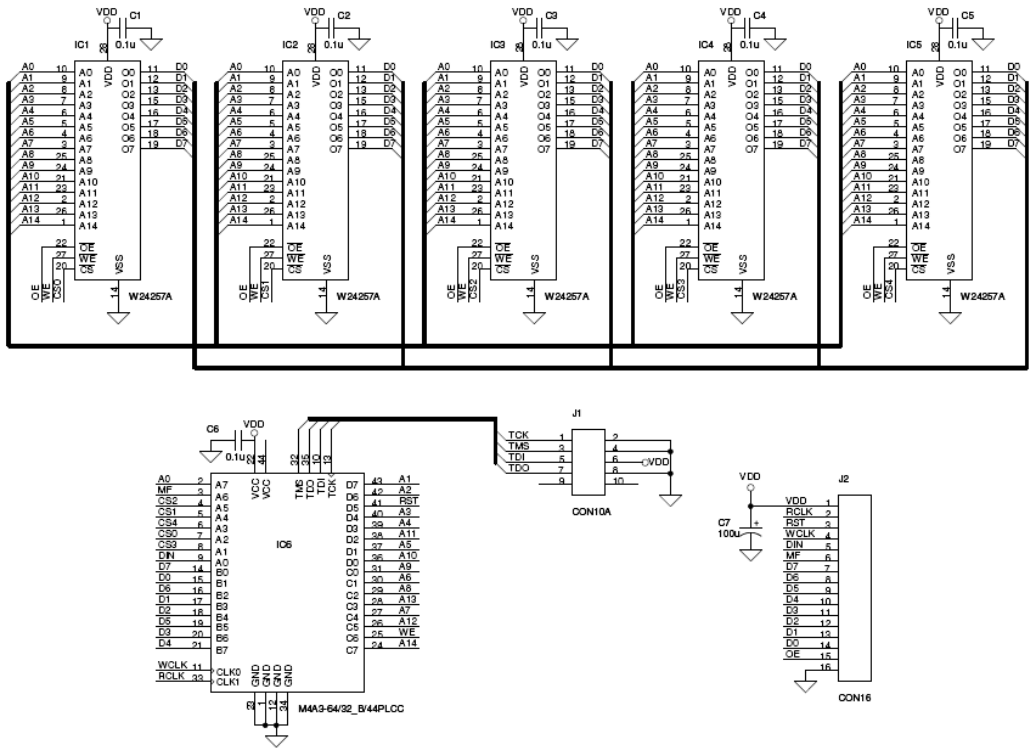


Fig. 4.89 *Banco di memoria e logica di controllo CPLD per la sua gestione*

Una logica di controllo CPLD (ispMACHM4A5-64/32 della Lattice Semiconductor [60]) è stata appositamente progettata per la gestione del banco di memoria. La memoria del singolo banco è vista dall'esterno come se fosse di tipo seriale: i dati sono automaticamente caricati in locazioni successive col procedere delle operazioni di lettura o di scrittura.
Il funzionamento è diverso per la scrittura e per la lettura:

- In scrittura i *bit* devono essere inviati singolarmente sulla linea DIN, e vengono acquisiti ad ogni fronte positivo di WCLK. Le due linee DIN e WCLK saranno, rispettivamente, collegate a DOUT e DCLK dei DDC112 connessi in *daisy*

chain. Quando il banco è completamente riempito, la CPLD attiva la linea MF (*Memory Full*), e viene inibita qualunque ulteriore operazione di scrittura.

- La lettura avviene per *byte* e le diverse locazioni sono indirizzate utilizzando la linea RCLK come *clock*. E' compito di chi legge comprendere quando si è raggiunta l'ultima locazione di memoria, altrimenti non si farà altro che ripetere nuovamente la lettura dalla prima locazione di indirizzo 0. L'operazione di scrittura è ripristinata attraverso una fase di *reset*.

N.B.: Il *reset* del sistema è determinato da opportuni comandi da parte del microcontrollore. Dopo che il microcontrollore ha terminato la lettura della RAM, deve almeno aspettare che la linea DXMIT dei convertitori sia tornata alta, cioè che sia terminata la trasmissione da parte dei DDC112 del corrente pacchetto di dati. Solo quando DXMIT è tornato alto si può ripartire con la scrittura di nuovi *bit* presenti sulla linea DIN. Tuttavia, conviene che il microcontrollore attenda l'arrivo dell'impulso di RESTART (pari al COUT fornito dalla CPLD di controllo dei DDC112), che segnala l'inizio effettivo di un nuovo ciclo di integrazione/conversione.

Il diagramma temporale dei segnali coinvolti nella memorizzazione dei *bit* che giungono sulla linea DIN dovrà essere del tipo riportato in fig. 4.90. Come detto, i segnali provenienti dai convertitori DDC112 sono DCLK e DOUT (indicati rispettivamente con WCLK e DIN come segnali d'ingresso per la memoria).

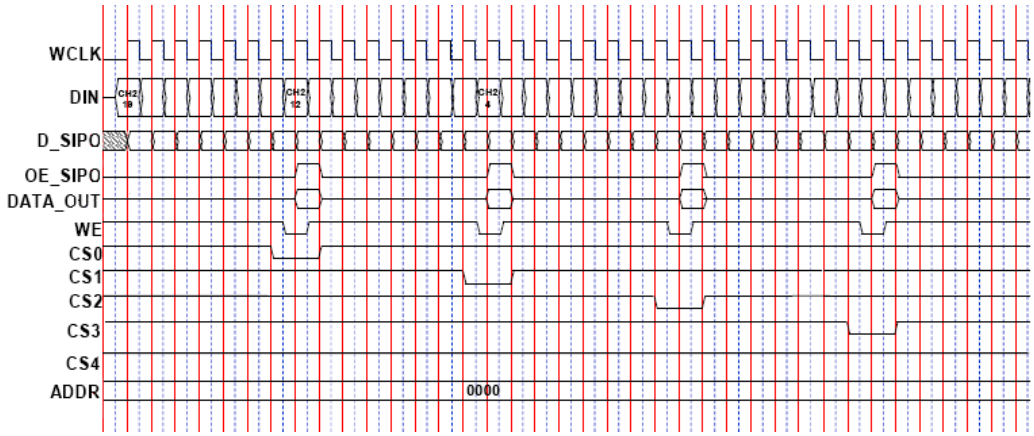


Fig. 4.90 *Diagramma temporale di scrittura in memoria*

Ad ogni fronte di WCLK il *bit* è immagazzinato in un registro a scorrimento ad ingresso seriale (D_SIPO) ed uscita parallela (DATA_OUT), dunque un registro SIPO. Completato il trasferimento di 8 *bit*, si ha un *byte* valido da memorizzare: viene allora generato il segnale WE (*Write Enable*), e viene attivato un *buffer* (attraverso il segnale OE_SIPO) che fa giungere il dato alla memoria. A cavallo di queste operazioni avviene l'attivazione dello specifico *chip*, tramite il segnale CS (*Chip Select*). Giunti all'ultimo *chip* (CS4), si deve incrementare l'indirizzo (ADDR).

Lo schema completo della CPLD progettata per la gestione del banco di memoria è visibile in fig. 4.91. Il numero di *pin* che collegano il microcontrollore al banco di memoria è limitato ai soli RCLK, SET, DATA e OE. Nel seguito del paragrafo verrà data una breve descrizione dei blocchi più importanti che compongono l'intera CPLD.

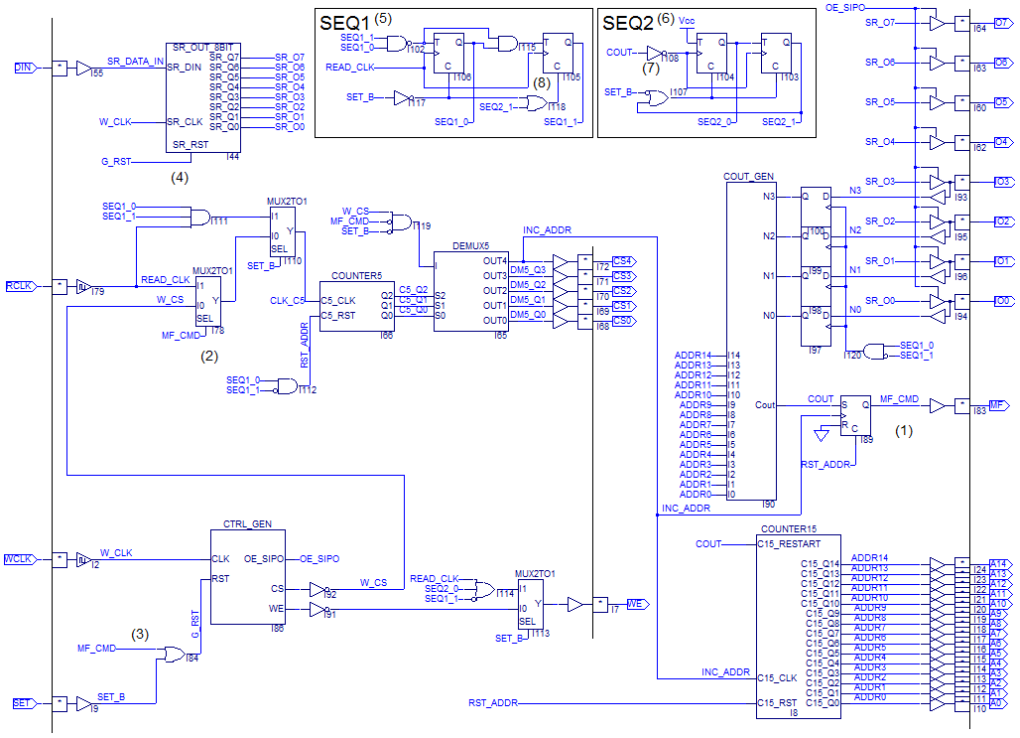


Fig. 4.91 *Schema circuitale della CPLD per la gestione del banco di memoria*

Le sezioni fondamentali del sistema CPLD per la gestione del banco di memoria sono:

- SR_OUT_8BIT: è un registro a scorrimento SIPO (*Serial Input Parallel Output*) con uscita a 8 *byte* (fig. 4.92). Tramite esso vengono creati i *byte* da memorizzare nelle diverse locazioni di memoria a partire dai *bit* seriali inviati dai convertitori DDC112.

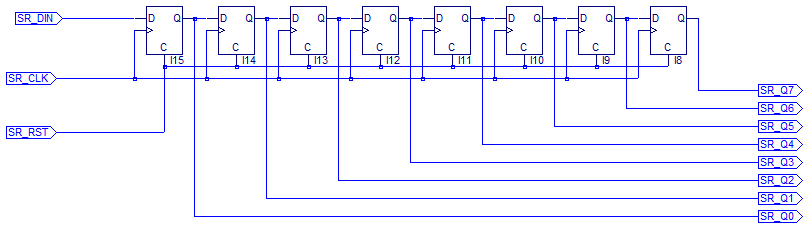


Fig. 4.92 *Schema del blocco SR_OUT_8BIT*

- COUNTER5, DEMUX5 e COUNTER15: sono un gruppo di blocchi progettati per formare e gestire gli indirizzi di memoria e i segnali di *chip select*. Il sistema memorizza le diverse parole pensandole come gruppi unici da 40 *bit* (cioè due canali acquisiti) memorizzandoli nei 5 *chip* W24257A allo stesso indirizzo di memoria. Si noti che COUNTER15 ha due linee di *reset*: C15_RST di tipo asincrono e C15_RESTART sincrono col *clock*. Viceversa COUNTER5 ha un solo *reset* di tipo asincrono. I codici ABEL relativi a questi blocchi sono riportati nel par. 4.6.4.

- CTRL_GEN: è un blocco di controllo utile alla generazione dei segnali di WE (*Write Enable*) e CS (*Chip Select*) durante l'operazione di scrittura seriale dei *bit* da parte dei convertitori. Il suo schema interno è riportato in fig. 4.93:

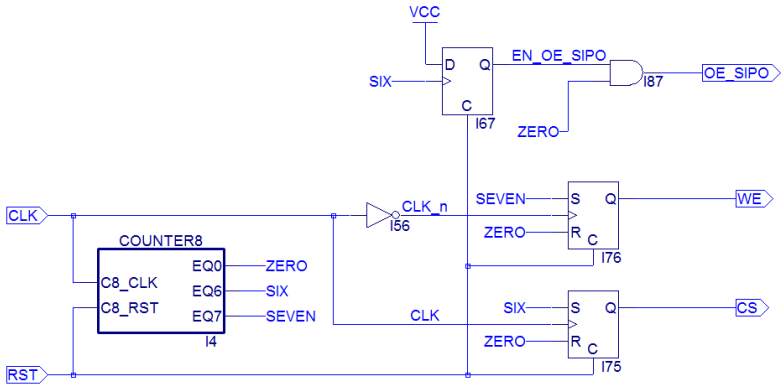


Fig. 4.93 *Schema del blocco CTRL_GEN*

- COUT_GEN: è un blocco che determina lo stato di memoria completamente riempita (genera il segnale MF, *Memory Full*) e quindi il termine dell'operazione di scrittura. E' possibile inoltre programmare un numero massimo di locazioni inferiore a quello effettivamente disponibile (2048 *frames*). Il blocco verrà illustrato descritto in dettaglio nel paragrafo 4.6.2.1.

- SEQ1 e SEQ2: sono blocchi progettati per la generazione dei segnali di controllo durante la fase di inizializzazione e *test* del banco di memoria.

4.6.1 Gestione delle fasi di scrittura e lettura

Un ingresso di controllo, denominato SET, determina le due possibili modalità di funzionamento del sistema. Quando SET = 0, siamo in condizioni di normale funzionamento, con il trasferimento al banco di memoria di ECSIC32 dei dati provenienti dai DDC112. In particolare si ha:

(i) la scrittura seriale per *bit* attraverso le linee WCLK e DIN;
(ii) la successiva lettura seriale per *byte* attraverso le linee RCLK e DATA BUS.

Nel caso in cui SET = 1, l'intero sistema viene invece resettato (avviene il *reset* del contatore degli indirizzi e del registro a scorrimento) ed inizializzato, ed eventualmente si può effettuare un *test* di funzionamento.

4.6.1.1 Fase di scrittura

Dopo l'inizializzazione dell'intero sistema (vedi par. 4.6.2), inizia la fase di scrittura seriale per *bit* dei dati provenienti dal blocco dei convertitori DDC112. La scrittura avviene sfruttando le sole linee WCLK e DIN: tramite WCLK vengono immagazzinati 8 *bit* (presenti sulla linea DIN) nel registro a scorrimento. I diversi *bit* dei canali verranno memorizzati a gruppi di 40 *bit* (2 canali) per ogni indirizzo sui 5 *chip* di memoria, ciascuno dei quali immagazzina un *byte*.
La fase di scrittura è caratterizzata dalle seguenti operazioni:

- Il segnale CS opportunamente negato (e indicato con W_CS, com'è possibile vedere nello schema completo della CPLD) viene posto in uscita dalla CPLD e diventa in maniera effettiva uno dei cinque *chip select*, a seconda della selezione su DEMUX5.
- Al fronte positivo di W_CS si commuta verso il successivo *chip select* con COUNTER5 (un contatore modulo 5 che riceve proprio W_CS come segnale di *clock*).
- Quando CS4, ultimo *chip select*, torna allo stato alto, si incrementa l'indirizzo: tutto è periodico con periodo 8x5 colpi di WCLK.

Il processo di scrittura ha termine quando il contatore degli indirizzi COUNTER15 raggiunge il valore massimo segnalato da COUT, che diventa 1 quando ADDR = 7FFF16: a questo punto viene attivato MF (*Memory Full*) se COUT è alto e arriva un ulteriore incremento al contatore degli indirizzi (cioè quando CS4 torna alto, indicando che è effettivamente terminata la scrittura dell'ultima locazione di memoria dell'ultimo *chip*). Il FF-D che genera MF è di tipo "*one-shot*": da questo momento il circuito viene inibito per la scrittura e può cominciare il processo di lettura, che avviene per *byte* per avere un *throughput* dei dati 8 volte superiore rispetto alla fase di scrittura.

4.6.1.2 Fase di lettura

Il processo di lettura avviene da parte del microcontrollore che acquisisce il contenuto del banco di memoria per gruppi di 8 *bit*: i diversi *byte* sono letti in modo "sequenziale" col segnale di *clock* RCLK. Con MF = 1 viene disabilitato il *buffer* SIPO, e viene generato il segnale W_RST, che tiene a 0 il blocco COUNTER8 interno a CTRL_GEN (e questo fa sì che WE e CS siano mantenuti alti). Inoltre, attraverso MUX2TO1, il *clock* del contatore dei *chip select* COUNTER5 diventa RLCK (il *clock* utilizzato per la lettura). L'ingresso I del DEMUX5 è sempre a 0 (questo vuol dire che almeno un *chip* deve essere selezionato). All'inizio del processo di lettura è già disponibile il primo *byte* memorizzato nella RAM, poiché è attivo CS0; gli altri *byte* sono accessibili in sequenza e scanditi col segnale RCLK.
Il ciclo di lettura può essere ripetuto quante volte si vuole, a meno che non si ponga SET = 1: una lettura ripetuta consente di effettuare ogni volta sui dati un'elaborazione diversa

(es. calcolo del valore atteso, calcolo dei valori massimo e minimo, calcolo del centroide, etc.).

4.6.2 Inizializzazione e *test* del banco di memoria

Si entra nella modalità di inizializzazione e *test* (eventuale) del banco di memoria ponendo SET = 1. Lo stato logico alto della linea di SET determina G_RST = 1: l'attivazione di questo segnale provoca il *reset* del registro a scorrimento SR_OUT_8BIT e del blocco CTRL_GEN, entrambi utilizzati durante il periodo di acquisizione seriale dei *bit* dalla linea DIN. L'inizializzazione del banco prosegue impostando il valore N_{max} di locazioni di memoria che si intendono effettivamente utilizzare, programmato da remoto (attraverso il microcontrollore)
Il segnale di *clock* durante l'inizializzazione e il *test* (sia nella fase di lettura di *test* che in quella di scrittura di *test*) è lo stesso *clock* di lettura usato in caso di normale funzionamento, ovvero RCLK.
Se SET viene mantenuto alto anche dopo la fine della fase di *reset* e inizializzazione, il microcontrollore effettuerà un *test* di funzionamento dei *chip* di memoria (senza aumentare il numero di *pin* di connessione):

- in un primo ciclo il μC potrà scrivere nella memoria un prefissato numero di *byte* noti;
- per verificare la corretta funzionalità delle locazioni di memoria, in un secondo ciclo di lettura il microcontrollore deve acquisire gli stessi *byte* che ha scritto nel precedente ciclo.

Avendo a disposizione il *data bus*, conviene leggere e scrivere per *byte*. I segnali di *chip select* (CS0, CS1, CS2, CS3 e CS4) e gli indirizzi sono gli stessi che si utilizzano per la scrittura seriale di *bit* presenti sulla DIN, gestita da WCLK.
Nel diagramma di fig. 4.94 sono evidenziate le diverse operazioni che avvengono in sequenza (gestite dai colpi di *clock* presenti sulla linea RCLK) in caso di SET = 1:

(i) avviene il *reset* del contatore degli indirizzi e del registro a scorrimento, e la scrittura di N_{max};
(ii) vengono generati N_{max} colpi di RCLK in cui il microcontrollore invia sul *data bus* i diversi *byte* che devono essere scritti nel banco di memoria;
(iii) vengono generati N_{max} colpi di RCLK in cui il microcontrollore riceve sul *data bus* i *byte* contenuti nel banco di memoria, scritti nella fase (ii).

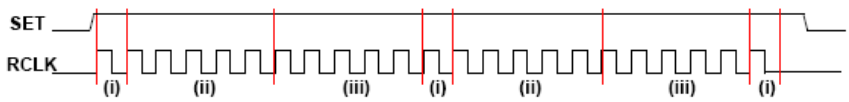


Fig. 4.94 *Sequenza di operazioni in caso di SET = 1*

I tre passi possono essere ripetuti per verificare la lettura/scrittura di altri *byte* corrispondenti a particolari configurazioni di *test*:

- Poiché si utilizza una logica TTL compatibile, la configurazione 00000000 corrisponde al caso in cui i *chip* di memoria sono effettivamente connessi, mentre la configurazione 11111111 è indicativa di una disconnessione.
- 10101010 e 01010101 sono configurazioni utilizzate per verificare la presenza di *bit* in cortocircuito.

Dopo la fase (iii) è necessario inserire un ulteriore colpo di RCLK per programmare il valore di N_{max} definitivo e resettare il sistema.
La linea SET controlla direttamente anche lo stato dei due *multiplexer* MUX2TO1 relativi all'uscita di WE (*Write Enable*) e all'ingresso di *clock* del contatore COUNTER5 usato per la generazione dei segnali di *chip select*. Si può notare che con SET = 1 le uscite di tali *multiplexer* dipendo dal segnale RCLK.
Inoltre, con SET = 1 viene eliminato il *reset* dei *sequencer* SEQ1 e SEQ2, che si occupano del controllo delle tre fasi enunciate in precedenza: (i) *reset*, (ii) *test* di scrittura e (iii) *test* di lettura.
Il primo *sequencer* SEQ1 è comandato proprio da RCLK, e servirà per contare i primi colpi di *clock* affinché il sistema entri nelle fasi (ii) e (iii) di *test*.
Il secondo *sequencer* SEQ2 è comandato dal segnale COUT, che viene generato alla fine di ciascuna operazione di scrittura/lettura quando gli indirizzi hanno raggiunto il valore massimo programmato: poiché l'attivazione di COUT si ha quando è raggiunto l'ultimo indirizzo e lo stato logico alto permane finché sono generati i 5 segnali di *chip select*, l'incremento del contatore SEQ2 dovrà avvenire sul fronte negativo di COUT, cioè alla fine dell'ultimo ciclo di scrittura o lettura.
Nel diagramma di fig. 4.95 è riportato il dettaglio dell'avvio delle operazioni di *reset* e avvio del *test* di scrittura. I numeri riportati in ciascun periodo di *clock* si riferiscono al valore assunto dal contatore SEQ1.

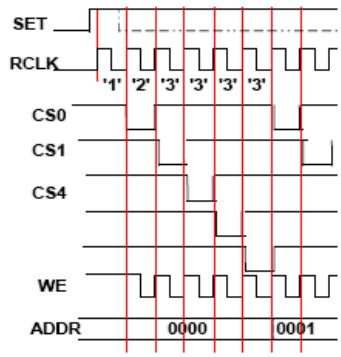


Fig. 4.95 *Diagramma temporale delle fasi di reset e test di scrittura*

4.6.2.1 Periodo di reset (i)

Dopo il primo fronte di RCLK l'uscita al valore 1 del contatore SEQ1 serve per inizializzare il sistema e per caricare in un registro a 4 *bit* il valore N_{max} presente sul *data bus*: l'inizializzazione consiste nel generare il segnale RST_ADDR, che resetta i contatori COUNTER15 degli indirizzi e COUNTER5 dei *chip select*. Viene inoltre

azzerato il contenuto del *flip-flop* D che controlla la linea MF (*Memory Full*). La condizione SEQ1 = 1 determina anche la generazione di un fronte di *clock* sui 4 *flip-flop* D in ingresso a COUT_GEN, che memorizzano lo stato delle 4 linee meno significative del *data bus*.
Il valore N_{max} che viene caricato corrisponde a quante linee di indirizzo (delle 15 disponibili), vogliamo "eliminare" tra le più significative. Il valore registrato può andare da un minimo di 0 (indirizzo a 15 *bit*) a un massimo di 11 (indirizzo a 4 *bit*): in questo modo il numero di acquisizioni potrà andare da un minimo di 640 *bit* (per N_{max} = 11 si hanno solo 4 linee di indirizzo, cioè 16 locazioni di memoria, ovvero 16 x 8 *bit* x 5 *chip* = 640 *bit*), a un massimo di 1310720 *bit* (5 *chip* interi da 32kB). Si noti che se N_{max} = 11, sarà possibile memorizzare un singolo *frame* (quadro) da 32 canali.
Se la linea SET viene abbassata dopo che RCLK ritorna basso, come indicato con la linea tratteggiata nel diagramma temporale della fig. 4.95, non si entra nella modalità di *test*: ciò vuol significare che è stato solamente inizializzato il sistema. I colpi di RCLK con SET = 0 non hanno alcun effetto, poiché la lettura della memoria durante il funzionamento normale si ha solo se la linea di comando MF_CMD è attivata, e questo avviene solo dopo che i DDC112 hanno completato il riempimento di N_{max} locazioni di memoria. Per evitare ogni confusione, se SET = 0, anche RCLK viene posto a 0 dal microcontrollore.
Nella fig. 4.96 è riportato lo schema del registro che memorizza il valore N_{max}.

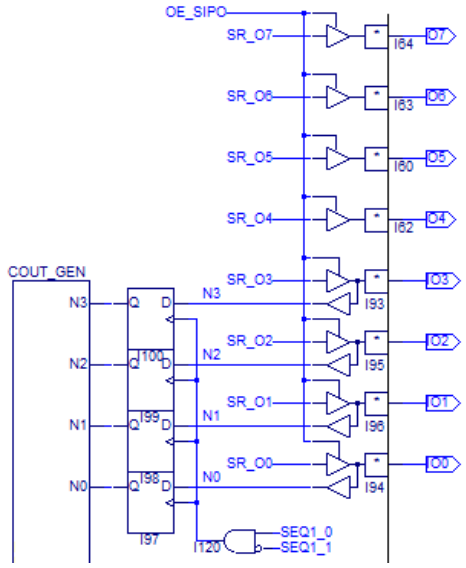
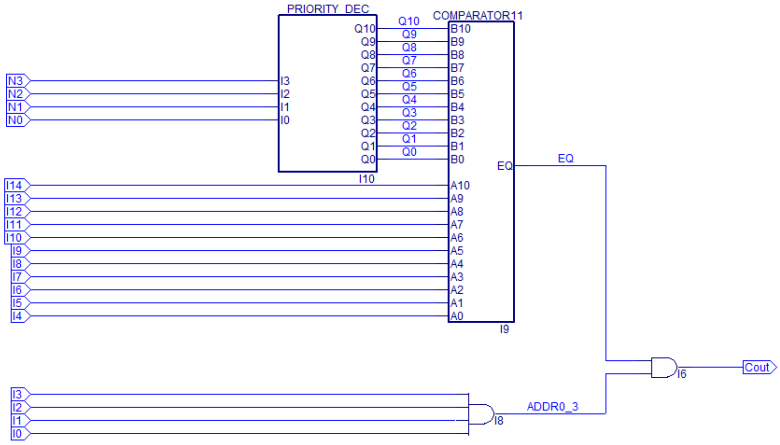


Fig. 4.96 *Schema circuitale per la registrazione del valore* N_{max}

Essendo necessari solo 4 *bit*, le linee IO (da 0 a 3) del *data bus* sono di tipo bidirezionale, per poter caricare nel registro a 4 *bit* I97 il valore N_{max} programmato. Come detto, il caricamento del dato N_{max} avviene quando SEQ1 raggiunge il valore 1, ovvero dopo il primo fronte di RCLK (col ritardo introdotto dal contatore SEQ1). Se viceversa SET = 0, SEQ1 viene resettato e quindi la porta AND I120 andrà a 0.

Il blocco COUT_GEN riceve N_{max} dalle linee N (da 0 a 3) di ingresso collegate in uscita al registro a 4 *bit*.
In fig. 4.97 è riportato lo schema interno del blocco COUT_GEN: le linee N (da 0 a 3) vanno a un *encoder* a priorità la cui tabella della verità è riportata sotto lo schema.




```
[I3,I2,I1,I0] -> [Q10,Q9,Q8,Q7,Q6,Q5,Q4,Q3,Q2,Q1,Q0]
[ 0, 0, 0, 0] -> [ 1 ,1 ,1 ,1 ,1 ,1 ,1 ,1 ,1 ,1 ,1 ];
[ 0, 0, 0, 1] -> [ 0 ,1 ,1 ,1 ,1 ,1 ,1 ,1 ,1 ,1 ,1 ];
[ 0, 0, 1, 0] -> [ 0 ,0 ,1 ,1 ,1 ,1 ,1 ,1 ,1 ,1 ,1 ];
[ 0, 0, 1, 1] -> [ 0 ,0 ,0 ,1 ,1 ,1 ,1 ,1 ,1 ,1 ,1 ];
[ 0, 1, 0, 0] -> [ 0 ,0 ,0 ,0 ,1 ,1 ,1 ,1 ,1 ,1 ,1 ];
[ 0, 1, 0, 1] -> [ 0 ,0 ,0 ,0 ,0 ,1 ,1 ,1 ,1 ,1 ,1 ];
[ 0, 1, 1, 0] -> [ 0 ,0 ,0 ,0 ,0 ,0 ,1 ,1 ,1 ,1 ,1 ];
[ 0, 1, 1, 1] -> [ 0 ,0 ,0 ,0 ,0 ,0 ,0 ,1 ,1 ,1 ,1 ];
[ 1, 0, 0, 0] -> [ 0 ,0 ,0 ,0 ,0 ,0 ,0 ,0 ,1 ,1 ,1 ];
[ 1, 0, 0, 1] -> [ 0 ,0 ,0 ,0 ,0 ,0 ,0 ,0 ,0 ,1 ,1 ];
[ 1, 0, 1, 0] -> [ 0 ,0 ,0 ,0 ,0 ,0 ,0 ,0 ,0 ,0 ,1 ];
[ 1, 0, 1, 1] -> [ 0 ,0 ,0 ,0 ,0 ,0 ,0 ,0 ,0 ,0 ,0 ];
```

Fig. 4.97 *Schema circuitale e tavola della verità dell'encoder a priorità*

Come detto in precedenza, il blocco COUT_GEN genera il segnale COUT, che segnala lo stato di memoria piena: tale stato si ottiene attraverso un confronto (fatto dal blocco COMPARATOR11, che è un classico comparatore a 11 *bit*, e dalla porta AND a 4 ingressi) tra il valore attuale di indirizzo e quello impostato.
In pratica, se N_{max} = 0, l'*encoder* genera 11 *bit* pari a 1 e si ha COUT = 1 quando viene raggiunto l'indirizzo $2^{15} - 1$, sfruttando così l'intera capacità del banco di memoria. Se N_{max} = 1, il confronto è fatto col valore 01111111111 uscente dall'*encoder* a priorità: in questo caso, COUT è generato quando è stata riempita metà della capacità del banco di memoria (se N_{max} = 2 si riempirà 1/4 della capacità totale, se N_{max} = 3 se ne riempirà 1/8, e così via fino a N_{max} = 11). In definitiva, il numero scritto dal microcontrollore (compreso tra 0 e 11) determina quanti *bit* in meno delle linee di indirizzo A14..0 di ciascun *chip* di memoria vengono effettivamente usati.

N.B. Dalla fig. 4.91 (schema completo) si può osservare come COUT divenga effettivamente la condizione *Memory Full* (MF = 1) solo quando giunge un fronte positivo di INC_ADDR: questo fronte è generato solo al termine della scrittura del quinto *chip* di memoria, essendo corrispondente al segnale (attivo basso) CS4.

4.6.2.2 Periodo di scrittura (ii)

Se SET è mantenuto al livello logico alto alla fine della fase di *reset* e inizializzazione, gli ulteriori colpi di RCLK avviano il *test* della memoria con una prima fase di scrittura in corrispondenza dei fronti positivi di RCLK: il dato memorizzato è il *byte* che il microcontrollore pone sul *data bus*. In realtà, il *byte* viene scritto nella locazione di memoria quando giunge il fronte di *clock* che segna l'inizio del ciclo successivo, questo perché il ciclo di scrittura termina quando la linea WE torna alta.

Le fasi per l'avvio del *test* devono essere due. Quando SEQ1 = 2, bisogna mantenere a 0 i contatori di *chip select* e degli indirizzi: se così non fosse, il secondo RCLK inviato potrebbe provocare un incremento in COUNTER5, attivando CS1 anziché CS0. L'incremento invece deve avvenire al *clock* successivo e cioè quando SEQ1 = 3: a questo punto, giunta a regime la fase di *test* in scrittura, il contatore può bloccarsi rimanendo a 3, in attesa che giunga il primo COUT a segnalare che si è giunti all'ultimo indirizzo utile.

Il *clock* che giunge ai contatori di *chip select* e degli indirizzi è lo stesso RCLK del caso in cui SET = 1: per tale operazione, è stato inserito il MUX I110, comandato dal segnale SET stesso. La porta AND I111 fa sì che il *clock* per l'avanzamento (RCLK) giunga solo quando SEQ1 = 3.

Lo schema della sezione qui coinvolta è riportato in fig. 4.98 (non è compreso il contatore degli indirizzi COUNTER 15, il cui *clock* è INC_ADDR, come detto precedentemente).

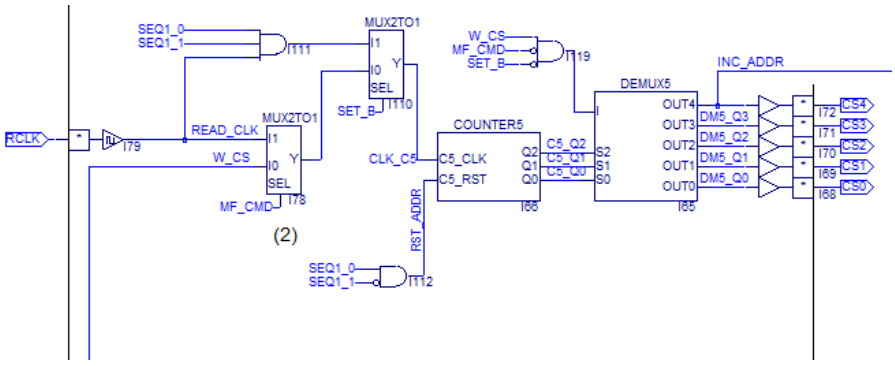


Fig. 4.98 *Parte della CPLD relativa alla gestione del periodo di scrittura*

L'ingresso di DEMUX5, che determina lo stato effettivo della linea CS selezionata, è sicuramente allo stato logico basso, poiché l'ingresso SET_B della AND I119 è pari a 1 (SET_B non è altro che la linea SET dopo il *buffer* di ingresso al *pin* della CPLD). Gli altri due ingressi, W_CS e MF_CMD, influenzano il valore di ingresso di DEMUX5 solo durante il periodo di scrittura seriale per *bit* dalla linea DIN. Nella fase di *test* la scrittura avviene invece per *byte*, e lo stato logico basso dell'ingresso di DEMUX indica che la linea CS selezionata è attiva per un intero periodo di RCLK (e non ogni 8 colpi di *clock* come si ha nel caso di scrittura seriale per *bit* dalla linea DIN).

Durante il periodo in cui una linea CS è bassa, occorre generare il segnale WE di *Write Enable*, che potrà giungere a partire dal periodo in cui SEQ1 raggiunge il valore 2. WE

potrà coincidere con RCLK: infatti CS torna alto dopo che RCLK è tornato alto, per cui non si avrà mai la deselezione del singolo *chip* con WE ancora basso.
Durante la scrittura, WE dovrà essere generato continuamente finché non si giunge all'istante in cui è generato il primo COUT. Il numero di COUT viene contato dal secondo contatore SEQ2, che conterà solo 0, 1 e 2: infatti, giunti al secondo COUT, tutto deve ricominciare a partire dalla fase (i) di inizializzazione. Come indicato in fig. 4.99, per implementare quanto detto, si fa utilizzo di un MUX a 2 ingressi e 1 segnale di controllo (SET). Supponiamo che SET = 1, quindi siamo in fase di inizializzazione o *test*: se SEQ1 = 2 o 3 (quindi il *bit* SEQ1_1 è alto) e se SEQ2_0 è basso (quindi non è ancora arrivato il COUT), si fa giungere a WE il *clock* RCLK. Poiché negli altri casi la linea WE deve stare sempre a 1, si usa una OR come porta di trasmissione di RCLK, e i comandi SEQ2_0 devono agire in logica negata.

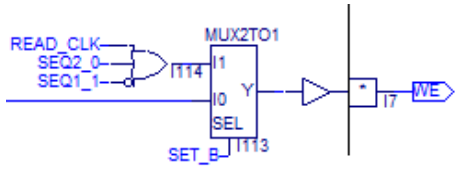


Fig. 4.99 *Generazione del Write Enable*

N.B. La commutazione di WE in anticipo rispetto alla commutazione di CS è garantita dal fatto che il ritardo su WE è dovuto solo alla porta OR e al MUX, mentre per i *chip select* CS si ha un ritardo (sicuramente superiore a quello su WE) precedente dovuto alla successione porta AND + MUX + COUNTER5 + DEMUX5.

4.6.2.3 Periodo di lettura (iii)

Il ciclo di scrittura relativo alla fase (ii) termina quando COUT commuta da alto a basso. Il contatore SEQ2 presenterà allora il valore 1 (SEQ2_0 = 1 e SEQ2_1 = 0): questo mantiene alta la linea WE, e la memoria non sarà più scritta. Il microcontrollore a questo punto dovrà attivare una alla volta le linee OE (*Output Enable*) di ciascun banco di memoria a cui è collegato per rileggere quanto ha precedentemente scritto.
Giunto il COUT, per la presenza dell'ingresso di RESTART al contatore indirizzi COUNTER15, la memoria sarà nuovamente indirizzata, a partire dall'indirizzo iniziale 0, non appena giunge un nuovo colpo di RCLK. Il *clock* RCLK deve arrivare ai contatori di indirizzo e di CS in modo che la memoria sia scansionata così come avveniva nella fase di scrittura: questo è garantito dal fatto che SEQ1 rimane "inchiodato" al valore 3.
Il processo di lettura deve terminare quando si ha il secondo fronte negativo di COUT. Quando questo arriva, SEQ2 passa al valore 2: la linea SEQ2_1 va a 1, e viene usata per azzerare i due contatori di controllo SEQ1 e SEQ2. Mentre SEQ2 viene effettivamente resettato al valore 0, lo stesso non deve avvenire per SEQ1: infatti si vuole che il ciclo ricominci come nella fase (i) e cioè con SEQ1 al valore 1. Poiché SEQ1 si trovava a 3, va resettato solo il *bit* SEQ1_1 mentre SEQ1_0 può rimanere a 1.
Lo schema dei due contatori di controllo è riportato in fig. 4.100. Si può notare che SEQ2 ha il *clock* pari a COUT negato, in modo da commutare al termine dell'operazione di lettura o scrittura dell'ultima locazione di memoria. Quando il *bit* più significativo SEQ2_1 diventa alto, SEQ2 viene resettato dalla OR I107. Lo stesso

comando arriva solo al *flip-flop* T più significativo (I105) del contatore SEQ1, consentendo il passaggio dello stato di SEQ1 da 3 a 1.
I due contatori sono invece completamente azzerati quando la linea SET è allo stato logico basso e cioè si è nella condizione di funzionamento normale di trasferimento dati sulla linea DIN.

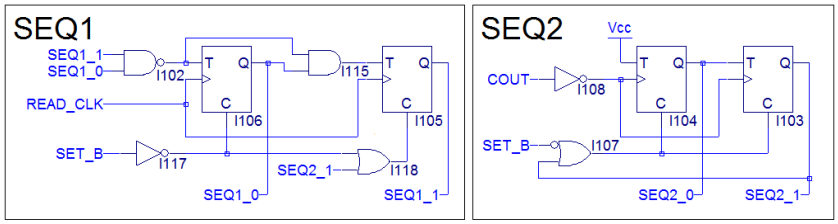


Fig. 4.100 *Schema circuitale dei due sequencer*

4.6.3 Verifica di funzionamento della CPLD per la gestione del banco di memoria

In questo paragrafo vengono riportati e commentati alcuni diagrammi temporali di simulazione (*timing simulation*) ottenuti con ispLEVER Project Navigator. Per effettuare la simulazione in tempi ragionevoli, il numero di *bit* del contatore degli indirizzi (che è il contatore COUNTER15 a 15 *bit*) è stato ridotto a 3. Le simulazioni riguardano diverse situazioni di funzionamento:

- fase di scrittura dei dati provenienti dal banco dei DDC112 (SET = 0);
- fase di lettura dei dati scritti in precedenza e provenienti dal banco dei DDC112;
- fase di inizializzazione e *test* (SET = 1).

4.6.3.1 Simulazione della fase di scrittura

Dall'analisi della simulazione temporale (fig. 4.101), si può notare come i diversi impulsi WE (*Write Enable*) siano attivati solo dopo l'attivazione del singolo CS (*Chip Select*), e siano disattivati prima della sua disattivazione. Si può notare inoltre come i diversi *chip* si attivino solo quando è stata completata la scrittura di 1 *byte* (8 colpi del *clock* di scrittura WCLK) nel *chip* precedente.

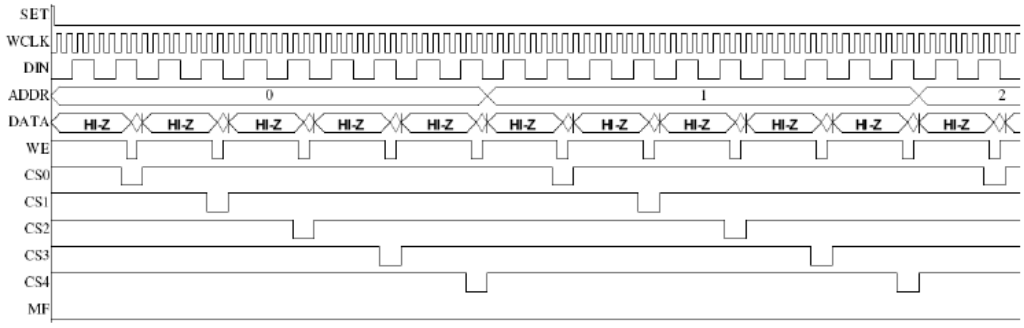


Fig. 4.101 *Simulazione della fase di scrittura*

Infine è evidente come il cambio d'indirizzo si abbia soltanto dopo l'attivazione (o disattivazione) dell'ultimo *chip* di memoria (CS4).

4.6.3.2 Simulazione della fase di lettura

La fase di lettura inizia correttamente quando è stata completata la fase di scrittura (è stato generato il segnale di *Memory Full*), e il segnale WE rimane sempre alto durante tutta la fase di lettura (fig. 4.102).
E' importante notare che il *chip* 0 è disponibile alla lettura (CS0 = 0) non appena completata la fase di scrittura sull'intero banco di memoria, ancor prima dell'invio del *clock* di lettura (RCLK). Si vede inoltre che, a differenza della fase di scrittura, i segnali di *chip select* non scalano ogni 8 colpi di *clock*, ma ad ogni colpo di *clock* (la lettura avviene infatti per *byte*). Come nella fase di scrittura, infine, anche in fase di lettura l'indirizzo (ADDR) è incrementato solo alla deselezione dell'ultimo *chip*.
Alla fine delle operazioni di lettura, si è deciso di resettare e re-inizializzare il sistema, ponendo SET = 1: in caso contrario, con SET mantenuto a 0, il processo di lettura si sarebbe ripetuto più volte.

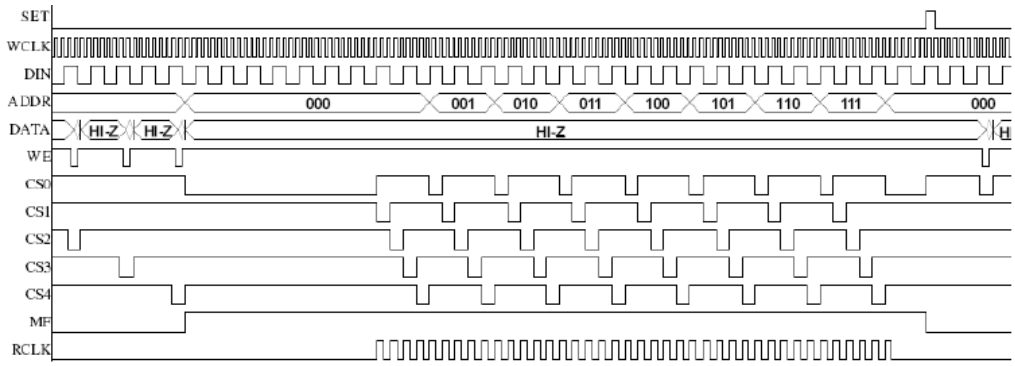


Fig. 4.102 *Simulazione della fase di lettura*

4.6.3.3 Simulazione delle fasi di inizializzazione e test

Il diagramma di fig. 4.103 si riferisce alle operazioni di inizializzazione (scrittura del valore N_{max} di locazioni di memoria da utilizzare) e scrittura di *test*. Si è scelto un N_{max} pari a 11 (configurazione 1011 presente sulle linee IO3, IO2, IO1 e IO0) in modo che siano riempite solo 16 locazioni di memoria di ogni *chip*, quindi si avranno 16 x 5 = 80 colpi di RCLK per ogni fase: si ricorda che RCLK viene usato nelle operazioni di *test* sia come *clock* di scrittura che come *clock* di lettura.
E' stato inoltre considerato il caso in cui le linee WCLK e DIN siano comunque attivate, perché i DDC112 continuano in ogni caso a funzionare (per la presenza dei MUX sulle linee, ad ogni modo, lo stato di queste ultime non influenzerà il funzionamento del circuito).
Si osserva che dopo 16 x 5 colpi di RCLK le linee di indirizzo ritornano effettivamente tutte a zero. Si può anche osservare che quando ADDR raggiunge il valore N_{max} pari a 0x000F viene generato il primo COUT: non appena COUT torna basso, viene attivata la linea MF (*Memory Full*), che nella modalità di *test* non ha alcuna influenza. Infine si

osservi che, dopo l'arrivo del segnale COUT, la linea WE (*Write Enable*) è posta al livello logico alto, indicando che il processo di scrittura ha avuto termine.

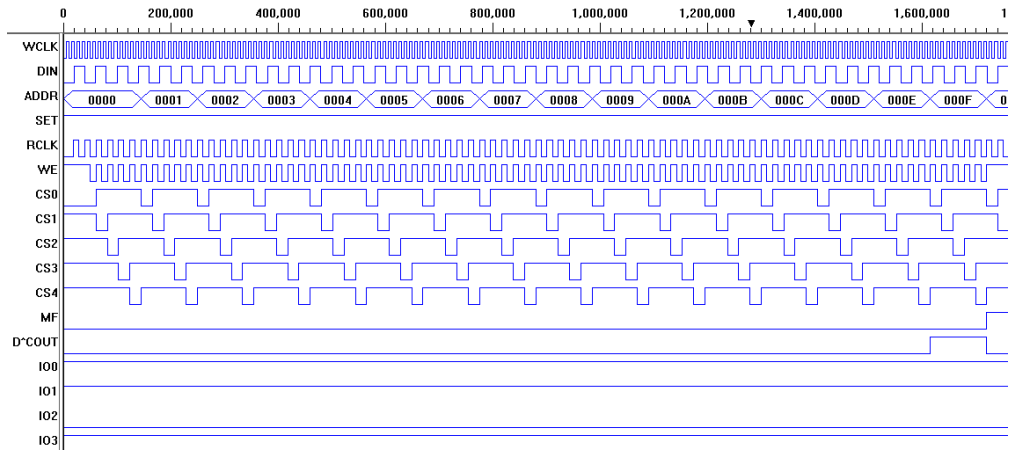


Fig. 4.103 *Simulazione della fase di inizializzazione e scrittura di test*

Nel diagramma di fig. 4.104 è riportato l'andamento dei segnali durante la fase di lettura dei *byte* che il microcontrollore ha precedentemente scritto in tutti i banchi di memoria (lettura di *test*). Il microcontrollore potrà rileggere il contenuto dei singoli *chip* dei diversi banchi attivando una alla volta le linee OE (*Output Enable*) di ciascun banco. Si osserva che ora la linea WE, ovviamente, rimane allo stato logico alto. Inoltre, finché non giungono gli 80 nuovi impulsi di RCLK, la linea di *Memory Full* rimane comunque alta, indicando che si sta effettivamente eseguendo un accesso in lettura della memoria.
La simulazione evidenzia anche l'avvio di un nuovo ciclo di inizializzazione e *test*: si osservi che il fronte positivo di RCLK che si ha dopo la conclusione della fase di lettura di *test* (cerchiato in figura) è quello necessario a caricare il valore N_{max} del nuovo ciclo.

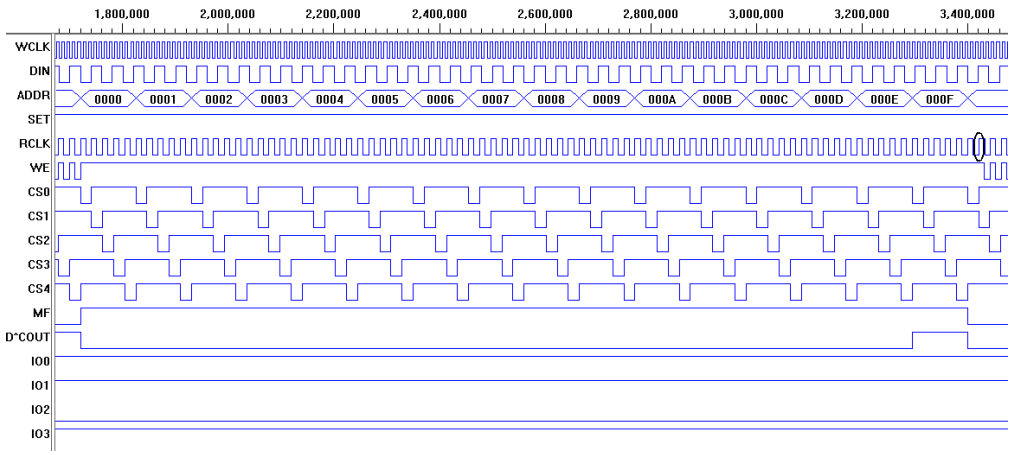


Fig. 4.104 *Simulazione della fase di lettura di test*

La simulazione riportata in fig. 4.105 si riferisce alla fine della fase di *test*. E'stato anche simulato l'inizio della scrittura dati dalla linea DIN del banco dei DDC112 (vedi par. 4.6.3.1). Come detto per la simulazione precedente di lettura di *test*, alla fine della lettura dal banco di memoria l'ulteriore colpo di RCLK serve per caricare il valore N_{max} scelto per le operazioni successive.

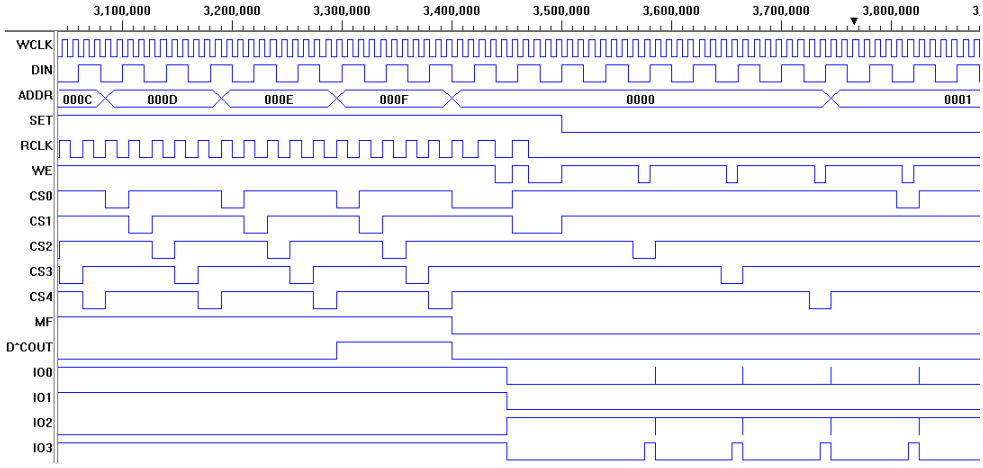


Fig. 4.105 *Simulazione della fase conclusiva del test*

4.6.4 Codici ABEL dei blocchi progettati

Vengono qui di seguito riportati i codici ABEL relativi ai blocchi funzionali presenti nella CPLD di gestione del banco di memoria:

1)

```
MODULE COUNTER15

TITLE '15 BIT COUNTER'

"inputs"
      C15_CLK             pin;   "clock input"
      C15_RST             pin;   "asynchronous reset"
      C15_RESTART         pin;   "RESTART INPUT"

"outputs"
C15_Q0 , C15_Q1 , C15_Q2, C15_Q3, C15_Q4      pin istype 'reg';
C15_Q5 , C15_Q6 , C15_Q7, C15_Q8, C15_Q9      pin istype 'reg';
C15_Q10, C15_Q11, C15_Q12, C15_Q13, C15_Q14   pin istype 'reg';

"sets"
      count=[C15_Q14..C15_Q0];

equations
      count.clk = C15_CLK;
      count.ar = C15_RST;
      when (C15_RESTART) then count:=0 else count:=count+1;

END
```

2)

```
MODULE COUNTER5

TITLE 'MOD-5 COUNTER'

"inputs"
      C5_CLK              pin;   "clock input"
      C5_RST              pin;   "asynchronous reset"

"outputs"
      Q0 , Q1 , Q2        pin istype 'reg';

"sets"
      count=[Q2..Q0];

equations

      count.clk = C5_CLK;
      count.ar = C5_RST;
      when (count==4) then count := 0 else count:=count+1;

END
```

3)

```
MODULE DEMUX5

TITLE '5 OUTPUT DEMULTIPLEXER'

"inputs"
      I                                  pin;   "input signal"
      S2, S1, S0                         pin;   "selection"

"outputs"
      OUT4, OUT3, OUT2, OUT1, OUT0     pin istype 'com';


"sets"
      SEL = [S2..S0];

equations

when (SEL == 0) then [OUT4,OUT3,OUT2,OUT1,OUT0]=[1,1,1,1,I];
when (SEL == 1) then [OUT4,OUT3,OUT2,OUT1,OUT0]=[1,1,1,I,1];
when (SEL == 2) then [OUT4,OUT3,OUT2,OUT1,OUT0]=[1,1,I,1,1];
when (SEL == 3) then [OUT4,OUT3,OUT2,OUT1,OUT0]=[1,I,1,1,1];
when (SEL == 4) then [OUT4,OUT3,OUT2,OUT1,OUT0]=[I,1,1,1,1];

END
```

4)

```
MODULE PRIORITY_DEC

TITLE 'PRIORITY DECODER'

"inputs"
I3, I2, I1, I0                          pin;   "bynary data"

"outputs"
Q0,Q1,Q2,Q3,Q4,Q5,Q6,Q7,Q8,Q9,Q10       pin istype 'com';

equations

truth_table ( [I3,I2,I1,I0]-> [Q10,Q9,Q8,Q7,Q6,Q5,Q4,Q3,Q2,Q1,Q0])
              [ 0, 0, 0, 0]-> [ 1 ,1 ,1 ,1 ,1 ,1 ,1 ,1 ,1 ,1 ];
              [ 0, 0, 0, 1]-> [ 0 ,1 ,1 ,1 ,1 ,1 ,1 ,1 ,1 ,1 ];
              [ 0, 0, 1, 0]-> [ 0 ,0 ,1 ,1 ,1 ,1 ,1 ,1 ,1 ,1 ];
              [ 0, 0, 1, 1]-> [ 0 ,0 ,0 ,1 ,1 ,1 ,1 ,1 ,1 ,1 ];
              [ 0, 1, 0, 0]-> [ 0 ,0 ,0 ,0 ,1 ,1 ,1 ,1 ,1 ,1 ];
              [ 0, 1, 0, 1]-> [ 0 ,0 ,0 ,0 ,0 ,1 ,1 ,1 ,1 ,1 ];
              [ 0, 1, 1, 0]-> [ 0 ,0 ,0 ,0 ,0 ,0 ,1 ,1 ,1 ,1 ];
              [ 0, 1, 1, 1]-> [ 0 ,0 ,0 ,0 ,0 ,0 ,0 ,1 ,1 ,1 ];
              [ 1, 0, 0, 0]-> [ 0 ,0 ,0 ,0 ,0 ,0 ,0 ,0 ,1 ,1 ];
              [ 1, 0, 0, 1]-> [ 0 ,0 ,0 ,0 ,0 ,0 ,0 ,0 ,0 ,1 ];
              [ 1, 0, 1, 0]-> [ 0 ,0 ,0 ,0 ,0 ,0 ,0 ,0 ,0 ,1 ];
              [ 1, 0, 1, 1]-> [ 0 ,0 ,0 ,0 ,0 ,0 ,0 ,0 ,0 ,0 ];

END
```

4.7 Interfacciamento per tastiera a matrice

La tastiera esterna alla *main board* rappresenta la periferica d'ingresso con la quale l'utente può fornire dati o comandi relativi ad una sessione di misura. Essendo il numero dei tasti relativamente elevato (i dieci pulsanti per le cifre decimali e un paio di uso speciale) l'organizzazione è di tipo matriciale: il singolo pulsante premuto cortocircuita la riga e la colonna a cui è collegato.
La tastiera può essere interfacciata con il microcontrollore adottando due soluzioni: un interfacciamento analogico, che consente di risparmiare sul numero di connessioni (*pin*) col μC, oppure un interfacciamento digitale, più semplice ma maggiormente oneroso dal punto di vista dell'occupazione dei *pin* del μC. La soluzione adottata per ECSIC32 è quella analogica, in quanto il numero di *pin* disponibili dell'LPC2103 è limitato dalla presenza delle connessioni con il banco dei DDC112: è stata lasciata però, in fase di realizzazione della *main board*, la possibilità di utilizzare un interfacciamento digitale per applicazioni future che prevedano l'utilizzo di componenti alternativi al DDC112, e che facciano uso di un numero minore di connessioni.
Nei successivi paragrafi verranno illustrati entrambi i tipi di interfacciamento.

4.7.1 Interfacciamento analogico

Poiché la maggior parte dei μC, come l'LPC2103, integra un ADC, nasce l'idea di poter interfacciare diversi tasti utilizzando solo la linea di ingresso analogica, risparmiando: con questo approccio, lo stato di un tasto dovrà determinare un particolare fattore di partizione di una tensione di riferimento. Se questa poi coincide con quella dell'ADC stesso, si eliminano molti dei problemi generalmente legati alla "qualità" del riferimento di tensione (per l'LPC2103 la tensione di riferimento del convertitore è pari a quella di alimentazione analogica V_{DDA}).
In fig. 4.106a è indicata quest'idea nel caso di quattro pulsanti: a seconda del pulsante premuto, cambia il fattore di partizione di V_{DDA}. Se i resistori hanno tutti lo stesso valore, otterremo i fattori 1, 1/2, 1/3 e 1/4, che porteranno a valori di V_{OUT} ben distinguibili in uscita, anche considerando la tolleranza sul valore di ogni resistore. È degno di nota che l'assorbimento di corrente da parte della tastiera si ha solo quando viene premuto un tasto: quindi, in condizioni di non utilizzo, non avendo alcuna dissipazione di potenza, questo tipo di circuito ben si adatta a possibili applicazioni portatili con alimentazione a batteria.
Lo schema con partitore controllato dai tasti disposti su riga singola può estendersi a più righe inserendo un nuovo resistore tra due righe adiacenti: si giunge allora al sistema riportato in fig. 4.106b, relativo a una matrice di tasti 4 x 4 (come quella utilizzata per ECSIC32), in cui valgono le stesse considerazioni di assorbimento nullo in condizioni di riposo.

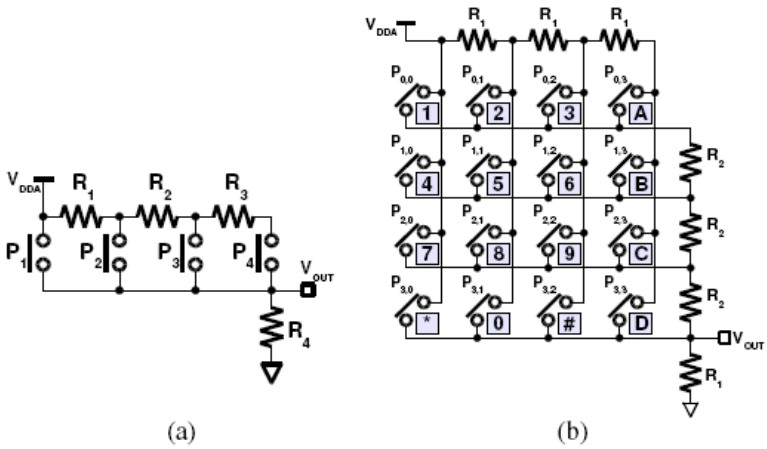


Fig. 4.106 *Schema base di tastiera a 4 pulsanti (a) e matrice di tasti 4x4 per il sistema ECSIC32 (b)*

La struttura è semplificata facendo in modo che il denominatore della frazione che esprime il fattore di partizione cambi, con il tasto premuto, seguendo una successione di numeri interi: a tal fine, i resistori R_2 indicati per le "nuove" righe dovranno avere un valore di resistenza almeno 4 volte superiore agli altri che avevamo già nel caso (a). La tensione V_{OUT} di uscita è funzione del tasto premuto $P_{n,m}$ (n,m = 0..3) secondo la relazione:

$$V_{OUT}(P_{n,m}) = \frac{V_{DDA}}{p(3-n)+m+1} \qquad \text{[IV.3]}$$

dove $p = R_2 / R_1$.
Essendo $p = 4$, otteniamo che i 16 possibili valori di tensione variano tra 1/16 ($P_{0,3}$ premuto) e 1 ($P_{3,0}$ premuto). Il denominatore della frazione di partizione cambia di un'unità tra un pulsante e quello adiacente. La differenza minima tra valori adiacenti della V_{OUT} si ha per i tasti $P_{0,2}$ e $P_{0,3}$ e vale $15^{-1} – 16^{-1} = 240^{-1}$. Poiché il convertitore A/D a 10 *bit* è in grado di risolvere 1/1024, sarà possibile una sua applicazione in questo tipo di interfacciamento.
Nella tabella 4.3, oltre al simbolo relativo al tasto, sono riportati i diversi fattori di partizione che si hanno in funzione del particolare pulsante premuto. Nell'ultima riga sono indicati i codici che idealmente il convertitore presenterà nei vari casi.

$P_{0,0}$	$P_{0,1}$	$P_{0,2}$	$P_{0,3}$	$P_{1,0}$	$P_{1,1}$	$P_{1,2}$	$P_{1,3}$	$P_{2,0}$	$P_{2,1}$	$P_{2,2}$	$P_{2,3}$	$P_{3,0}$	$P_{3,1}$	$P_{3,2}$	$P_{3,3}$
'1'	'2'	'3'	'A'	'4'	'5'	'6'	'B'	'7'	'8'	'9'	'C'	'*'	'0'	'#'	'D'
1/13	1/14	1/15	1/16	1/9	1/10	1/11	1/12	1/5	1/6	1/7	1/8	1	1/2	1/3	1/4
78	**73**	**68**	**64**	**113**	**102**	**93**	**85**	**204**	**170**	**146**	**128**	**1023**	**512**	**341**	**256**

Tab. 4.3 *Fattori di partizione in funzione del tasto premuto*

4.7.1.1 Codice di gestione della tastiera a matrice

Lalgoritmo che consente di rivelare quale pulsante sia stato premuto verifica se il codice presentato dall'ADC è superiore o inferiore a valori di soglia prefissati. E' naturale scegliere le soglie quali *valori intermedi* tra quelli scritti nella tabella di cui sopra: per esempio, se l'ADC fornisce un valore maggiore di 137 ma minore di 158, siamo certi che il tasto premuto è quello corrispondente a "9".
Per la tolleranza T sul valore dei singoli resistori e l'errore di *offset* dell'ADC, i codici di uscita non sono esattamente quelli indicati in tabella. L'errore che si può commettere deve essere inferiore a quello che fa confondere la decodifica di due tasti adiacenti (aventi cioè i fattori di partizione più prossimi): il caso peggiore si ha per i tasti di posizione $P_{0,2}$ e $P_{0,3}$ ("3" e "A") che, come già notato, presentano i fattori di partizione più vicini. In particolare per $P_{0,3}$ (trascurando l'errore di *offset* del convertitore) deve essere:

$$\frac{1024}{1+15\left(\frac{1-T}{1+T}\right)} < 66 \qquad \text{[IV.4]}$$

avendo considerato peggiore il caso in cui il resistore di uscita assuma il valore più alto $R_1(1+T)$, mentre tutti gli altri quello minimo, dato dal contributo totale: $3R_1(1-T)$ +3 $R_2(1-T)$. Dall'espressione precedente si ricava che la tolleranza sui resistori dev'essere inferiore a 0.0164, e che quindi resistori con tolleranza del ±1% sono idonei a garantire la funzionalità del sistema.
Per i fattori di partizione adiacenti che danno luogo a codifiche relativamente distanti, è possibile anche riconoscere casi in cui siano premuti due o più tasti contemporaneamente.

L'algoritmo per la decodifica del tasto premuto prevede una serie di *if* in cascata. Nel listato che segue è riportato il *main* del programma usato come *test*: alla pressione di un pulsante, sul *display* appare il carattere stampato su di esso. Nel caso in cui il valore `ADCdata` restituito dall'ADC sia inferiore a 32, si è certi che nessun tasto è stato premuto: questa condizione è quella più probabile, e per questo motivo si è scelto di iniziare i confronti con la soglia di valore più basso. Inoltre, in corrispondenza del primo *if*, il cursore dell'LCD viene posizionato all'estrema sinistra del *display*.

```
/*****************************************************
* File Name: main.c
* Project : keyboard16
* Abstract : the mV value on AD0.0 (P0.22) refers to a key
* pressed on a 4x4 keyboard.
* LDC displays the key value.
*****************************************************/

#include "lpc2103.h"
#include "framework.h"
#include "HD44780.h"
#include "adc.h"

int main(void)
{
   unsigned char dummyval;
   unsigned int ADCdata;
   SEL_ADC0; //set ADC CH0 (P0.22)
   dummyval = InitLCD();
   InitADC();
   PutCommand(DISPLAY_CLEAR); delayMs(5);

   while(1)
   {
      ADCdata = ReadADC(CH0, 4); //sample CH0, 16 readings
      if (ADCdata<32) {PutCommand(RETURN_HOME); delayMs(5);}

      else if (ADCdata<66) WriteChar('A');
        else if (ADCdata<71) WriteChar('3');
          else if (ADCdata<75) WriteChar('2');
            else if (ADCdata<81) WriteChar('1');
              else if (ADCdata<89) WriteChar('B');
                else if (ADCdata<97) WriteChar('6');
                  else if (ADCdata<107) WriteChar('5');
                    else if (ADCdata<120) WriteChar('4');
                      else if (ADCdata<137) WriteChar('C');
                        else if (ADCdata<158) WriteChar('9');
                          else if (ADCdata<187) WriteChar('8');
                            else if (ADCdata<230) WriteChar('7');
                              else if (ADCdata<298) WriteChar('D');
                                else if (ADCdata<107) WriteChar('#');
                                  else if (ADCdata<768) WriteChar('0');
                                    else WriteChar('*');
      delayMs(250);
      }
return 0;
}// EOF
```

La *routine* di gestione può essere inserita in una di servizio dell'*interrupt* generato quando si agisce su un qualunque tasto: è necessario, in questo caso, inserire un comparatore che riveli se V_{OUT} ha superato una soglia di ampiezza inferiore alla tensione che si ha con $P_{0,3}$ (pari a 210 mV circa).

4.7.2 Interfacciamento digitale

In caso di interfacciamento digitale, la lettura della posizione di un pulsante premuto su una tastiera organizzata a matrice è fatta con metodo di scansione: se abbiamo una matrice di *n* righe e *m* colonne, tale soluzione circuitale impegna *n* + *m* linee I/O digitali di connessione verso il μC.
L'interfacciamento della tastiera a matrice può essere demandato ad un *hardware* specifico nelle tastiere codificate, oppure a una gestione *software* da parte del processore per periferiche non codificate: in entrambi i casi, ad ogni tasto dovrà corrispondere un codice che consente al programma di gestione di identificare univocamente quale tasto è stato premuto.
In questo paragrafo sono mostrati i due metodi *software* che normalmente sono impiegati per la codifica del tasto premuto: i) scansione di colonna e lettura di riga; ii) con scambio del verso di lettura.

4.7.2.1 Metodo a scansione

Il metodo a scansione è illustrato in fig. 4.107. Un registro d'uscita, *Rc*, è collegato alle colonne della matrice di tasti. Ad ogni ciclo viene attivata una sola colonna ponendo allo stato logico basso il corrispondente *bit*. Il contenuto del registro di lettura *Rr* di riga viene acquisito e, se nessun tasto è premuto, tutti i suoi *bit* saranno a 1: in questo caso il contenuto di *Rc* viene fatto ruotare di un *bit* per attivare la colonna adiacente. Il ciclo si ripete finché almeno un *bit Rr* non è diverso da 1.

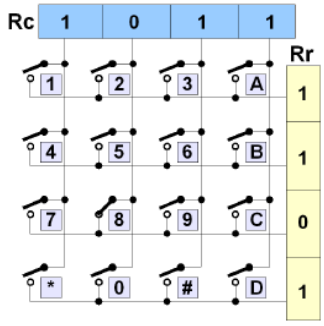


Fig. 4.107 *Metodo a scansione*

Nell'esempio di fig. 4.107, la pressione del tasto "8" bloccherà la scansione quando il contenuto di *Rc* è 0100, rilevando 0010 per *Rr*. I due valori saranno utilizzati per risalire al codice del tasto premuto ed avviare le operazioni eventualmente programmate.
Per verificare la suddetta funzionalità, i *pin* P0.[8..11] e P0.[12..15] dell'LPC2103 sono stati rispettivamente usati per pilotare le colonne e leggere le righe della tastiera 4 x 4.

Quattro resistori di *pull-up* assicurano lo stato P0.[12..15] = [1111] quando nessun tasto è premuto.
Nel *main* del programma di verifica si parte con l'inizializzazione del sistema di scansione. Le linee P0.[8..15] in uscita sono poste tutte pari a 0 tranne P.08.

```
IODIR = 0xF00;
IOSET = 0xF00; IOCLR = 0x800;
```

Nel ciclo *while(1)* si dovrà inserire il codice per la scansione:

```
if( (IOPIN & 0xF00) == 0xE00)
     IOPIN = 0x700;
else IOPIN = IOPIN >> 1;
```

che sarà a sua volta inserito nel ciclo utile alla lettura delle righe:

```
while( (IOPIN & 0xF000) == 0xF000);
```

da cui si esce se almeno un *bit* del *nibble* P0.[12..15] è pari a zero.
Quando l'esecuzione esce dal ciclo *while*, ciò che è contenuto nel *byte* b8..15 di `IOPIN` corrisponde al "codice" del tasto premuto. Come indicato nel listato qui di seguito, il suo valore, salvato nella variabile `dummyVal`, può essere osservato inserendo un opportuno *breakpoint* nel programma di *test*: il processore subirà un *halt* ogni volta che si preme un tasto, e si potrà andare a leggere il contenuto di `dummyVal`. Per avviare un nuovo ciclo di acquisizione sarà sufficiente premere *Resume* (F8).

```
#include "lpc2103.h"
#include "framework.h"

int main(void)
{
       unsigned int dummyVal;
       IODIR = 0xF00; // P0.8..11 as output (columns)
       IOSET = 0xF00; IOCLR = 0x800; // P0.8 = 1 (1st column)

       while(1)
       {
              while( (IOPIN & 0xF000)==0xF000)
                 if( (IOPIN & 0xF00) == 0xE00)
                       IOPIN = 0x700;
                 else
                       IOPIN = IOPIN >> 1;

              dummyVal = ((~(IOPIN & 0xFF00))>>8)&0xFF;

       /* INSERT A BREAKPOINT HERE TO VISUALIZE THE KEY "CODE" */
       }
return 0;
} // EOF
```

Non è mai opportuno che il codice di gestione di una periferica si trovi interamente nel *main* stesso del programma. Il *main* è solitamente pensato per lo svolgimento di un lavoro descritto a più alto livello. Nel caso generale, il listato precedente sarà inserito in una funzione che il *main* chiama quando si vuole conoscere lo stato della tastiera per l'acquisizione di comandi o dati.

4.7.2.2 Metodo con scambio del verso di lettura

Un secondo modo per acquisire il codice del tasto premuto si basa sull'idea espressa con la fig. 4.108.
In un primo periodo, il registro di colonna è configurato come uscita con tutti i *bit* a livello logico basso; il registro di riga, di ingresso, rileva l'evento di tasto premuto non appena uno dei suoi *bit* va a livello basso. A questo punto, si entra nel secondo periodo, in cui il registro di riga è configurato come uscita con contenuto pari alla configurazione rilevata al momento del cambio; analogamente, anche il registro di colonna cambia la sua funzionalità, diventa d'ingresso ed è ora in grado di memorizzare la posizione della colonna attiva.

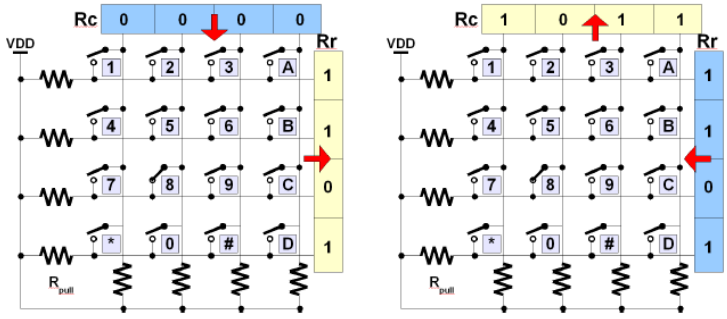


Fig. 4.108 *Metodo con scambio del verso di lettura*

Lo schema di montaggio è del tutto simile a quello del metodo a scansione, ma occorre aggiungere altri quattro resistori di *pull-up* sulle linee P0.[8..11].
La soluzione *software* di questa seconda metodologia è relativamente più semplice di quella vista nel paragrafo precedente: ora infatti non è necessaria alcuna scansione ma solo uno scambio nella "direzione" dei due registri coinvolti.
Inizialmente il *nibble* di colonna è configurato come uscita e i suoi 4 *bit* sono posti a 0:

```
IODIR = 0xF00;
IOCLR = 0xF00;
```

a cui segue l'attesa sul *nibble* di riga:

```
while( (IOPIN & 0xF000) == 0xF000);
```

In uscita dal *while* si deve avere l'acquisizione della configurazione dello stato delle righe, lo scambio della direzione delle due linee e la scrittura della configurazione acquisita per le righe:

```
dummyVal = IOPIN;
IODIR = 0xF000;
IOPIN = dummyVal;
```

L'ultima istruzione è per l'acquisizione del *byte* relativo al codice del tasto premuto:

```
dummyVal = IOPIN & 0xFF00;
```

Inserendo un *breakpoint* dopo quest'ultima assegnazione (vedi listato qui di seguito), la verifica del funzionamento del sistema sarà condotta come svolto per il metodo precedente di scansione. L'intero codice è qui di seguito riportato:

```
#include "lpc2103.h"
#include "framework.h"

int main(void)
{
	unsigned int dummyVal;

	while(1)
	{
		IODIR = 0xF00; // P0.8..11 as output (columns)
		delayMs(1);
		IOCLR = 0xF00; // all outputs to zero

		while( (IOPIN & 0xF000)==0xF000);
			dummyVal = IOPIN; //read rows configuration
			IODIR = 0xF000; // P0.12..15 as outputs (rows)
			delayMs(1);
			IOPIN = dummyVal; //rewrite rows configuration

			dummyVal = ((~(IOPIN & 0xFF00))>>8)&0xFF;

	/* INSERT A BREAKPOINT HERE TO VISUALIZE THE KEY "CODE" */

	}
return 0;
} // EOF
```

4.7.3 Schema d'interfacciamento tra tastiera e µC LPC2103

Nella fig. 4.109, è infine indicato lo schema adottato sulla *main board* di ECSIC32 per l'interfacciamento della tastiera con il µC (è presente, oltre alla tastiera, anche il *display* LCD con relativo connettore). E'implementato un interfacciamento analogico, con righe e colonne della matrice della tastiera collegate (attraverso un connettore), ad un partitore resistivo montato sulla *main board*; viene inoltre impiegato il "classico" comparatore LM311, che necessita di un'alimentazione di 5 V [64].
E' prevista però anche la possibilità di utilizzare, in un'altra applicazione, un interfacciamento digitale. A tale scopo sono stati inseriti i due *jumper* J4 e J5: la soluzione analogica (tastiera collegata a KEYB-A) prevede J4 e J5 chiusi, mentre la soluzione digitale (tastiera collegata a KEYB-D) prevede J4 e J5 aperti.

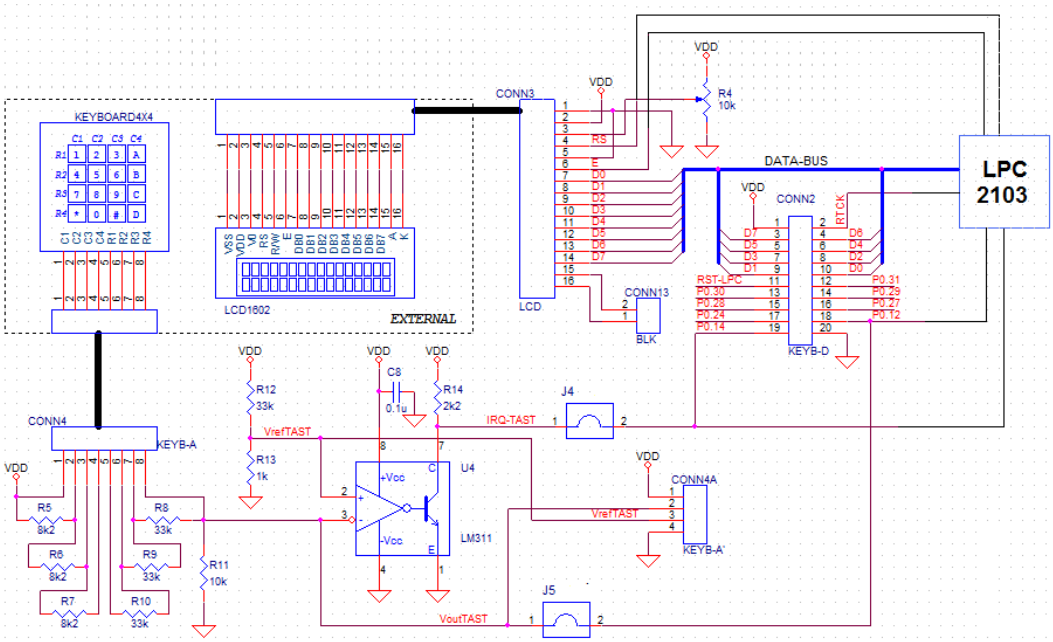


Fig. 4.109 *Schema circuitale del sistema d'interfacciamento con la tastiera a matrice*

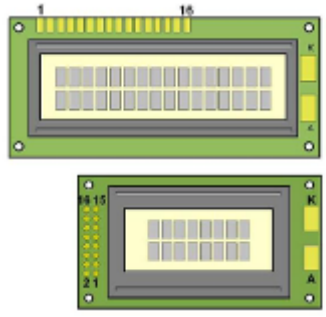
4.8. Interfacciamento con *display* alfanumerici a cristalli liquidi

I *display* alfanumerici (fig. 4.110) a matrice di punti a cristalli liquidi (LCD = *Liquid Crystal Display*) sono ampiamente utilizzati nei sistemi *embedded*: presentando un limitatissimo assorbimento di corrente, essi risultano particolarmente indicati in apparati alimentati a batteria.

Fig. 4.110 *Esempi di display LCD*

Il circuito di controllo e pilotaggio a bordo di tali LCD è studiato per limitare al minimo i problemi di interfacciamento di un microcontrollore (μC) all'LCD stesso: il lavoro del μC, infatti, è principalmente limitato alla semplice trasmissione dei dati che devono essere visualizzati sul *display*.

I caratteri che compongono la stringa sono inviati dal μC come sequenza di *byte* codificati (generalmente in ASCII) che il *controller* dell'LCD memorizza nella sua RAM interna; i codici sono quindi tradotti nei segnali utili al corretto pilotaggio degli elementi dell'LCD sul quale apparirà la stringa inviata.
Il circuito integrato HD44780 (Hitachi), per la sua ampia diffusione, è ormai considerato uno standard *de facto* quale *controller/driver* per i *display* alfanumerici [65]. Il *controller* può ricevere comandi specifici anziché dati (inizializzazione e *on/off* del *display*, *shift* e *blinking* del cursore, etc.). Non da ultimo, è anche possibile definire un certo numero di caratteri personalizzati. Per consentire l'*handshake* col sistema di controllo, l'HD44870 può rispondere ad interrogazioni inviando l'informazione utile alla determinazione dello stato corrente dell'LCD (*busy*).
Come mostrato in fig. 4.110, generalmente gli LCD alfanumerici sono montati su una basetta in vetroresina che facilita il montaggio del nell'apparato che lo impiega. La basetta consente la saldatura di un connettore a 16 *pin* su singola o doppia fila di contatti di collegamento verso il μC, a seconda che questi siano disposti sul lato maggiore o su quello minore della basetta. La piedinatura per questa possibile coppia di connettori è riportata in fig. 4.111:



Pin	nome	funzione
1	Vss	GND
2	Vdd	+5 V
3	Vo	contrasto
4	RS	reg. select
5	R/W	read/write
6	E	enable
7	DB0	data bus
8	DB1	data bus
9	DB2	data bus
10	DB3	data bus
11	DB4	data bus
12	DB5	data bus
13	DB6	data bus
14	DB7	data bus
15	A	4.2 V LED
16	K	0 V LED

Fig. 4.111 *Piedinatura per connettori di display LCD*

Oltre a due *pin* di alimentazione (1, 2), si deve fornire una tensione V_o sul *pin* 3 per regolare il contrasto dell'LCD. Seguono i tre piedini di controllo (4, 5, 6) e il *bus* dati a 8 *bit*. Infine, gli ultimi due *pin* (15, 16) servono per alimentare, con una tensione di 4 ÷ 5 V, il LED presente nei modelli con retro-illuminazione.

4.8.1 Il controller HD44870

Rappresentando ormai uno standard, si vuole dedicare un paragrafo alla breve descrizione del *controller* HD44780, il cui schema a blocchi semplificato è riportato in fig. 4.112.
Il dispositivo è compatibile con un microprocessore su *bus* dati a 8 *bit*: tale processore esterno verrà nel seguito denominato MPU (*MicroProcessor Unit)*. Le linee di controllo dell'HD44780 sono: RS (*Register Select*), R/W (*Read/Write*), ed E (*Enable*). La linea RS consente all'MPU di stabilire se l'invio di un *byte* sul *data bus*, DB0..7, si riferisce a un dato (RS=1) oppure a un'istruzione (RS=0). La linea R/W permette, sempre all'MPU, di selezionare l'invio (R/W=0) o la ricezione (R/W=1) di un *byte*. Infine, la linea E di abilitazione permette di sincronizzare le suddette operazioni di lettura o scrittura.

A seconda dello stato della linea RS, il *byte* inviato dall'MPU sarà caricato nel registro di istruzione (IR) per essere così decodificato, o in quello dati (DR): quest'ultimo è poi connesso al *bus* interno a 8 *bit* relativo alla memoria del *display*, DDRAM, avente la dimensione di 80 *byte*.

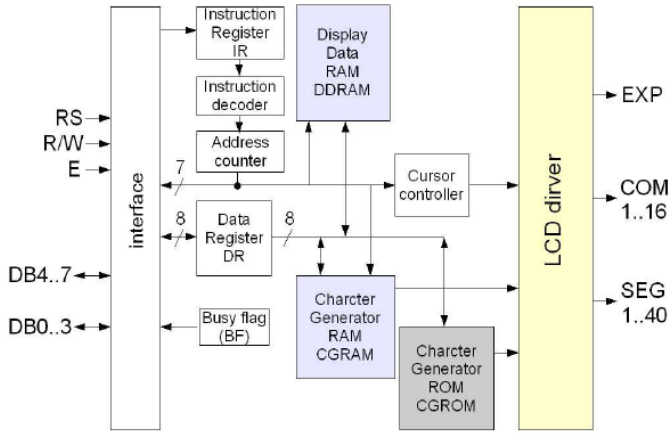


Fig. 4.112 *Schema a blocchi del controller HD44870*

Il *controller* dispone inoltre di un contatore di indirizzi (*Address Counter*) utilizzato per la scansione della DDRAM: questo potrà essere incrementato o decrementato automaticamente ad ogni invio di un *byte* che corrisponde al codice da memorizzare nella DDRAM stessa. Infine, per segnalare all'MPU che l'HD44780 è impegnato in un'operazione interna, è possibile accedere al *Busy Flag* (BF). Il contenuto dell'*Address Counter* e di BF è letto dall'MPU rispettivamente sui *bit* DB0..6 e DB7 del *data bus.*
Con una RAM per i dati da 80 *byte*, il *controller* è in grado di gestire LCD che visualizzino fino a 80 caratteri, distribuiti anche su più righe. Tuttavia il *driver* integrato nell'HD44780, attraverso le linee COM1..16 e SEG1..40, può pilotare LCD di 8 o 16 caratteri distribuiti su una o due righe seguendo gli schemi di fig. 4.113.

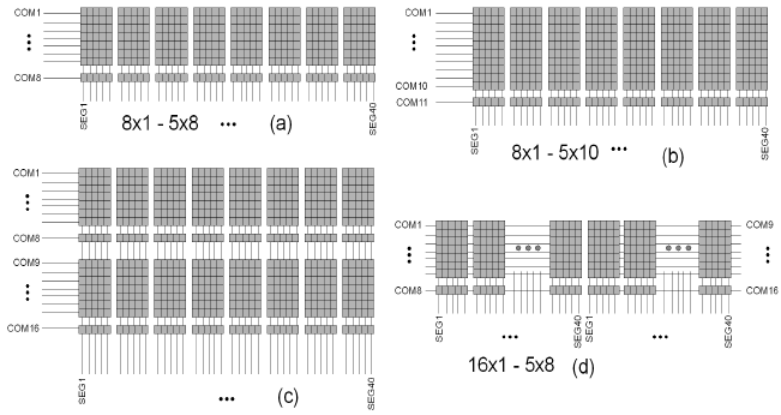


Fig. 4.113 *Schemi di pilotaggio di LCD a 8 o 16 caratteri*

Per *display* di dimensioni maggiori deve essere affiancato all'HD44780 un *driver* esterno interfacciato attraverso le linee EXP di fig. 4.112.
L'HD44780 contiene una memoria non volatile denominata CGROM (*Character Generator ROM*) con la quale il codice a 8 *bit* è convertito nell'opportuna sequenza dei 5 *bit* di riga di ciascun carattere (i caratteri hanno un *font* 5 x 8 o 5 x 10 pixel). La dimensione della CGROM è pari a 9920 *bit*, così da rendere disponibile un *set* composto da 208 caratteri con *font* 5 x 8 (8320 *bit*) e da 32 caratteri con *font* 5 x 10 (1600 *bit*). Nella CGROM cod. A00 è memorizzato il *set* di caratteri riportato in fig. 4.114, in cui sono presenti alcuni caratteri sillabici (*kana*) giapponesi. I caratteri con *font* 5 x 10, presenti nelle ultime due colonne, hanno un codice con il *nibble* alto pari a 0xE o 0xF. Si può notare che nella prima colonna, con *nibble* alto pari a 0x0, sono riportate le locazioni di memoria della CGRAM (*Character Generator* RAM), una memoria volatile da 64 *byte* che consente di memorizzare fino a 8 caratteri personalizzati con *font* da 5 x 8 pixel o 4 caratteri con *font* 5 x 10 pixel.

Fig. 4.114 *Set di caratteri presenti nella CGROM del controller HD44780*

L'HD44780 possiede un *set* di istruzioni che garantisce il controllo completo dell'LCD da parte dell'MPU. I comandi che l'MPU può impartire sono:

- *Function set*: permette di impostare l'interfaccia (a 4 o 8 *bit*), il numero di linee dell'LCD (1 o 2) e il tipo di *font* (5 x 8 o 5 x 10 pixel).
- *Clear display*: permette di cancellare l'intero *display* mentre il contatore degli indirizzi è resettato.
- *Return home*: è usata per azzerare il contatore indirizzi e porre il *display* nella sua posizione iniziale (così da visualizzare la stringa di caratteri memorizzata nella DDRAM a partire dall'indirizzo 0x00).
- *Display on/off*: è utile per spegnere o accendere sia il *display* che il cursore, e per impostare o meno il *blinking* del cursore.
- *Entry mode set*: permette di impostare il movimento del cursore (verso destra o verso sinistra) oppure stabilire se l'intera stringa visualizzata deve subire uno *shift* all'invio di ogni nuovo carattere.
- *Cursor or display shift*: viene usata per spostare il cursore o i caratteri visualizzati di una posizione verso destra o verso sinistra.

Tutte le istruzioni sono a 8 *bit*, quindi il singolo comando è impartito inviando lo specifico *byte* sul *bus* DB0..7, mantenendo a livello logico basso la linea RS. La scrittura dell'istruzione nel registro IR avviene sul fronte negativo di E, e mantenendo a livello logico basso l'ingresso R/W.
Per la scrittura o lettura dei dati o degli indirizzi si hanno invece:

- *Set DDRAM address*: permette di impostare l'indirizzo della DDRAM nel contatore indirizzi. L'operazione di scrittura o di lettura successiva a tale istruzione riguarderà la locazione di memoria qui indicata.
- *Set CGRAM address*: permette di impostare l'indirizzo della CGRAM per poter inviare i *byte* relativi ai caratteri personalizzati. L'operazione di scrittura o di lettura successiva a tale istruzione riguarderà la locazione di memoria qui indicata.
- *Read busy flag and address*: viene utilizzata per la lettura dello stato di BF e del contenuto del contatore di indirizzi.
- *Write data*: permette di scrivere un *byte* nella DDRAM o CGRAM. Dopo questa istruzione il contatore d'indirizzi della RAM è automaticamente aggiornato (incrementato o decrementato).
- *Read data*: permette di leggere un *byte* dalla DDRAM o CGRAM. Dopo questa istruzione il contatore d'indirizzi della RAM è automaticamente aggiornato (incrementato o decrementato).

Anche queste istruzioni sono inviate al *controller* seguendo le stesse modalità enunciate per i comandi [66]. Per le ultime due, ovviamente, la linea RS dovrà stare a livello logico alto, affinché il *byte* si riferisca al registro DR dei dati. Per le operazioni di lettura, R/W dovrà essere posta a livello logico alto. Il segnale E costituisce sempre il segnale di "sincronismo", sia per le operazioni di scrittura che di lettura.
Nel caso in cui l'interfaccia sia a 4 *bit*, ciascun comando o *byte* di dato dovrà essere inviato o letto con un doppio impulso di E: il primo si riferisce al *nibble* più significativo, il secondo a quello meno significativo del *byte* da trasferire.

4.8.2 Schema d'interfacciamento tra *display* LCD e µC LPC2103

In fig. 4.115 è riportato lo schema seguito per i collegamenti tra il *display* LCD utilizzato per il sistema ECSIC32 (modello PC1602ARU [67]) e il µC LPC2103.

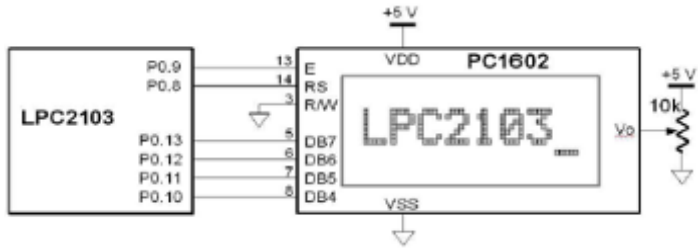


Fig. 4.115 *Schema di collegamento fra display LCD e microcontrollore*

Per limitare il numero delle connessioni verso l'LPC2103 è stato scelto di usare l'interfacciamento a 4 *bit* anziché 8: in questo caso viene solo usato il *nibble* alto DB4..7 del *bus* dati. La linea R/W è connessa a massa, e non sarà dunque possibile stabilire una comunicazione con *handshake* che impieghi il *flag* BF per sapere quando l'HD44780 è libero: tuttavia, noti i tempi massimi di elaborazione del *controller* stesso, sarà sufficiente inserire degli opportuni ritardi tra comandi successivi inviati dal µC.
Infine si osservi che l'LCD deve essere alimentato con 5 V, e non con 3.3 V. Entrambe le tensioni sono disponibili sulla *main board* che viene impiegata.

4.8.3 *Software* di gestione per LCD con *controller* HD44780

In questo paragrafo viene presentata l'infrastruttura *software* con cui è possibile definire costanti e comandi utili a un controllo agevole dell'LCD. Saranno per esempio definite funzioni quali l'inizializzazione, la scrittura di un carattere, oppure di una stringa, o ancora di un numero decimale a più cifre.
Il *file* di intestazione (*header*) è denominato HD44780.h, ed è relativo alla "libreria" di funzioni del *controller*. In base ai collegamenti con l'LPC2103, le prime definizioni sono quelle relative alle connessioni di E e RS; si definiscono poi il *bus* di controllo (E, RS) e il *bus* dati a 4 *bit* (DB4..7). Si può notare che vengono posti a 1 i *bit* corrispondenti alle posizioni P0.x scelte per i collegamenti:

```
#define E  0x00000200 //P0.9 is E
#define RS 0x00000100 //P0.8 is RS
#define CTRL_BUS 0x00000300 //P0.[9:8] EN, RS
#define DATA_BUS 0x00003C00 //P0.[13:10]DB7..4
#define FULL_BUS 0x00003F00 //P0.[13:8]=DB7..4,E,RS
```

Qualora venga adottato uno schema di collegamento diverso, sarà sufficiente un riadattamento dei valori delle singole definizioni, mentre potrà rimanere del tutto invariato qualunque altro codice che ad esse fa riferimento.
L'invio di un *nibble* sul *bus* DB4..7 potrà in generale derivare dal contenuto di una variabile i cui *bit* dovranno essere spostati in modo che l'LSB coincida con P0.10 (= DB4). Per questo motivo è opportuna la definizione:

```
#define DB4_PIN_SHIFT 10 //DB4 is on P0.10
```

equivalente al valore 10. Qualunque dato a 4 *bit* che dovrà andare o provenire dall'HD44780, sarà spostato di `DB4_PIN_SHIFT` *bit* affinché possa giungere correttamente sul *data bus* dell'LCD.
Conviene anche definire l'inizializzazione del *bus*: i *pin* devono essere in uscita e tutti posti allo stato logico basso. Per questo vengono inserite le seguenti righe di codice:

```
#define FULL_BUS_CLEAR IOCLR = FULL_BUS
#define FULL_BUS_OUT IODIR |= FULL_BUS
#define INIT_BUS {FULL_BUS_CLEAR;FULL_BUS_OUT;}
```

in cui si può osservare che la terza definizione sfrutta le prime due. All'inizializzazione del sistema, per impostare correttamente lo stato del *bus*, è sufficiente scrivere `INIT_BUS`.
Per l'HD44780 la scrittura nel registro istruzione (dati) è determinata dallo stato basso (alto) della linea RS. È utile allora definire i due valori con i nomi IR e DR, più immediati da ricordare:

```
#define IR 0 //write to instruction register
#define DR 1 //write to data register
```

Infine, viene specificata la dimensione dell'LCD utilizzato, sia in termini di dimensione della singola riga che del numero totale di caratteri:

```
#define TOTAL_CHARACTERS_OF_LCD 16
#define LCD_LINE_LENGTH 16
```

Altrettanto fondamentali sono le definizioni relative ai comandi che possono essere impartiti al *controller* HD44780. Per maggiore versatilità nell'uso di tali comandi, la loro definizione impiega nomi che riprendono l'azione del comando specifico; nella definizione sono inoltre specificati i valori che devono essere inviati al *controller* per l'esecuzione delle istruzioni specifiche. Ad esempio:

```
#define DISPLAY_CLEAR 0x01 //0b00000001
#define RETURN_HOME 0x02 //0b00000010
```

Allo stesso modo saranno definiti comandi del tipo: `ENTRY_MODE`, `CURSOR_MOVE`, `SHIFT`, etc.
Infine, il *file HD44780.h* contiene le definizioni delle funzioni fondamentali che permettono di gestire idoneamente il *display*, quali:

- *InitLCD:* inizializzazione dell'LCD;
- *WriteByte*: invio di un *byte*;
- *WriteChar*: scrittura di un carattere (inviando il codice ASCII);
- *WriteString*: scrittura di una stringa di caratteri;

- *WriteLine*: scrittura di una stringa su una linea specifica (utile per LCD a più righe);
- *GoToLine*: spostamento del cursore su una linea (per LCD a più righe);
- *PutCommand*: invio di un'istruzione all'LCD;
- *Write_ndigitvalsval*: scrittura di un numero decimale a più cifre (passando il valore dello stesso e non il codice ASCII);
- *Write_2digitvalsval*: scrittura di un numero decimale a due cifre (passando il valore dello stesso e non il codice ASCII).

Relativamente alle due ultime funzioni, conviene inserire definizioni utili alla trasformazione di un valore decimale a una cifra nel corrispondente codice ASCII (e viceversa). Si passa dal valore numerico al codice semplicemente aggiungendo 0x30; viceversa, per passare dal codice ASCII della cifra al suo valore sarà sufficiente sottrarre la quantità 0x30. Quindi si può definire:

```
#define NUM_TO_CODE(num) (num+0x30) //0-9 ROM codes
#define CODE_TO_NUM(code) (code-0x30)//0-9 ROM num
```

Qui di seguito viene riportato il listato completo del *file HD44780.h*:

```
/*****************************************************
* Filename: HD44780.h
*
* Processor: LPC2103
* Compiler : gcc version 4.1.1
* Date/Time: 07/11/09
* Abstract : HD44780 header file
*****************************************************/

#ifndef __HD44780_H
#define __HD44780_H

/* MODULE HD44780 */

#include "lpc2103.h"
#include "framework.h"

#define E  0x00000200 //P0.9 is E
#define RS 0x00000100 //P0.8 is RS
#define FULL_BUS 0x00003F00 //P0.[13:8] DB7,DB6,DB5,DB4,EN,RS
#define CTRL_BUS 0x00000300 //P0.[9:8] EN, RS
#define DATA_BUS 0x00003C00 //P0.[13:10]DB7,DB6,DB5,DB4
#define DB4_PIN_SHIFT 10 //DB4 is on P0.10
#define DB4_PIN 0x00000400 //DB5 is on P0.10
#define DB5_PIN 0x00000800 //DB5 is on P0.11
#define FULL_BUS_CLEAR IOCLR = FULL_BUS //Full bus clear
#define FULL_BUS_OUT IODIR |= FULL_BUS //Full bus as output
#define INIT_BUS {FULL_BUS_CLEAR;FULL_BUS_OUT;}
#define IR 0 //write to instruction register
#define DR 1 //write to data register
#define TOTAL_CHARACTERS_OF_LCD 8
#define LCD_LINE_LENGTH 8
```

```
#define DISPLAY_CLEAR 0x01 //0b00000001
#define RETURN_HOME 0x02 //0b00000010
#define ENTRY_MODE_DEC_NO_SHIFT 0x04 //0b00000100
#define ENTRY_MODE_DEC_SHIFT 0x05 //0b00000101
#define ENTRY_MODE_INC_NO_SHIFT 0x06 //0b00000110
#define ENTRY_MODE_INC_SHIFT 0x07 //0b00000111
#define DISP_OFF_CUR_OFF_BLINK_OFF 0x08 //0b00001000
#define DISP_OFF_CUR_OFF_BLINK_ON 0x09 //0b00001001
#define DISP_OFF_CUR_ON_BLINK_OFF 0x0A //0b00001010
#define DISP_OFF_CUR_ON_BLINK_ON 0x0B //0b00001011
#define DISP_ON_CUR_OFF_BLINK_OFF 0x0C //0b00001100
#define DISP_ON_CUR_OFF_BLINK_ON 0x0D //0b00001101
#define DISP_ON_CUR_ON_BLINK_OFF 0x0E //0b00001110
#define DISP_ON_CUR_ON_BLINK_ON 0x0F //0b00001111
#define CURSOR_MOVE_SHIFT_LEFT 0x10 //0b00010000
#define CURSOR_MOVE_SHIFT_RIGHT 0x14 //0b00010100
#define DISPLAY_MOVE_SHIFT_LEFT 0x18 //0b00011000
#define DISPLAY_MOVE_SHIFT_RIGHT 0x1C //0b00011100
#define FOUR_BIT_ONE_LINE_5x8 0x20 //0b00100000
#define FOUR_BIT_ONE_LINE_5x10 0x24 //0b00101000
#define FOUR_BIT_TWO_LINE_5x8 0x28 //0b00101000
#define FOUR_BIT_TWO_LINE_5x10 0x2C //0b00101100
#define EIGHT_BIT_ONE_LINE_5x8 0x30 //0b00110000
#define EIGHT_BIT_ONE_LINE_5x10 0x34 //0b00110100
#define EIGHT_BIT_TWO_LINE_5x8 0x38 //0b00111000
#define EIGHT_BIT_TWO_LINE_5x10 0x3C //0b00111100


#define NUM_TO_CODE(num) (num+0x30) // 0-9 ROM codes
#define CODE_TO_NUM(code) (code-0x30) // 0-9 ROM num

unsigned char WriteByte(unsigned char rs,
                        unsigned char data_to_LCD);
unsigned char InitLCD(void);
unsigned char WriteChar(unsigned char character);
unsigned char PutCommand(unsigned char Command);
unsigned char GoToLine(unsigned char line);
void WriteString(unsigned char
                 LineOfCharacters[TOTAL_CHARACTERS_OF_LCD]);
void WriteLine(unsigned char
               LineOfCharacters[LCD_LINE_LENGTH], int line);
void Write_ndigitsval(unsigned int dummyVal,
                      unsigned char ndigits);
void Write_2digitsval(unsigned int dummyVal);

/* END HD44780 */

#endif /* __HD44780_H*/
```

Il *file* termina con la dichiarazione delle nove funzioni di utilità che permettono di gestire più facilmente l'LCD: i codici specifici di tali funzioni sono riportati in un *file HD44780.c* in cui, per poter utilizzare quanto definito nell'*header*, dovrà apparire la direttiva:

```
#include "HD44780.h".
```

Nelle successive sezioni saranno illustrate le funzioni di utilità, sottolineando le peculiarità che alcune di esse presentano.

4.8.3.1 Funzione 'InitLCD' di inizializzazione

In fig. 4.116 è riportato il diagramma di flusso dell'algoritmo da seguire, secondo le specifiche del *datasheet*, nel caso in cui si voglia predisporre l'interfacciamento a 4 *bit*.
Si può notare che è necessario ripetere tre volte l'invio della configurazione di *bit* "0011XXXX" sul *data bus* con diverse pause. La linea RS = 0 indica che il trasferimento è verso IR. Durante l'invio della sequenza ha senso solo il *nibble* alto del *data bus*.
La sequenza suddetta viene riconosciuta dall'HD44780 come operazione di "*reset software*" e consente di predisporre il *controller* alla corretta comunicazione con l'MPU.
Nel listato dell'*header* sono state appositamente inserite le definizioni:

```
#define DB4_PIN 0x00000400 //DB5 is on P0.10
#define DB5_PIN 0x00000800 //DB5 is on P0.11
```

che, in *InitLCD*, rendono immediato il *set* o *clear* proprio dei *bit* DB4 e DB5 qui necessari.

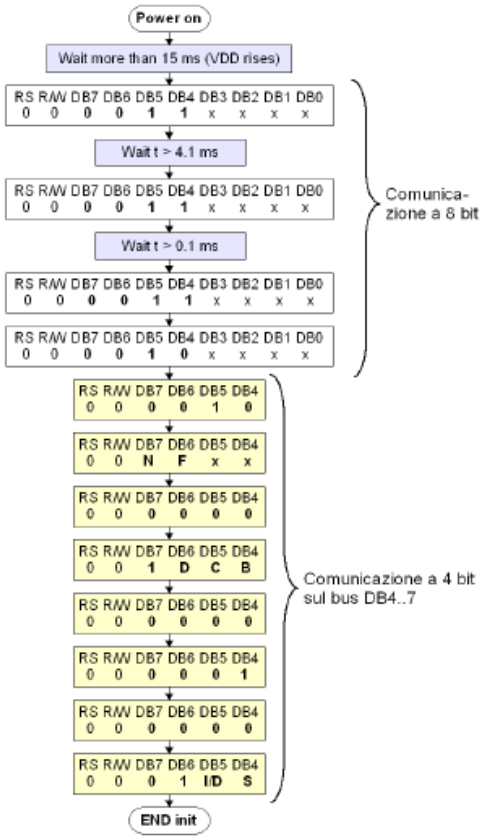


Fig. 4.116 *Diagramma di flusso per interfacciamento a 4 bit*

Dopo la sequenza iniziale, viene inviata al controllore l'istruzione "0010XXX": con essa l'HD44780 riconosce il *bit* DB4 = DL (*Data Length*) a livello logico basso, che corrisponde al comando di interfacciamento a 4 *bit*. Dopo questa istruzione, la comunicazione potrà procedere sfruttando appieno le sole linee DB4..7, inviando un intero *byte* con due impulsi di E. Il diagramma di flusso termina con quattro istruzioni di configurazione verso il registro istruzione (RS = 0):

- "0010NFXX": con cui, oltre ad indicare l'interfacciamento a 4 *bit*, si stabilisce la dimensione dell'LCD, numero di righe e *font*;
- "00001DCB": *on/off* del *display*, del cursore e del *blinking*;
- "00000001": cancellazione del *display*;
- "000001IS": incremento o decremento, e *shift on/off*.
-

Il codice completo della funzione *InitLCD* è qui di seguito riportato:

```
unsigned char InitLCD(void)
      {
            unsigned char errCode;
            INIT_BUS //P0.[13:8] def.: FULL_BUS_CLEAR; FULL_BUS_OUT
            delayMs(100);// Delay a minimum of 100 ms

        // This part is on 8 bit interface
            IOSET = DB5_PIN; // DB7=0, DB6=0, DB5=1, DB4=1,
            IOSET = DB4_PIN; // NU=0, E=0, RW=0, RS=0
            delayUs(1); // Wait a minimum of 60ns

            IOSET = E;  // Raise E start write operation
            delayUs(1); // Wait a minimum of 450ns
            IOCLR = E;  // Clear E finish write operation
            delayMs(5); // Wait a minimum of 4.1 ms

            IOSET = E;  // Raise E start write operation
                  delayUs(1); // Wait a minimum of 450ns
            IOCLR = E;  // Clear E finish write operation
            delayUs(200); // Wait a minimum of 100 us

            IOSET = E;  // Raise E start write operation
            delayUs(1); // Wait a minimum of 450ns
            IOCLR = E;  // Clear E finish write operation
            delayUs(100); // Wait a minimum of 100 us

            IOSET = DB5_PIN; //DB[7..4]=0010,NU=0,E=0, RW=0, RS=0
            IOCLR = DB4_PIN; // SETS 4-bit operation
            delayUs(1); // Wait a minimum of 60ns
            IOSET = E;  // Raise E start write operation

            delayUs(1); // Wait a minimum of 450ns
            IOCLR = E;  // Clear E finish write operation
            delayMs(5); // Wait a minimum of 5 ms 10ns
        // This part is on 8 bit interface

        // Define function set 4 bit interface,1-line display,5x10 font
        // errCode = WriteByte(IR, FOUR_BIT_ONE_LINE_5x10);
```

```
    // Define function set 4 bit interface,1-line display, 5x8 font
        errCode = WriteByte(IR, FOUR_BIT_ONE_LINE_5x8);
        delayMs(1); // Wait until LCD is free

    // Define display on/off control
        errCode = WriteByte(IR, DISP_ON_CUR_OFF_BLINK_OFF);
        delayMs(1); // Wait until LCD is free
        errCode = WriteByte(IR, DISPLAY_CLEAR); // Clear display
        delayMs(1); // Wait until LCD is free

    // Entry mode set cursor direction incr., do not shift display
        errCode = WriteByte(IR, ENTRY_MODE_INC_NO_SHIFT);
        delayMs(10); // Wait until LCD is free
        return errCode;
}
```

La prima parte si riferisce all'impostazione "0011XXXX" del *data bus* inviata tre volte generando tre impulsi sulla linea di enable E: le pause da 1 µs inserite dopo il *set* della linea garantiscono una durata superiore ai 450 ns degli impulsi positivi, come specificato nel *datasheet* del componente. Segue l'invio del *nibble* "0010", che indica che la comunicazione procederà su un *bus* a 4 *bit*.
Vengono poi inviate le quattro istruzioni che, oltre a cancellare l'intero contenuto della DDRAM, configurano l'HD44780 con la dimensione dell'LCD (numero di linee e pixel per carattere), nonché la modalità di incremento o decremento automatico e le funzionalità per il cursore.
Per semplificare l'invio delle quattro istruzioni, conviene utilizzare una funzione specifica di scrittura del *byte* da inviare: *WriteByte*. I parametri d'ingresso saranno: RS (pari a IR o DR a seconda che il *byte* debba essere caricato nel registro istruzione o in quello dati) e il contenuto del *byte* stesso da scrivere.
Per esempio, secondo le definizioni date in *HD44780.h*:

```
WriteByte(IR, FOUR_BIT_ONE_LINE_5x8);
```

permetterà di stabilire un interfacciamento a 4 *bit*, una dimensione di una riga e un *font* di 5 x 8 pixel per l'LCD. La funzione *WriteByte* verrà descritta in dettaglio nel paragrafo seguente.

4.8.3.2 Funzione 'WriteByte'

L'intero *byte* deve essere inviato con due impulsi di E partendo dal *nibble* più significativo. Inoltre si deve prevedere se gli 8 *bit* dovranno essere memorizzati nel registro istruzione oppure in quello dati. Per tale motivo si inserisce nella dichiarazione il parametro di ingresso *rs* (oltre a *data_to_LCD* a 8 *bit*): se *rs* è pari a 0, allora la linea RS dovrà subire un *clear*, in ogni altro caso la linea sarà settata.
Per inviare il *byte* partendo dal *nibble* più significativo è necessario selezionare i 4 *bit* superiori del *byte* `data_to_LCD`, e spostarli fino ad allinearli con le uscite P0.10..13 corrispondenti a DB4..7. Per questo vengono inserite le seguenti righe di codice:

```
dataVal = data_to_LCD & 0xF0;
dataVal = dataVal << (DB4_PIN_SHIFT-4);
```

La prima istruzione copia il *byte* da trasmettere nella variabile locale `dataVal`, selezionando con l'operatore AND il solo *nibble* più significativo. La seconda sposta il contenuto di *dataVal* del numero di *bit* necessario per l'allineamento con i *pin* d'uscita (il *nibble* più significativo si trova già 4 *bit* più sinistra, quindi, rispetto a quanto definito per `DB4_PIN_SHIFT`, serve uno *shift* diminuito di 4 *bit*.)
Si passa poi a configurare lo stato del registro `IOPIN`: si azzera il contenuto dei *bit* corrispondenti a i *pin* DB4..7 ponendo `IOPIN` in AND con `DATA_BUS` negato. Successivamente il valore `dataVal`, opportunamente già spostato, è posto in OR col risultato dell'operazione logica precedente:

```
IOPIN = (IOPIN & ~DATA_BUS) | dataVal;
```

Dopo aver generato un impulso sulla linea E si può inviare il successivo *nibble* meno significativo di `data_to_LCD` con righe di codice simili a quelle descritte sopra:

```
dataVal = data_to_LCD & 0x0F;
dataVal = dataVal << DB4_PIN_SHIFT;
IOPIN = (IOPIN & ~DATA_BUS) | dataVal;
```

Il listato completo della funzione è il seguente:

```
unsigned char WriteByte(unsigned char rs, unsigned char data_to_LCD)
      {
         unsigned int dataVal;
            if(rs)
                  IOSET = RS; // Set RS (DR operation)
            else
                  IOCLR = RS; // Clear RS (IR operation)

            delayUs(1); // Wait a minimum of 60ns
            IOSET = E;  // Set E to start first write operation
            dataVal = data_to_LCD & 0xF0; //Copy upper nibble
            dataVal = dataVal << (DB4_PIN_SHIFT-4); //align on DATA_BUS
            IOPIN = (IOPIN & ~DATA_BUS)|dataVal; //DB=upper nibble
            delayUs(1); // Wait a minimum of 195ns
            IOCLR = E;  // Clear E finish first write operation
            delayUs(1); // Wait a minimum of 530ns between writes
            IOSET = E;  // Raise E start second write operation
            dataVal = data_to_LCD & 0x0F; // Copy lower nibble
            dataVal = dataVal << DB4_PIN_SHIFT; //align to DB4
            IOPIN = (IOPIN & ~DATA_BUS)|dataVal;//DB=lower nibble
            delayUs(1); // Wait a minimum of 195ns
            IOCLR = E;  // Clear E finish second write operation
            delayUs(1); // Wait a minimum of 30ns
            return 0;

      }
```

4.8.3.3 Funzione 'PutCommand'

Avendo definito *WriteByte*, è immediato costruire la funzione che invia un comando all'HD44780. Sarà sufficiente chiamare la funzione *WriteByte(IR, comando)*, dove "*comando"* sarà scritto in base alle definizioni date in *HD44780.h*:

```
unsigned char PutCommand(unsigned char Command)
{
	delayMs(1); // Wait until LCD is free
	return WriteByte(IR, Command); // Write a byte to IR
}
```

4.8.3.4 Funzione 'WriteChar'

Altrettanto immediato è definire la funzione che invia un carattere all'LCD: basterà chiamare *WriteByte(DR, carattere)* dove "*carattere*" corrisponde al codice del carattere da inviare. L'unica differenza formale rispetto alla precedente funzione è il valore del parametro *rs* che viene passato, ora pari a *DR*:

```
unsigned char WriteChar(unsigned char character)
{
	delayMs(1); // Wait until LCD is free
	return WriteByte(DR, character); // Write a character to DR
}
```

4.8.3.5 Funzione 'WriteString'

L'estensione della funzione che invia un solo carattere è rappresentata da una funzione che consente la trasmissione di una stringa completa di caratteri all'LCD. Si definisce allora una funzione *WriteString* che riceve come parametro di ingresso un vettore che abbia dimensione massima pari a quella che l'LCD permette, e restituisce un parametro d'uscita nullo:

```
void WriteString(unsigned char
                 LineOfChar[TOTAL_CHARACTERS_OF_LCD])
```

La funzione preleva un elemento alla volta dal vettore e lo invia all'LCD con la funzione *WriteChar*. La scansione si arresta nel caso in cui si giunge alla fine della stringa: normalmente il carattere utilizzato è EOS (*End Of String*), che corrisponde al codice ASCII pari a zero (carattere *null*). Se il carattere EOS non viene incontrato, ha senso arrestare la scansione quando si è giunti alla dimensione massima della DDRAM. Da tali considerazioni si giunge facilmente alla soluzione riportata nel listato qui di seguito: si noti che è stata aggiunta la possibilità di distribuire l'intera stringa su tutte le righe dell'LCD.

```
void WriteString
(unsigned char LineOfCharacters[TOTAL_CHARACTERS_OF_LCD])
{
	unsigned char i=0, errCode, line=0;
	while(LineOfCharacters[i] && i<TOTAL_CHARACTERS_OF_LCD)
	{
		if((i%LCD_LINE_LENGHT)==0) GoToLine(++line);
		errCode = WriteChar(LineOfCharacters[i]);
		i++;
	}
}
```

N.B. Per la chiamata alla funzione *WriteString* si dovrà ricorrere al puntatore alla stringa che si vuole inviare. Nel linguaggio C per indicare l'uso di un puntatore si usa il simbolo

“ * ”. Così, se per esempio vogliamo scrivere “LPC2103” sul *display*,si dovrà compilare la seguente riga:

```
WriteString((unsigned char*)"LPC2103");
```

4.8.3.6 Funzione ‘Write_2digitsval’

Molto spesso la visualizzazione sul *display* riguarda il valore di una variabile che deve essere scritto in formato decimale. Quando il valore è rappresentato da una cifra soltanto è sufficiente utilizzare la definizione `NUM_TO_CODE` nell’argomento di *WriteChar*:

```
WriteChar(NUM_TO_CODE(val));
```

dove `val` è la variabile il cui contenuto dovrà essere compreso tra 0 e 9. Se invece il numero è a due cifre decimali occorre ricavare il codice della cifra delle unità e quella delle decine. È opportuno allora definire una funzione specifica che invii i due codici:

```
void Write_2digitsval(unsigned int dummyVal)
{
      WriteChar(NUM_TO_CODE(dummyVal/10));
      dummyVal %= 10;
      WriteChar(NUM_TO_CODE(dummyVal));
}
```

Si noti l’ovvio utilizzo dell’operatore “ % “ che consente di calcolare il resto della divisione.

4.8.3.7 Funzione ‘Write_ndigitsval’

Se il numero decimale è a più cifre, la funzione precedente deve essere estesa iterando il calcolo di ciascuna cifra che compone il numero. Si definisce a tale scopo la funzione *Write_ndigitsval* in cui viene inviato, oltre al valore del numero da visualizzare, anche il numero di cifre di cui esso è composto:

```
void Write_ndigitsval(unsigned int dummyVal, unsigned char ndigits)
```

Il valore numerico è inviato attraverso la variabile `dummyVal`. Il calcolo della singola cifra viene svolto a partire da quella più significativa: occorre allora dividere il numero per 10 *n*, con *n* decrementato di una unità ad ogni iterazione e variabile da `ndigits` a 0.
Il calcolo di un esponenziale prevede la necessità di disporre di una libreria matematica: è allora più semplice ricorrere al codice

```
while(--ndigits) ten_base*=10;
```

che, partendo da `ten_base` = 1, arriva dopo `ndigits` passi al valore esattamente pari a 10 `ndigits`. Il codice riduce il calcolo di una potenza con base 10 alla semplice moltiplicazione per 10.
Calcolato il valore di `ten_base` attuale, la singola cifra sarà estratta con la divisione:

```
digit = dummyVal / ten_base;
```

L’aggiornamento del valore da visualizzare avviene col calcolo del resto della divisione per `ten_base`:

```
dummyVal %= ten_base;
```

Ad ogni iterazione il valore di `ten_base` deve essere diminuito di un fattore 10. Il codice completo della funzione *Write_ndigitsval* è scritto qui di seguito:

```
void Write_ndigitsval(unsigned int dummyVal, unsigned char ndigits)
      {
            unsigned int ten_base=1, digit, leading_zeroes_flag = 1;
            while(--ndigits)
            ten_base*=10;
            while(ten_base)
            {
                 digit = dummyVal/ten_base;
                 if(digit)
                      leading_zeroes_flag = 0;
                      WriteChar(leading_zeroes_flag&&(ten_base>1)? ' ':
                                    NUM_TO_CODE(digit));
                  dummyVal %= ten_base;
                  ten_base/=10;
            }
      }
```

4.8.3.8 Programma di test per l’LCD

Con le funzioni ora costruite è stato compilato un semplice programma di *test* per l’LCD. Nel file *main.c* creato è stata inserita a tale scopo la direttiva:

```
#include "HD44780.h"
```

in modo che le funzioni precedentemente descritte possano essere utilizzate.
Oltre alla definizione di eventuali variabili locali, prima del ciclo *while(1)* va inserita l’istruzione di inizializzazione del *display*:

```
errCode = InitLCD();
```

che avrebbe il medesimo effetto anche scrivendo `InitLCD()`, senza cioè l’assegnazione a `errCode` del valore restituito.
Nel ciclo di lavoro (infinito) sono stati inseriti i seguenti comandi:

```
PutCommand(DISPLAY_CLEAR); delayMs(5);
PutCommand(RETURN_HOME); delayMs(5);
WriteString((unsigned char*)"LPC2");
WriteChar('1');
Write_2digitsval(3);
```

che consentono di far apparire sul *display* la scritta di *test* "LPC2103". Si noti che per i comandi `DISPLAY_CLEAR` e `RETURN_HOME` va inserito un ritardo di alcuni ms come specificato nel *datasheet* dell'HD44780.

4.9 Interfacciamento con il PC

Nel sistema ECSIC32 l'interfacciamento con il PC (al quale vengono inviati i dati), avviene tramite l'interfaccia seriale RS232: i due dispositivi messi in comunicazione dall'interfaccia RS232 sono la periferica UART (interna al microcontrollore LPC2103) ed il PC. La trasmissione avviene in modalità *half-duplex*, ovvero è bidirezionale ma non contemporanea nei due versi: in un determinato istante, dei due dispositivi uno trasmette e l'altro riceve e, quando necessario, si scambiano di ruolo. Alla comunicazione sono riservate le due linee *Tx* (*talker*) e *Rx* (*listener*), rispettivamente assegnate ai *pin* P0.0 e P0.1 dell'LPC2103.

4.9.1 L'interfaccia seriale RS-232

E' uno standard costituito da una serie di protocolli meccanici, elettrici ed informatici che rendono possibile lo scambio di informazioni a bassa velocità tra dispositivi digitali.
Lo standard RS232 nacque nei primi anni '60 per opera della *Electronic Industries Association* (EIA) ed era orientato alla comunicazione tra i *mainframe* e i terminali attraverso la linea telefonica, utilizzando un *modem*: esso includeva le caratteristiche elettriche dei segnali, la struttura e temporizzazioni dei dati seriali, la definizione dei segnali e dei protocolli per il controllo del flusso di dati seriali su un canale telefonico, il connettore e la disposizione dei suoi *pin*, ed infine il tipo e la lunghezza massima dei possibili cavi di collegamento.
Nel corso degli ultimi 50 anni lo standard si è evoluto pur mantenendosi in larga parte invariato: l'ultima revisione è del 1997 (RS232f), ma la versione più diffusa è probabilmente la RS232c del 1969 [68].
Pur essendo uno standard piuttosto vecchio, attualmente l'interfaccia RS232 è ancora largamente utilizzata per la comunicazione a bassa velocità tra microcontrollori, dispositivi industriali ed altri circuiti relativamente semplici che non necessitano di particolare velocità.
L'interfaccia RS232 utilizza un protocollo seriale asincrono, il segnale elettrico è non bilanciato, la trasmissione è binaria, ed infine il collegamento è di tipo *point-to-point* (un solo trasmettitore e un solo ricevitore).

4.9.1.1 Caratteristiche di un segnale RS-232

La cosa più semplice per descrivere un segnale RS-232 è partire con un esempio. In fig. 4.117 è visualizzato, in modo idealizzato, un segnale RS-232 (del tipo "8n2") rappresentante il valore binario 00110000. Si può notare innanzitutto che nello standard RS-232 un segnale alto (+12 V) rappresenta lo 0 logico, mentre uno basso (-12 V) rappresenta un 1 logico.

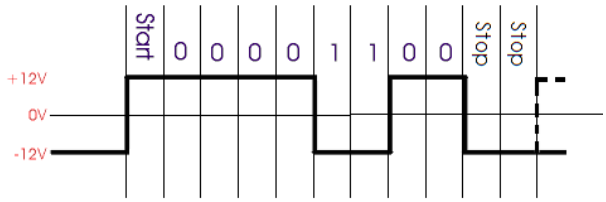


Fig. 4.117 *Esempio di segnale RS-232*

La linea si trova inizialmente nello stato di riposo, bassa (nessun dato in transito). La prima transizione da basso in alto indica l'inizio della trasmissione (1 *bit* di *start*); segue il *bit* meno significativo (LSB), poi il secondo *bit*, e così via fino al *bit* più significativo (MSB). Segue infine un periodo di riposo della linea di 2 *bit* di *stop* e quindi (eventualmente) inizia un nuovo pacchetto di dati con un nuovo *bit* di *start* (linea tratteggiata). C'è da notare che il *byte* è trasmesso al contrario, cioè va letto da destra verso sinistra.
Si possono trasmettere 6, 7, 8 o anche 9 *bit* di dati (ma quest'ultima possibilità non è prevista dalle porte seriali dei normali PC) seguiti eventualmente da un ultimo *bit*, il *bit* di parità, per verificare la correttezza del dato ricevuto.
In genere il formato del pacchetto trasmesso è indicato da una sigla composta da numeri e cifre, per esempio "8n1" e "7e2" (nel caso di ECSIC32 il formato scelto è "8n1"):

- La prima cifra indica quanti *bit* di dati sono trasmessi (nei due esempi rispettivamente 8 e 7)
- La lettera indica il tipo di parità: "n" nessuna, "e" parità pari e "o" parità dispari
- La seconda cifra indica il numero di *bit* di *stop* (rispettivamente 1 e 2)

Lo standard originale prevede una velocità di trasmissione fino a 20 k*bit*/s. Uno standard successivo (RS-562) ha portato il limite a 64 k*bit*/s, lasciando gli altri parametri elettrici praticamente invariati, e rendendo quindi i due standard compatibili a bassa velocità. Nei normali PC le interfacce seriali RS-232 arrivano in genere almeno a 115 k*bit*/s, 230 k*bit*/s o anche più: nel caso del sistema ECSIC32, è stata verificata con successo la trasmissione/ricezione di dati a una velocità di 460 k*bit*/s (più precisamente 460800 *bit*/s).
E' importante garantire il rigoroso rispetto della durata dei singoli *bit*: non è infatti presente alcun segnale di *clock* comune a trasmettitore e ricevitore, e l'unico elemento di sincronizzazione è dato dal fronte di salita del *bit* di *start*. Come linea guida, occorre considerare che il campionamento in ricezione è effettuato di norma al centro di ciascun *bit*: l'errore massimo ammesso è quindi, teoricamente, pari alla durata di mezzo *bit* (circa il 5% della frequenza di *clock*), il che impone di utilizzare *clock* con una precisione migliore dell'1% (come gli oscillatori a quarzo).

4.9.1.2 Parametri elettrici della RS-232 e compatibilità TTL

La tensione di uscita da un trasmettitore RS-232 deve essere compresa in valore assoluto tra 5 V e 25 V (quest'ultimo valore è stato ridotto a 13 V in alcune revisioni dello

standard), mentre il ricevitore deve funzionare correttamente con tensioni di ingresso comprese, sempre in modulo, tra i 3 V ed i 25 V.
In genere, però, i segnali utilizzati dai sistemi digitali sono "TTL compatibili", cioè variano tra 0 e 5 V, oppure tra 0 V e 3.3 V: non sono quindi direttamente compatibili con la standard RS-232. In commercio esistono comunque appositi traslatori di livello, che hanno il compito di fornire sia in trasmissione che in ricezione gli opportuni livelli (non modificando la forma del segnale trasmesso).
Il traslatore di livello utilizzato per ECSIC32 è l'integrato TRS232E della Texas Instruments: è un circuito integrato che permette la compatibilità tra logiche TTL (o CMOS) a 5 V e lo standard RS-232, partendo da una sola alimentazione a 5 V; opera fino a 250 k*bit*/s, ed è caratterizzato da una bassa corrente di alimentazione (8 mA). La sezione ricevente è costituita da due *inverter* che accettano in ingresso una tensione compatibile allo standard RS-232 (es. ± 12 V, ma possono accettare fino a ± 30 V) ed in uscita presentano un segnale TTL compatibile (0 ÷ 5 V). La sezione trasmittente ha due *driver* invertenti con un in ingresso TTL compatibile (0 ÷ 5 V) e capaci di erogare a vuoto una tensione di ± 7 V (valore tipico), compatibile con lo standard RS-232.
Per ricavare le tensioni positive e negative necessarie per garantire i livelli richiesti dalla RS-232, viene usata una configurazione *a pompa di carica*, costituita dall'integrato TRS232E [69] e da quattro condensatori esterni da 1 µF (oppure da 0.1 µF), come si può vedere in fig. 4.118: la tensione di alimentazione viene duplicata per ottenere il livello logico basso (+ 7 V) compatibile con lo standard RS-232, mentre per ottenere il livello logico alto (-7 V) viene dapprima duplicata e poi invertita.

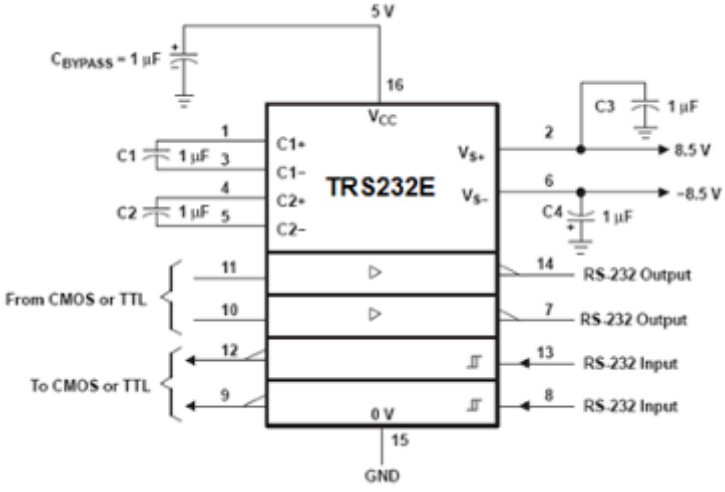


Fig. 4.118 *Configurazione a pompa di carica per la compatibilità TTL*

4.9.1.3 Connettori RS-232

Nei PC sono disponibili due tipi di connettori RS-232: DB9 (9 *pin*, più recenti) e DB25 (25 *pin*), entrambi maschi e praticamente identici dal punto di vista funzionale (fig. 4.119). Oggi, in ogni caso, lo standard RS-232 tende ad essere utilizzato solo in ambito industriale e di laboratorio, per cui nei PC è stato sostituito dallo standard USB.

Fig. 4.119 *Tipi di connettori RS-232*

Di seguito viene riportata una tabella in cui sono indicati i nomi dei segnali, la loro sigla, il numero del *pin* corrispondente sul connettore RS-232 e la direzione del segnale (O = uscita dal PC).
In teoria (e spesso anche in pratica, come nel caso del sistema ECSIC32), per ricevere e trasmettere un segnale RS-232 bastano tre fili: ricezione, trasmissione e massa. Gli altri fili servono per l'*handshake* tra il PC e una periferica (o tra due PC), ovvero per sincronizzare in *hardware* la comunicazione.

Sigla	**25 *pin***	**9 *pin***	**In/Out**	**Nome**
TxD o Tx	2	3	O	*Transmitted Data*
RxD o Rx	3	2	I	*Received Data*
RTS	4	7	O	*Request To Send*
CTS	5	8	I	*Clear To Send*
DTR	20	4	O	*Data Terminal Ready*
DSR	6	6	I	*Data Set Ready*
RI	22	9	I	*Ring Indicator*
DCD	8	1	I	*Data Carrier Detect*
GND	7	5	-	*Ground (massa)*

Tab. 4.4 *Segnali RS-232 e pin corrispondenti sui connettori*

4.9.2 Generalità sui dispositivi UART

L'UART (*Universal Asynchronous Receiver/Transmitter*) è un dispositivo *hardware* in grado di convertire flussi di *bit* di dati da un formato parallelo a un formato seriale asincrono o viceversa, ed è per questo comunemente utilizzato per gestire le comunicazioni delle interfacce seriali RS-232. Il termine "universale" sta ad indicare che il formato dei dati e la velocità di trasmissione sono configurabili.
Un dispositivo UART (nel nostro caso interno al μC) prende interi *byte* di dati e trasmette individualmente i *bit* in modo seriale: a destinazione (internamente al PC) un secondo UART riassembla i *bit* in *byte* completi. Ogni UART contiene un registro a scorrimento per permettere la conversione fra i formati seriale e parallelo.
L'UART, generalmente, non trasmette (o riceve) i segnali esterni in modo diretto, ma fa utilizzo di dispositivi d'interfacciamento che rendono i segnali esterni compatibili con la

logica dell'UART (es. TTL). I segnali esterni possono essere di tipologie diverse: alcuni esempi di standard per segnali in tensione sono l'RS-232, l'RS-422 e l'RS-485.
La trasmissione asincrona permette di trasmettere i dati senza che il ricevitore abbia bisogno di un *clock* inviato dal trasmettitore: la sincronizzazione fra unità trasmittente e unità ricevente avviene attraverso l'invio di *bit* speciali aggiunti al *byte* di dati, e attraverso stringenti specifiche temporali che devono essere soddisfatte di comune accordo.
Quando un *byte* di dati viene fornito all'UART per la trasmissione asincrona, un *bit* di *start* viene aggiunto all'inizio della parola da trasmettere: tale *bit* è usato per avvertire il ricevitore che sta per arrivare un *byte* di dati, e per forzare il *clock* del ricevitore a sincronizzarsi con il *clock* del trasmettitore. Dopo il *bit* di *start*, vengono inviati i singoli *bit* del dato, con il *bit* meno significativo (LSB) inviato per primo. Ogni *bit* viene trasmesso per un intervallo di tempo esattamente uguale a quello riservato agli altri *bit* (*data rate*): il ricevitore, più o meno a metà del periodo di tempo riservato alla trasmissione di un *bit*, determina se il valore del *bit* è 1 o 0, dopo di che attende un periodo di *data rate* e passa a valutare un altro *bit*, e così via.
Quando l'intero *byte* di dati è stato inviato, il trasmettitore può aggiungere un *bit* di *parità* (che può essere usato dal ricevitore per verificare la presenza di errori) e deve inviare almeno un *bit* di *stop*.
Quando il ricevitore ha ricevuto il *byte* di dati, cercherà l'eventuale *bit* di parità (ci deve essere un accordo preventivo fra trasmettitore e ricevitore sull'utilizzo o meno del *bit* di parità) e il *bit* di *stop*: se il *bit* di *stop* non arriva, il ricevitore UART considera non valida l'intera parola, e riporterà un errore al suo μP di riferimento (*framing error*). In ogni caso, sia che l'intero pacchetto di *bit* sia stato ricevuto correttamente oppure no, il ricevitore UART scarta automaticamente i *bit* di *start*, parità e *stop*.
Se un'altra parola è pronta per la trasmissione, il *bit* di *start* della nuova parola può essere inviato subito dopo l'ultimo *bit* di *stop* della parola precedente.
Per una dettagliata descrizione della periferica UART integrata nel μC LPC2103 si rimanda al par. 4.4.4.6.

4.10 Descrizione del *layout*

In questo paragrafo verranno mostrati alcuni *layout* del sistema ECSIC32, ed i relativi schemi circuitali. Tutti i *layout* sono stati progettati con il programma OrCAD *Layout* 10.5.
Il primo esempio che viene mostrato, in fig. 4.120, è il *layout* dello strato superiore (*top*) della scheda principale (*main board*), in cui sono alloggiati: una scheda Olimex LPC-P2103 [70] (sulla quale sono montati il microcontrollore LPC2103 e i quarzi), le due CPLD per la gestione del banco di DDC112, la rete di alimentazione per l'intero sistema, i sistemi d'interfacciamento con la tastiera a matrice (comparatore e rete resistiva) e con il PC (traslatore di livello in configurazione a pompa di carica), e diversi connettori per future espansioni a più di 32 canali e per i collegamenti con il banco di memoria.

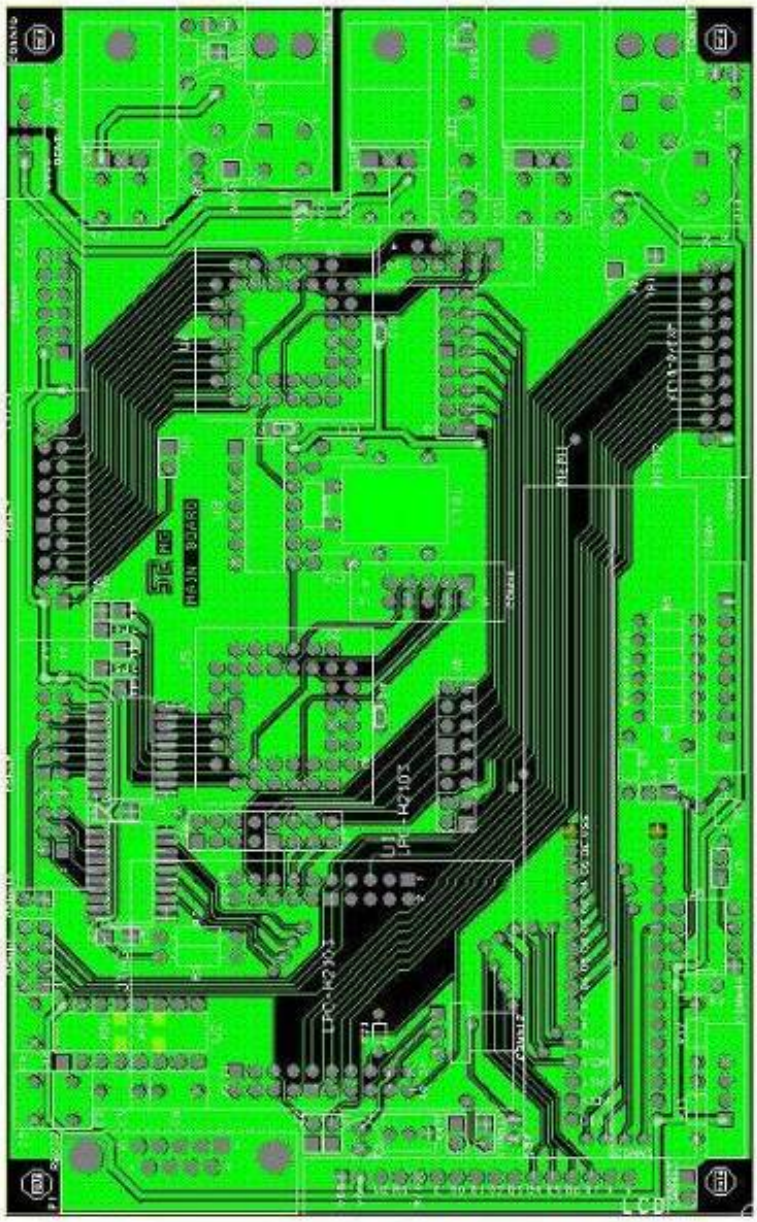
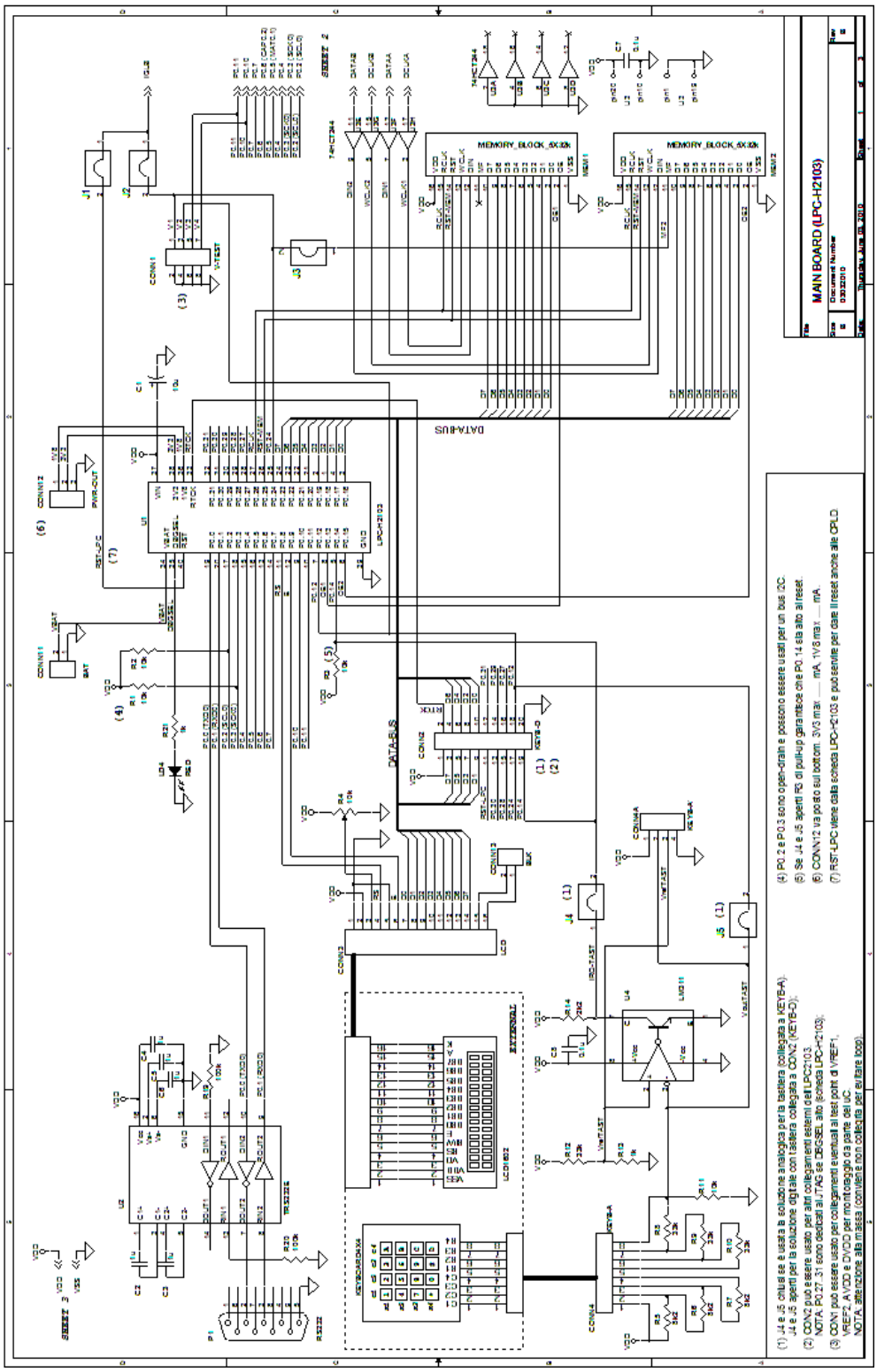
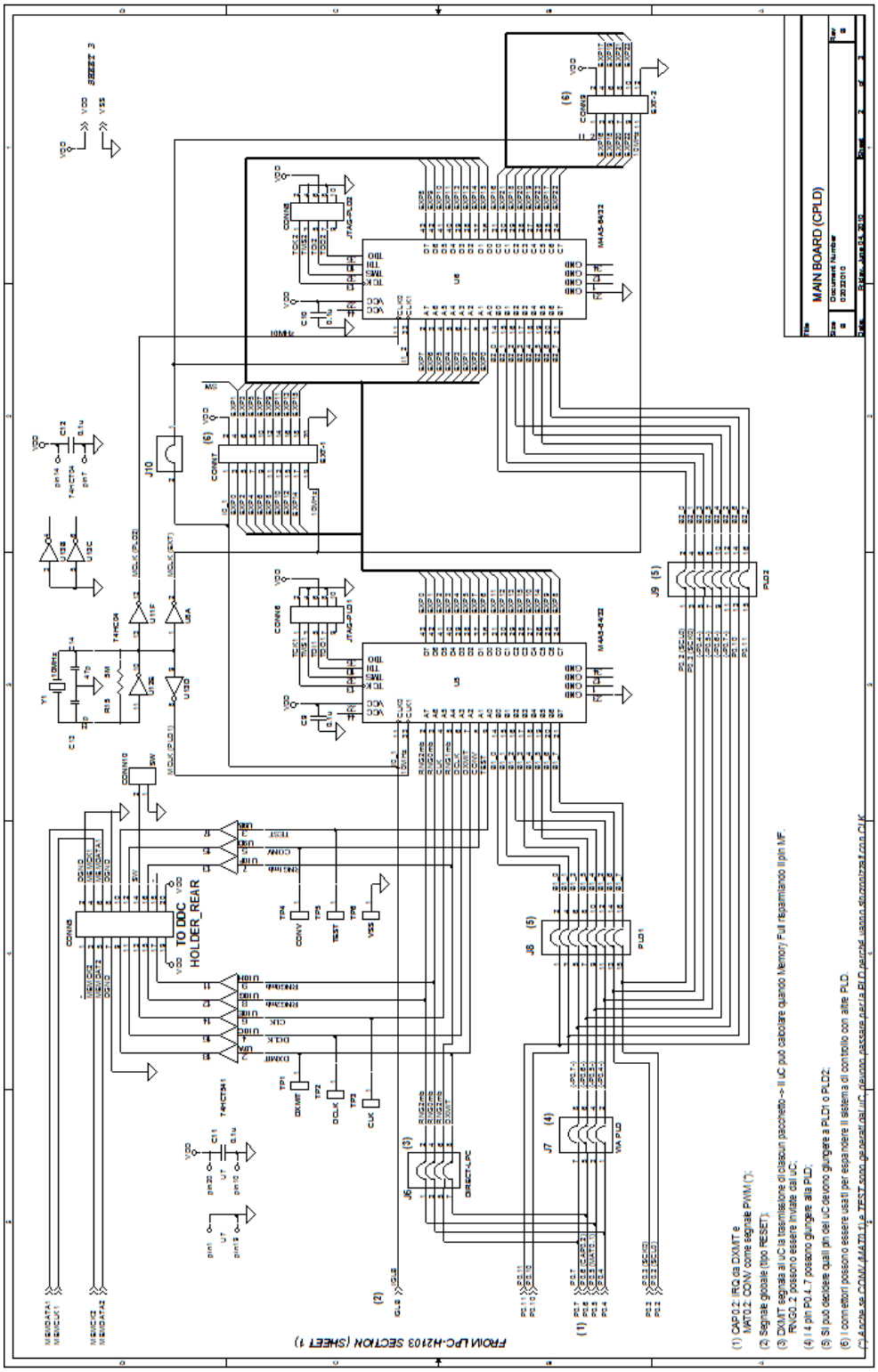
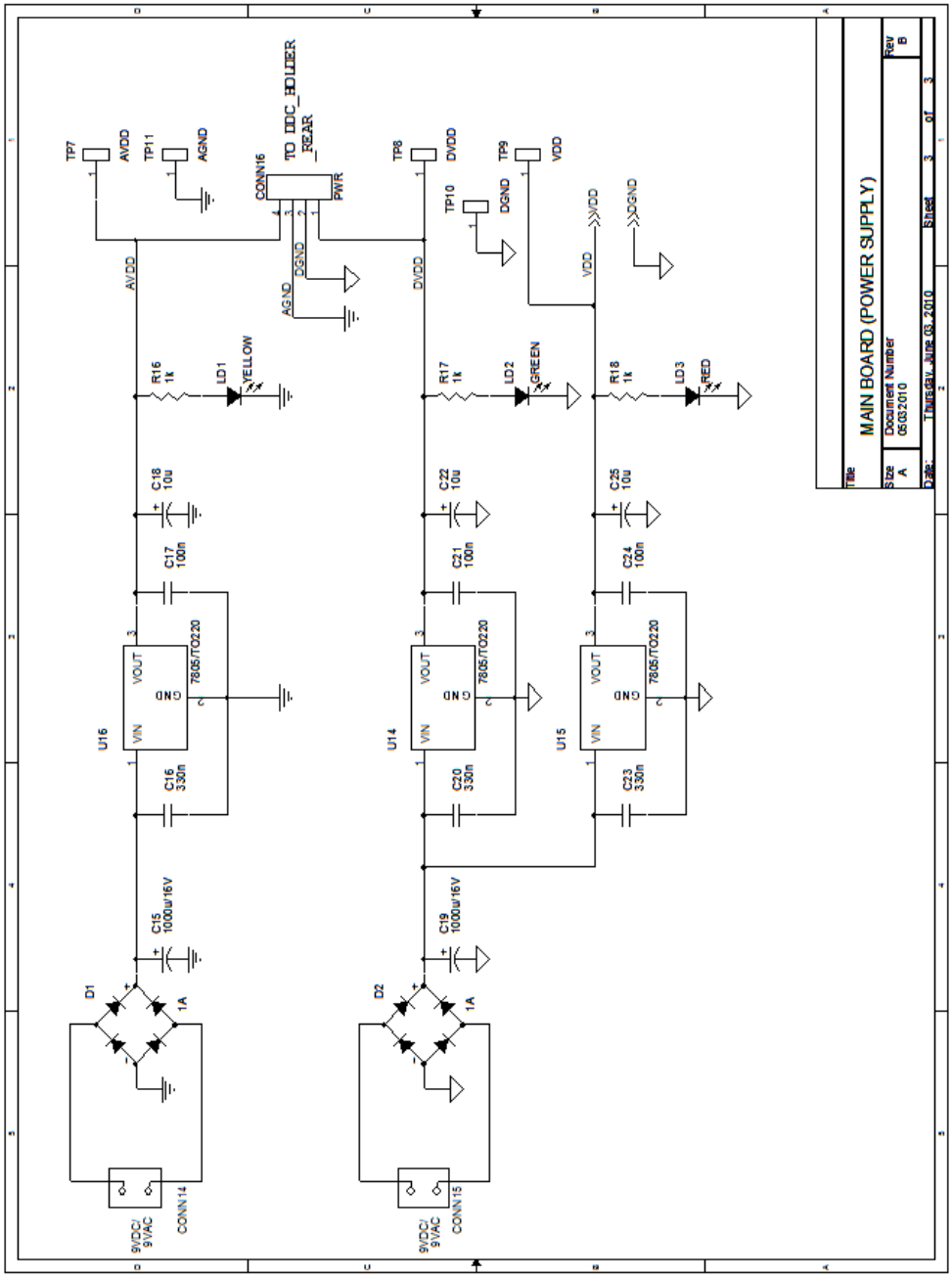


Fig. 4.120 *Layout della main board di ECSIC32*

Nelle pagine successive sono riportati gli schemi circuitali relativi alle 3 diverse parti funzionali *della main board*, nell'ordine: quella relativa al microcontrollore, la parte relativa alle CPLD per la gestione del banco dei DDC112, e lo schema di alimentazione (*power supply*) .


MAIN BOARD (LPC-H2103)
(1) J4 e J5 chiusi se è usata la soluzione analogica per la tastiera (collegata a KEYB-A).
J4 e J5 aperti per la soluzione digitale con tastiera collegata a CON2 (KEYB-D);
(2) CON2 può essere usato per altri collegamenti esterni dell LPC2103.
NOTA: P0.27..31 sono dedicati al JTAG se DBGSEL alto (scheda LPC-H2103);
(3) CON1 può essere usato per collegamenti eventuali al test point di VREF1,
VREF2, AVDD e DVDD per monitoraggio da parte del uC.
NOTA: attenzione alla massa (conviene non collegata per evitare loop).
(4) P0.2 e P0.3 sono open-drain e possono essere usati per un bus I2C.
(5) Se J4 e J5 aperti R3 di pull-up garantisce che P0.14 sia alto al reset.
(6) CONN12 va posto sul bottom. 3V3 max ___ mA, 1V8 max ___ mA.
(7) RST-LPC viene dalla scheda LPC-H2103 e può servire per dare il reset anche alle CPLD.
DATA-BUS
MEMORY_BLOCK_6X32k
LPC-H2103
EXTERNAL
LCD1602
KEYBCARDAX4
SHEET 2
SHEET 3


MAIN BOARD (CPLD)
TO DDC HOLDER_REAR
FROM LPC-H2103 SECTION (SHEET 1)
(1) CAP0.2: IRQ da DXMIT e
MAT0.2: CONV come segnale PWM(*);
(2) Segnale globale (tipo RESET);
(3) DXMIT segnala al uC la trasmissione di ciascun pacchetto --> il uC può caricare quando Memory Full ripartimando il pin MF.
RXD0..2 possono essere inviate dal uC;
(4) I 4 pin P0.4..7 possono giungere alla PLD;
(5) Si può decidere quali pin dei uC devono giungere a PLD1 o PLD2;
(6) I connettori possono essere usati per espandere il sistema di controllo con altre PLD.
(*) Anche se CONV/MAT0.1 e TEST sono generati dal uC, devono passare per la PLD perché vanno sincronizzati con CLK


MAIN BOARD (POWER SUPPLY)
Document Number
05032010
Rev
B
Size
A
Date:
Thursday, June 03, 2010
Sheet
3
of
3
TP7
AVDD
TP11
AGND
CONN16
TO DDC_HOLDER
_REAR
PWR
TP8
DVDD
TP9
VDD
TP10
DGND
AVDD
DVDD
VDD
DGND
AGND
R16
1k
LD1
YELLOW
R17
1k
LD2
GREEN
R18
1k
LD3
RED
C18
10u
C17
100n
C22
10u
C21
100n
C25
10u
C24
100n
U16
U14
U15
7805/TO220
VIN
GND
VOUT
C16
330n
C20
330n
C23
330n
C15
1000u/16V
C19
1000u/16V
D1
D2
1A
9VDC/
9VAC
CONN14
CONN15

In fig. 4.121 è riportato il *layout* della scheda destinata al banco di memoria (*memory board*): su di essa sono alloggiati i 5 *chip* di memoria e la logica di controllo CPLD per la gestione dell'intero banco, nonché i connettori per il collegamento con la main board e con il banco di DDC112. Lo schema circuitale del blocco di memoria è già stato mostrato nel par. 4.6 (fig. 4.89)

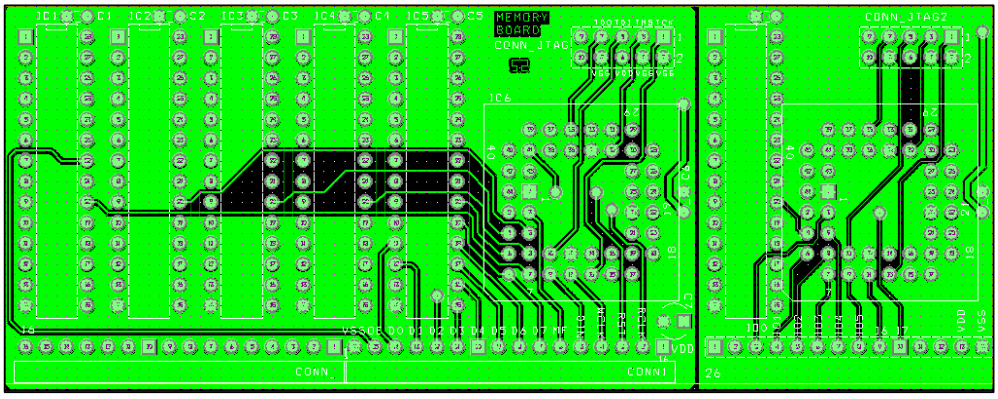


Fig. 4.121 *Layout della memory board*

In fig. 4.122 è riportato il *layout* di una delle 4 schede ECSIC8 (elettronica di *front-end* a 8 canali) utilizzate nel sistema ECSIC32 a 32 canali: tutte le schede sono montate verticalmente su un unico *backplane* comune (vedi par. 4.1.3), realizzando una struttura "a pettine". E'stata prevista anche la realizzazione di un secondo *backplane* (anch'esso per 4 schede ECSIC8) per una futura espansione a 64 canali: i due pettini a 32 canali (*backplane* + 4 schede ECSIC 8) si andranno ad incastrare l'uno nell'altro, conferendo grande compattezza all'intera struttura (l'ingombro è pressoché identico per 32 o 64 canali).

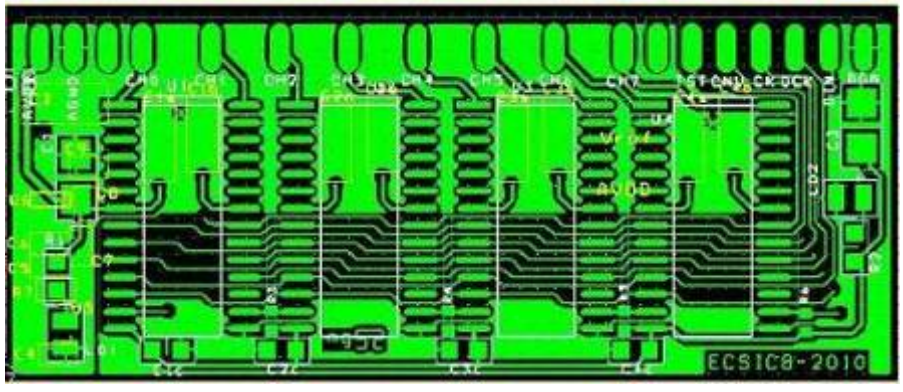


Fig. 4.122 *Layout della scheda ECSIC8*

In fig. 4.123 è riportato lo schema circuitale della scheda ECSIC8 con i 4 integrati DDC112:

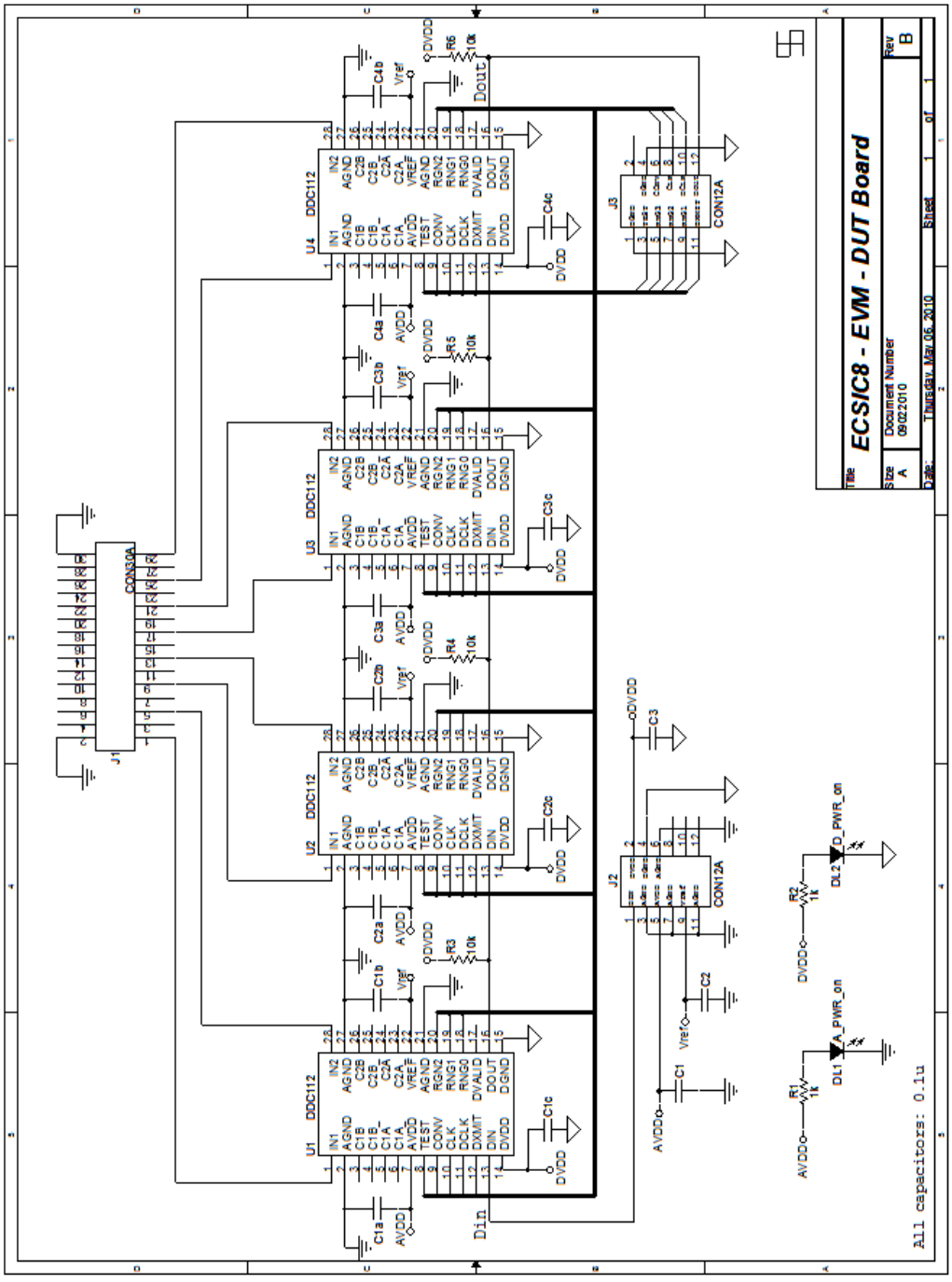


Fig. 4.123 *Schema circuitale della scheda ECSIC8*

In fig. 4.124, è riportato il *layout* della scheda denominata *backplane rear*: su di essa sono alloggiati i *buffer* per il trasferimento di segnali digitali da (e verso) il banco dei DDC112 (sezione digitale della scheda), ed inoltre su di essa è stato realizzato il circuito per la generazione della tensione di riferimento necessaria al funzionamento dei DDC112 (sezione analogica della scheda).

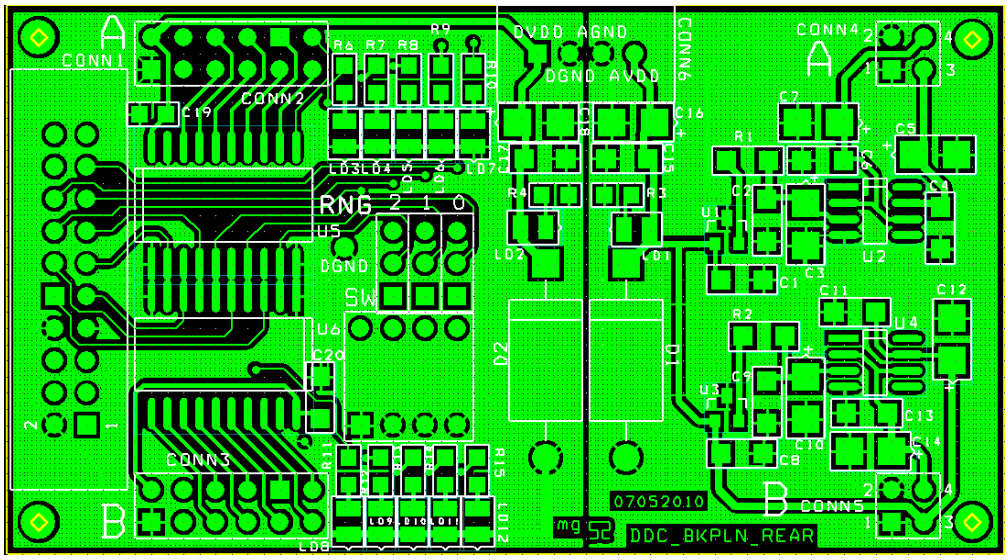


Fig. 4.124 *Layout della scheda 'backplane rear'*

Ad ogni modo, ognuna delle schede ECSIC8 prevede la possibilità di implementare il proprio circuito per la generazione del riferimento di tensione, allo scopo di limitare il carico dell'amplificatore OPA350 (vedi fig. 4.125) e diminuire il rumore [71].

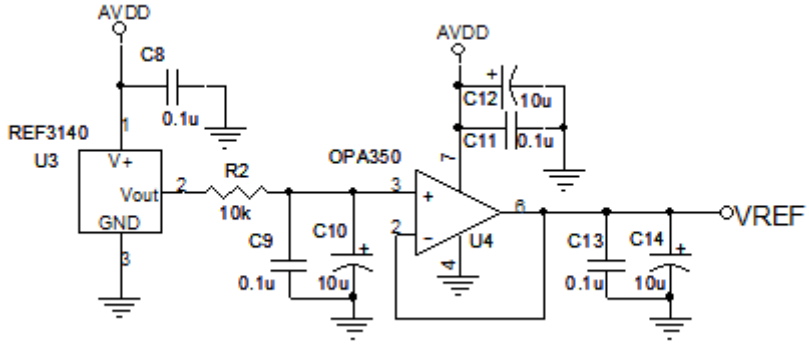


Fig. 4.125 *Schema circuitale del riferimento di tensione*

Infine, in fig. 4.126, è riportata una panoramica complessiva del sistema ECSIC32. In primo piano si può notare il sensore in diamante montato sul suo *sample holder*: immediatamente dietro al sensore c'è tutta l'elettronica di condizionamento del segnale, costituita dal banco dei DDC112 (4 schede ECSIC8 montate su unico *backplane*). Un

bus connette poi la scheda *backplane rear* dell'elettronica di condizionamento alla *main board*, dove risiede l'elettronica di elaborazione: sono evidenti in modo particolare la scheda del microcontrollore (dov'è visibile un LED acceso) e la scheda di memoria, montata verticalmente. Infine, sono ben visibili la tastiera per le impostazioni dell'utente esterno e il *display* LCD su cui compaiono le informazioni volute relative alla sessione di misura.

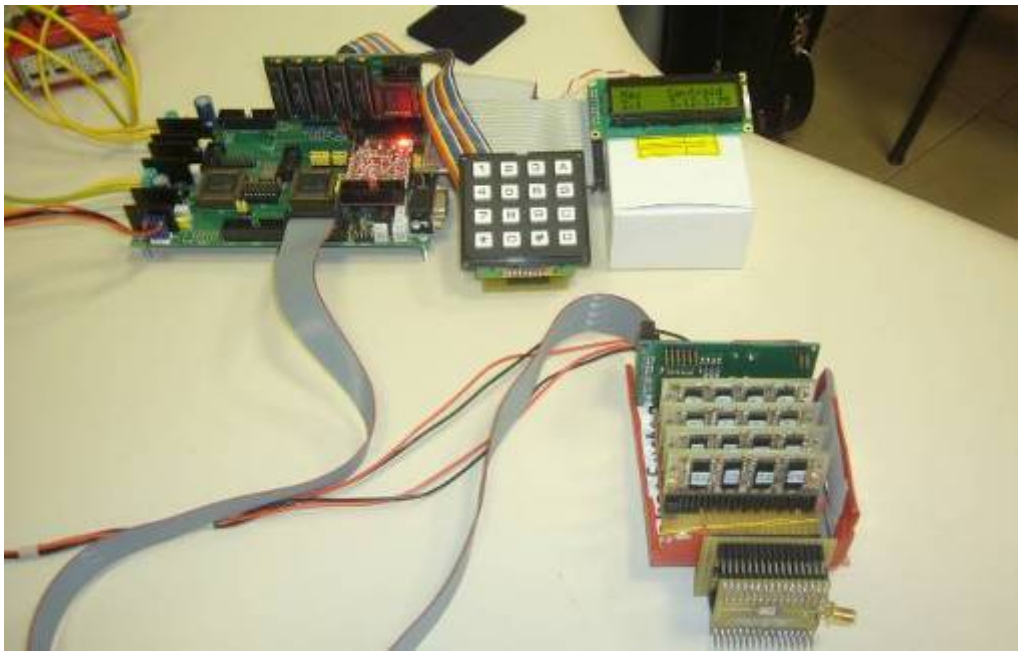


Fig. 4.126 *Il sistema ECSIC32*

Ovviamente, in tutte le fasi di acquisizione di profili, in particolar modo di sorgenti X (potenzialmente dannose per i componenti elettronici), è stata prevista una schermatura metallica per l'elettronica di *front-end*, utile anche come isolamento da fonti esterne di rumore.

4.11 Procedure di *test* e collaudo dell'elettronica

Alla fine del processo di realizzazione del prototipo, sono state compiute una serie di verifiche preliminari, le più importanti delle quali verranno brevemente descritte in questo paragrafo.
Una primo essenziale *test* è stato verificare i *pin* di alimentazione e di massa di tutti gli integrati e di tutti i connettori presenti sulle diverse schede del sistema (*main board*, *memory board*, etc.), per scongiurare la presenza di errori nelle piste di connessione (corti, circuiti aperti, etc.).

E'stato inoltre verificato con un oscilloscopio il funzionamento del generatore del *master clock* (un multivibratore astabile realizzato con un quarzo da 10 MHz e con l'integrato SN74HC04 [72]): la forma d'onda generata è correttamente un'onda quadra con *duty cycle* 50% e livelli 0 ÷ 5 V.
Passando alle logiche CPLD, per verificare che tutti i *pin* fossero in grado di trasferire e ricevere segnali, ognuna delle CPLD è stata preliminarmente programmata per agire come un contatore a 16 *bit*, con un *clock* fornito esternamente dal generatore di *clock* testato in precedenza. Con un oscilloscopio si è poi monitorata l'uscita dei 32 *pin* I/O di ogni CPLD: il corretto funzionamento implica la presenza in uscita dai 32 *pin* I/O di 16 onde quadre (ognuna di frequenza pari alla metà della frequenza dell'onda precedente) e delle loro negate. Il funzionamento del *pin* globale CLK0 è stato verificato utilizzandolo come *reset* per il contatore a 16 *bit* programmato.
Per quel che riguarda il collaudo del microcontrollore LPC2103 e dei dispositivi da esso gestiti, sono stati elaborati (e poi eseguiti), una serie di programmi di *test*, qui di seguito elencati e brevemente descritti:

- **Test1**: per verificare il funzionamento della schedina Olimex LPC-P2103 e la programmabilità del microcontrollore, il LED rosso sulla schedina si accende e si spegne finche viene premuto il tasto presente sulla schedina stessa.
- **Test2**: per scongiurare la presenza di corti e verificare la "visibilità" di tutti i *pin* GPIO, le uscite GPIO del microcontrollore passano da 0 a 1 con frequenza 2 Hz.
- **Test3LCD**: per la verifica del corretto funzionamento del *display* LCD, vengono scritti su di esso una serie di caratteri e stringhe.
- **Test4KEY**: per verificare il corretto funzionamento della tastiera, viene letta l'uscita dell'ADC interno al microcontrollore e compare sul *display* LCD (a ciclo continuo) il nome del tasto premuto.
- **Test5KEYRQ**: come Test4KEY, ma stavolta il microcontrollore legge l'uscita dell'ADC su richiesta, ovvero quando riceve un *interrupt request* da parte del comparatore LM311 (presente in uscita alla tastiera).
- **Test6RNG**: per verificare che, quando si specifica da tastiera un numero compreso tra 1 e 7 (corrispondente ai 3 *bit* che specificano il *range* di carica in ingresso al DDC112), la scelta effettuata sia presente sui *pin* RNG 0,1,2 della *main board*. Inoltre il valore di *range* scelto appare sul *display* LCD.
- **Test7CONVPLD1**: per la verifica del corretto andamento dei segnali CONV e TEST necessari per il banco dei DDC112, questo programma permette di selezionare un particolare tempo d'integrazione (per il CONV) e un certo numero di pacchetti di carica (per il TEST). I segnali CONV e TEST vengono poi visualizzati con un oscilloscopio.
- **Test8CONV2**: è un programma analogo a Test7CONVPLD1, ma con l'aggiunta della gestione da tastiera delle informazioni riguardanti il tempo d'integrazione e i pacchetti di carica.
- **Test9MEMORY** e **Test10MEMORY**: sono due programmi per verificare il corretto funzionamento del banco di memoria (*test* di scrittura, lettura, *reset* e inizializzazione). Sono stati utilizzati anche per la programmazione definitiva della logica CPLD per la gestione del banco di memoria.

- **Test11ECSIC8**: dopo aver montato le schede ECSIC8 sul *backplane*, questo programma è stato utilizzato per testare una per una, attraverso una serie di acquisizioni e memorizzazioni, le 4 schede dell'elettronica a 8 canali, allo scopo di correggere eventuali errori di saldatura.
- **Test12ECSICinputAcq**: è un programma analogo a Test11ECSIC8, ma con l'aggiunta della visualizzazione dei risultati sul *display* LCD.
- **Test13ECSICinputAcqPC**: è un programma analogo a Test12ECSICinputAcq, in cui è stata aggiunta la verifica del corretto trasferimento dei dati al PC tramite interfacciamento UART/RS-232.
- **Test14**: è il programma finale (*main.c*) contenente il listato definitivo per il microcontrollore. In esso è contenuta la gestione del *menu* con il quale, da tastiera, è possibile impostare il tempo d'integrazione T_{INT}, la modalità di funzionamento del banco dei DDC112 (normale o "*test*"), il *range* di carica e infine il numero complessivo di acquisizioni per ciclo. E' incluso anche il *software* di calcolo per valore atteso, valore massimo e centroide, nonché il *software* per la gestione del trasferimento dei dati al PC.

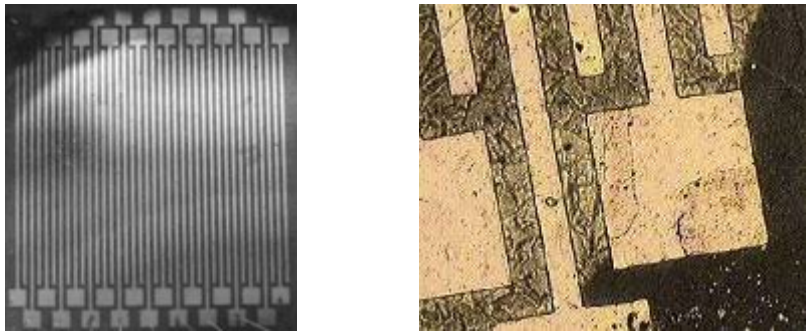
Capitolo V

Risultati sperimentali per rivelatori multipixel

5.1 Rivelatori multipixel 1D e 2D

In seguito alla caratterizzazione di rivelatori a singolo pixel e alla progettazione di un'elettronica di lettura multicanale (sistema ECSIC), si è proceduto alla realizzazione di rivelatori multipixel 1D e 2D. I passi di processo sono del tutti analoghi a quelli presentati nel par. 3.3 per i rivelatori a singolo pixel, a parte ovviamente il *layout* delle maschere. Per entrambi i tipi di rivelatori si è scelta una configurazione *sandwich*, per i vantaggi che essa presenta rispetto alla struttura planare (vedi par. 3.4.2). Le strutture realizzate, tutte con contatti metallici in argento di spessore 200 nm, sono elencate qui di seguito:

- Rivelatore 1D a *strip* (fig. 5.1) costituito da 32 strisce di larghezza 80 μm e lunghezza 6 mm, spaziate di 80 μm: è stato utilizzato come PSD (*Position Sensitive Detector*) 1D, e per ottenere il profilo monodimensionale di una sorgente UV di debole intensità (lampada Xe-Hg accoppiata ad un monocromatore).



(a) **(b)**

Fig. 5.1 *Rivelatore 1D a strip (a) e particolare al microscopio ottico della struttura dei contatti (b)*

- Rivelatore 2D a 64 pixel (8 x 8), ciascuno di area 500 x 500 μm^2, con spaziatura orizzontale e verticale 200 μm (fig. 5.2): è stato utilizzato per effettuare una

caratterizzazione preliminare a 8 canali (ECSIC8), emulando il profilo 2D di una sorgente UV di debole intensità (vedi par. 5.2.2). E'stato progettato in origine per essere accoppiato ad un'elettronica a 64 canali, poi difficoltà connesse al *wire bonding* dei contatti hanno determinato il passaggio ad una struttura a 36 pixel.

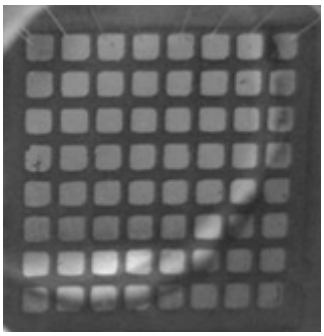


Fig. 5.2 *Rivelatore 2D a 64 pixel: da notare la connessione dei soli pixel della prima fila*

- Rivelatore 2D "*leaf-shaped*" (fig. 5.3), costituito da 32 strisce di larghezza 96 μm e lunghezza variabile, con spaziatura 600 μm: è stato utilizzato come sensore 2D di posizione (PSD). Il *design* "a foglia" della struttura superiore dei contatti è stato progettato per agevolare la connessione con l'elettronica di *front-end*, pur mantenendo un'area attiva del dispositivo di dimensioni adeguate.

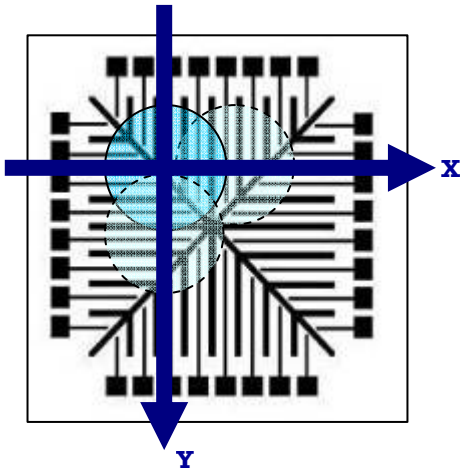
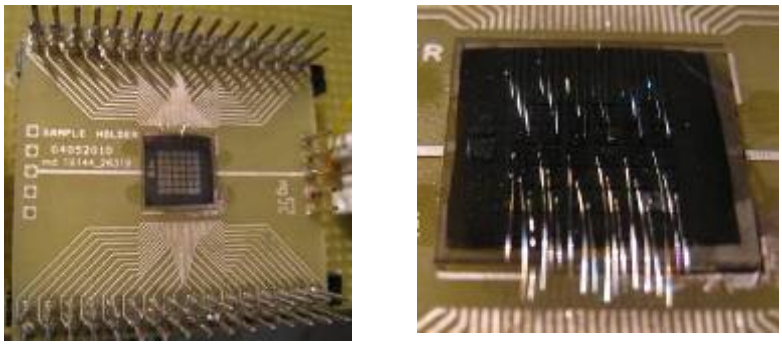


Fig. 5.3 *Rivelatore 2D "leaf-shaped" utilizzato come PSD*

- Rivelatore 2D a 36 pixel (6 x 6), ciascuno di area 750 x 750 μm^2, con spaziatura orizzontale e verticale 150 μm. E' la struttura definitiva utilizzata per l'analisi di profili 2D di sorgenti UV di debole intensità, di laser a eccimeri, e infine di sorgenti X. Il rivelatore è stato montato su un *sample holder* appositamente progettato (fig. 5.4a), e successivamente connesso tramite *wire bonding* (fig.

5.4b) alle linee di acquisizione del segnale: per poter utilizzare un'elettronica a 32 canali (ECSIC32), i 4 pixel alle estremità dell'area attiva risultano non connessi.



(a) **(b)**

Fig. 5.4 *Rivelatore 2D a 36 pixel: montaggio sul sample holder (a) e wire bonding dei contatti (b)*

In fig. 5.5 è riportato il principio di funzionamento dei rivelatori, indipendentemente dalla tipologia (1D o 2D) e dalla geometria dei contatti. Un contatto d'argento di spessore 50 nm e dimensioni pari all'area attiva del rivelatore, riceve i fotoni in arrivo: il piccolo spessore rende semitrasparente il contatto (caratteristica importante in particolar modo per le radiazioni UV piuttosto che X, che hanno una maggior capacità di penetrazione), e dunque i fotoni sono in grado di interagire con il materiale *bulk* del rivelatore, ovvero il diamante. L'energia dei fotoni in arrivo viene assorbita dando luogo alla creazione di coppie elettrone-lacuna, poi raccolte dagli elettrodi in seguito all'applicazione di una tensione di polarizzazione V_{bias}.

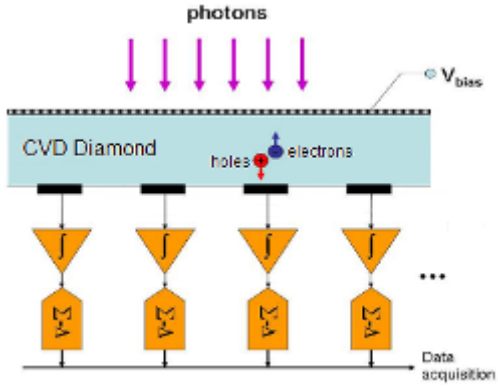


Fig. 5.5 *Principio di funzionamento dei rivelatori multipixel realizzati*

Il segnale in corrente che ne deriva viene infine condizionato dall'elettronica di *front-end* e acquisito: come ampiamente descritto nel cap. IV, ogni pixel è connesso ad una sua catena di misura dedicata, costituita da un integratore di precisione e da un convertitore analogico/digitale Σ-Δ.

5.2 Caratterizzazione preliminare a 8 canali

Prima di effettuare le misure sperimentali con ECSIC32 (elettronica di lettura a 32 canali), è stata compiuta una verifica preliminare di funzionamento dell'elettronica di lettura a 8 canali (ECSIC8): a tale scopo, per ottimizzare i tempi, sono comunque stati utilizzati i rivelatori realizzati per l'elettronica a 32 (o 64) canali, prelevando da essi i segnali provenienti da 8 pixel (o *strip*) opportunamente scelti a seconda dei casi.
Le misure sperimentali compiute in questa fase riguardano:

- Valutazione dei rivelatori come sensori di posizione 1D e 2D per radiazioni UV
- Emulazione 1D e 2D a 32 (e 64) canali di profili di sorgenti UV di bassa intensità

Il *setup* sperimentale prevede in entrambi i casi l'utilizzo di una lampada UV (Xe-Hg) accoppiata ad un monocromatore (vedi par. 3.5.1, in particolare la fig. 3.9): per effettuare le misure, è stata selezionata una radiazione monocromatica avente lunghezza d'onda λ = 235 nm e potenza incidente $P_{INC} \sim 1$ μW, con uno *spot-size* sul rivelatore di dimensioni 5 x 5 mm^2.

5.2.1 Sensori di posizione

Per effettuare misure di posizione del fascio UV selezionato, i rivelatori sono stati montati su un sistema di movimentazione micrometrica X-Y avente una risoluzione di 10 μm. Una prima verifica è stata compiuta utilizzando il rivelatore 1D a 32 *strip* (fig. 5.1).

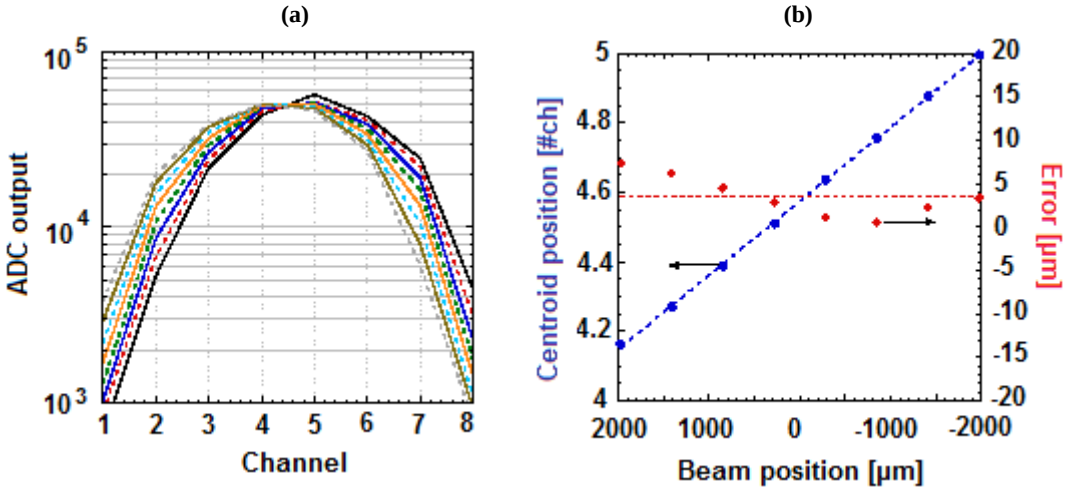


Fig. 5.6 *1D PSD: 8 diversi profili UV (a) e posizione del centroide del fascio in funzione dello spostamento (e relativo errore) (b)*

Naturalmente, essendo l'elettronica a 8 canali, sono state connesse soltanto 8 strisce (più precisamente 1 striscia ogni 4, per ottenere una struttura periodica): il rivelatore risultante simula dunque una struttura a 8 strisce di larghezza 80 μm e spaziatura 560 μm. La risoluzione spaziale di un rivelatore di questo tipo è, nel peggiore dei casi (*spot* focalizzato esattamente a metà della spaziatura fra due strisce), pari alla metà della spaziatura (circa 280 μm): per approfondire questo punto, lo spot UV è stato gradualmente traslato (a passi di 50 μm) nello spazio compreso fra due strisce adiacenti. Elaborando dopo ogni traslazione i segnali provenienti dalle 8 strisce, sono stati ottenuti 8 diversi profili 1D del fascio incidente (fig. 5.6a):
Dalla fig. 5.6a è evidente lo spostamento ad ogni traslazione del centroide del fascio, assicurando dunque una risoluzione spaziale di almeno 50 μm. Il centroide del fascio è stato poi calcolato, in termini di numero di striscia (canale), come media pesata dei segnali N_i raccolti dalle 8 strisce:

$$Centroid\ [\#ch] = \frac{\sum_i N_i \cdot i}{\sum_i i} \qquad [V.1]$$

dove i pesi *i* sono le posizioni delle strisce (canali). Nella fig. 5.6b è riportata la posizione del centroide in funzione dello spostamento del fascio: è da notare la relazione lineare, con un'incertezza massima di 5 μm (valutata come la distanza massima dei valori sperimentali dalla curva di *best fit*).
Per quanto riguarda misure di posizione nel caso 2D, è stato utilizzato il rivelatore 2D "*leaf-shaped*" di fig. 5.3: in particolare, è stato selezionato uno solo dei quattro quadranti in cui possibile suddividere l'area attiva del rivelatore, ottenendo complessivamente 8 elettrodi (4 strisce adiacenti orizzontali e 4 strisce adiacenti verticali). In questo caso, lo *spot* UV è stato gradualmente traslato lungo l'asse X, raccogliendo ed elaborando contemporaneamente i 4 segnali "X" e i 4 segnali "Y" provenienti dalle 8 strisce orizzontali e verticali.

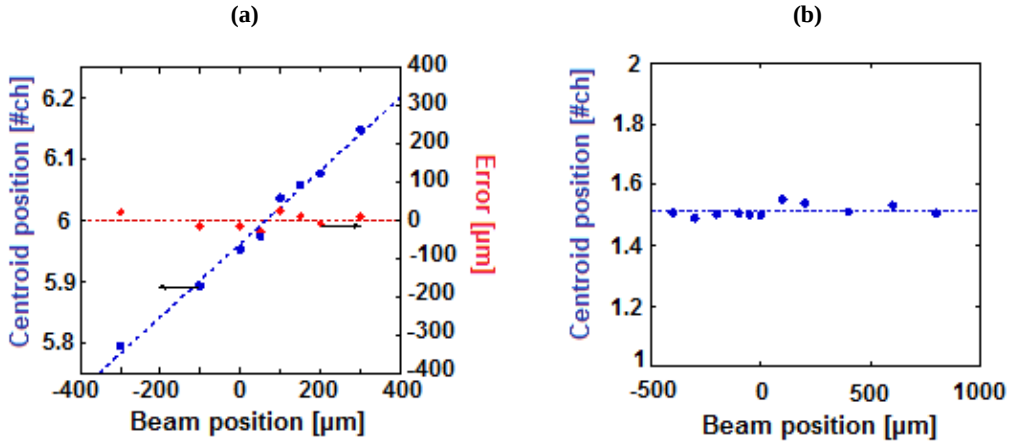


Fig. 5.7 *2D PSD con fascio UV in movimento lungo X: posizione del centroide del fascio in funzione dello spostamento lungo X (e relativo errore) (a) e lungo Y*

Si può notare ancora una relazione lineare fra posizione del centroide e spostamento del fascio lungo X (fig. 5.7a), con un'incertezza massima di 8 μm (maggiore che nel caso 1D, probabilmente a causa di una spaziatura maggiore fra gli elettrodi). Come previsto, inoltre, la posizione del centroide lungo Y è praticamente costante con lo spostamento del fascio lungo X (fig. 5.7b). Risultati del tutto analoghi sono stati ottenuti traslando il fascio UV lungo l'asse Y.

5.2.2 Emulazione di profili UV

Per valutare la discriminazione fra segnali provenienti da strisce (o pixel) adiacenti, sono state compiute delle emulazioni di profilo UV a 32 canali (per il caso dello *strip detector* 1D) e 64 canali (per il caso del *pixel detector* 2D), sempre utilizzando l'elettronica a 8 canali ECSIC8.
Nel caso monodimensionale, si è utilizzata ancora la configurazione descritta nel par. 5.2.1, ovvero uno *strip detector* a 32 strisce, di cui solo 8 connesse (1 ogni 4). Dopo di che, si è proceduto all'acquisizione successiva di 4 profili a 8 canali: ad ogni acquisizione, il rivelatore è stato traslato orizzontalmente di una quantità pari alla distanza interasse fra due strisce adiacenti (40 + 80 + 40 = 160 μm), in modo tale da poter ricostruire, alla fine delle 4 acquisizioni, un profilo a 32 canali. Questo profilo emula dunque il profilo UV che si sarebbe ottenuto, in un'unica acquisizione, con tutte e 32 le strisce connesse e con l'utilizzo di un'elettronica a 32 canali (ECSIC32). In fig. 5.8 è riportata in scala colorimetrica una delle 4 acquisizioni preliminari a 8 canali: è evidente l'andamento crescente dell'intensità del fascio nella zona centrale del rivelatore.

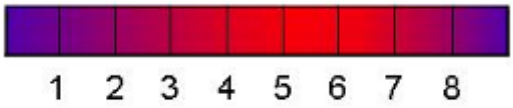


Fig. 5.8 *Profilo UV a 8 canali in scala colorimetrica*

In fig. 5.9 è invece riportata l'emulazione del profilo finale a 32 canali ricostruito alla fine delle 4 acquisizioni a 8 canali. Ad ogni singola acquisizione, sono stati prelevati 100 campioni da ognuna delle 8 strisce connesse, e di ogni pacchetto da 100 campioni è stato calcolato e riportato il valor medio: da notare, oltre ad una buona discriminazione dei segnali provenienti da due strisce adiacenti, è anche la bassa deviazione standard (barre d'errore) dei codici in uscita dall'ADC (asse delle ordinate), probabilmente dovuta alla schermatura in rame dell'elettronica di *front-end*.
Un'analoga procedura di emulazione, stavolta a 64 canali, è stata compiuta con un rivelatore 2D a 64 pixel (vedi fig. 5.2): in tal caso è stata connessa all'elettronica a 8 canali solo la prima fila da 8 pixel, poi sono state fatte 8 acquisizioni successive traslando verticalmente il rivelatore di una quantità pari alla distanza interasse fra due file di pixel adiacenti (250 + 200 + 250 = 700 μm), in modo tale da poter ricostruire, alla fine delle 8 acquisizioni, un profilo UV a 64 canali.

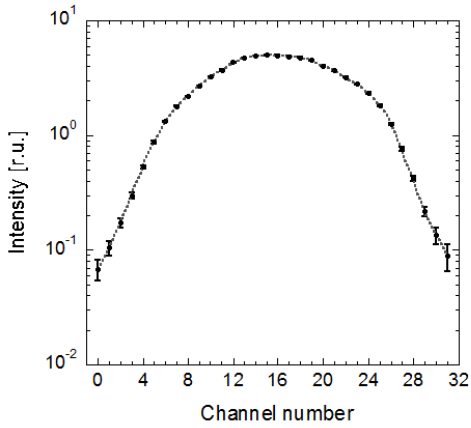


Fig. 5.9 *Emulazione di profilo UV monodimensionale a 32 canali*

Il profilo riportato in fig. 5.10 emula dunque il profilo UV che si sarebbe ottenuto, in un'unica acquisizione, con tutti e 64 i pixel connessi e con l'utilizzo di un'elettronica a 64 canali (ECSIC64).

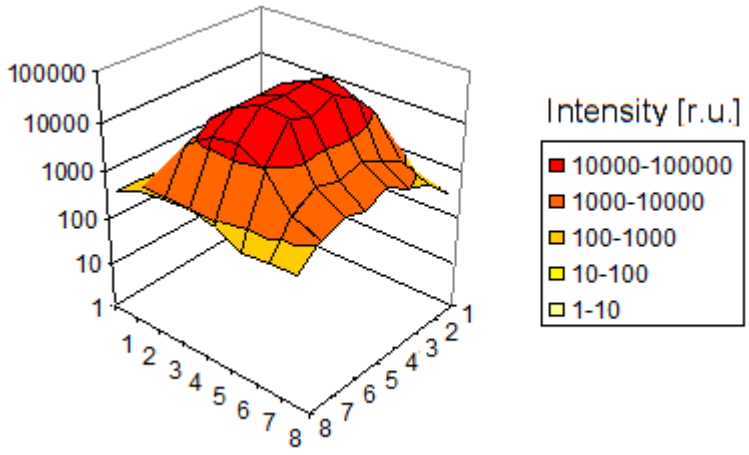


Fig. 5.10 *Emulazione di profilo UV bidimensionale a 64 canali*

5.3 *Imaging* di sorgenti UV di bassa intensità

In seguito alla verifica di funzionamento del sistema a 8 canali, presentata nel precedente paragrafo, si è proceduto ad una serie di acquisizioni di profili di sorgenti UV di bassa intensità attraverso l'utilizzo del rivelatore 2D a 36 pixel (vedi fig. 5.4) e dell'elettronica di condizionamento e lettura a 32 canali (sistema ECSIC32).
In tutte le operazioni di misura, prima dell'acquisizione dei profili, è stata seguita una procedura di calibrazione preliminare dell'intero sistema. Innanzitutto, il rivelatore è

stato montato su un sistema di movimentazione micrometrica X-Y avente una risoluzione di 10 μm; dopo di che, la radiazione UV è stata fatta arrivare sul rivelatore attraverso una fenditura circolare (*pinhole*) di 500 μm di diametro, in modo da illuminare solo una piccola porzione dell'area attiva.
La procedura di calibrazione parte con l'allineamento del centroide del fascio sul primo pixel (connesso) della prima riga, seguito da una prima acquisizione del segnale; successive traslazioni orizzontali e verticali pari alla distanza interasse fra i pixel (375 + 150 + 375 = 900 μm), seguite da altrettante acquisizioni, permettono di valutare tutti i fattori di normalizzazione da prendere in considerazione durante l'acquisizione ordinaria "parallela" dei 32 segnali in uscita dal rivelatore.
Un primo esempio di profilo reale UV bidimensionale, ottenuto illuminando uniformemente l'intera area attiva del dispositivo con una lampada Xe-Hg accoppiata ad un monocromatore, è riportato in fig. 5.11:

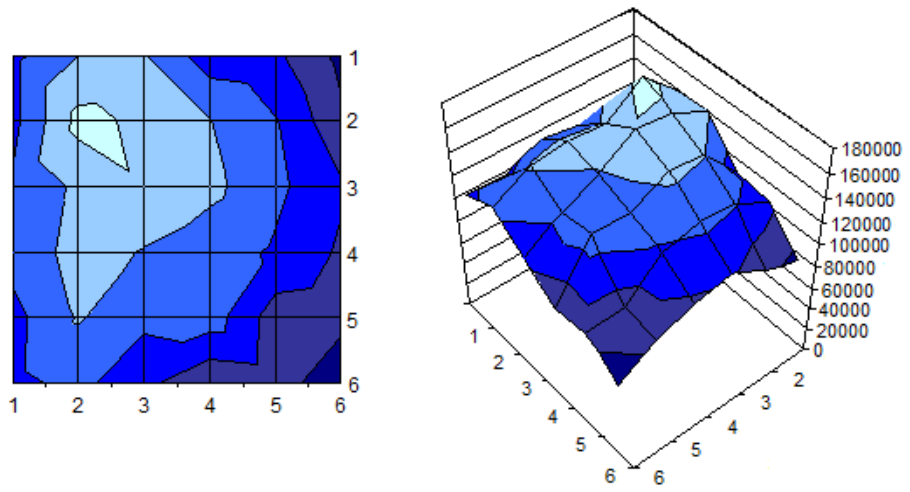


Fig. 5.11 *Esempio di profilo 2D a 32 canali di sorgente UV a Xe-Hg di bassa intensità*

Successivamente, per poter effettuare una valutazione della capacità del sistema di calcolare in tempo reale il centroide del fascio al variare della posizione d'incidenza sull'area attiva del rivelatore, è stato cambiato il tipo di sorgente UV: più precisamente, si è utilizzata una lampada UV a deuterio, della quale è stata selezionata (sempre attraverso un monocromatore) la componente a λ = 235 nm e avente potenza incidente pari a $P_{INC} \sim 4$ μW.
Dall'analisi del profilo del fascio ottenuto con questo nuovo *setup* (vedi fig. 5.12), si può notare come la radiazione incidente sul rivelatore abbia un profilo più netto e definito rispetto al caso precedente (fig. 5.11), e soprattutto meglio gestibile in termini di posizionamento sul rivelatore.
Dopo questo *beam profiling* preliminare, in cui la posizione iniziale del centroide del fascio sul rivelatore è del tutto casuale, è stata compiuta un'operazione di posizionamento del centroide al centro della zona attiva del rivelatore: mantenendo costante la distanza dalla sorgente, il rivelatore è stato traslato nelle due dimensioni

trasversali (X e Y) finché il risultato sulla misura del centroide non desse X = 3.5 e Y = 3.5.

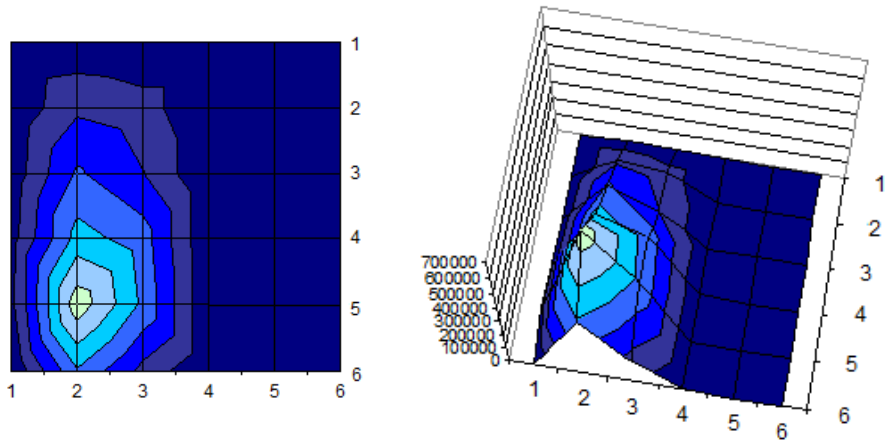
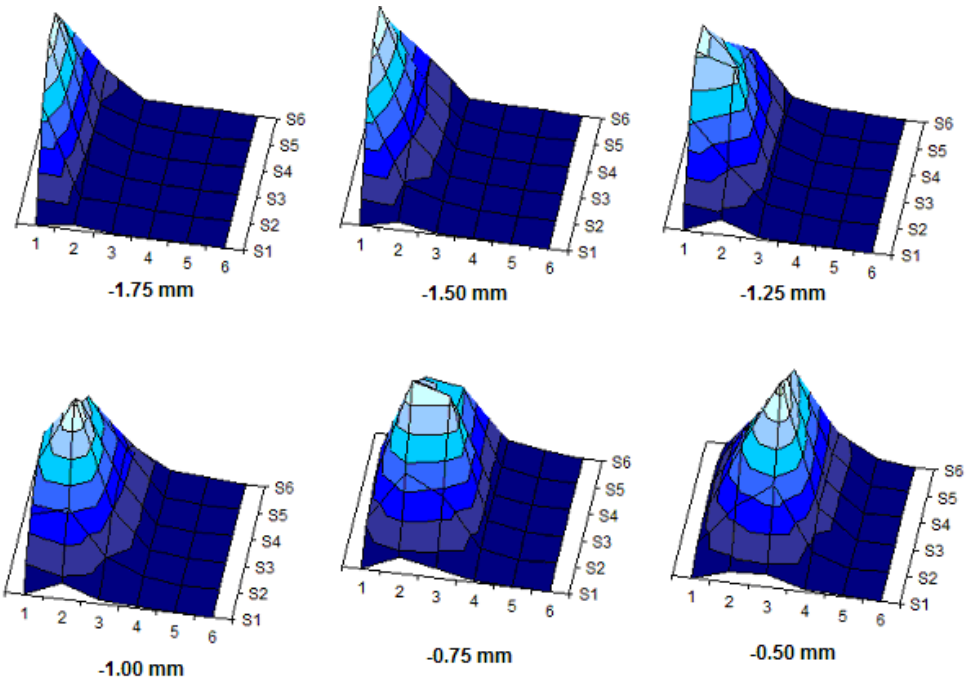


Fig. 5.12 *Esempio di profilo 2D a 32 canali di sorgente UV a deuterio di bassa intensità*

Dopo di che, partendo da questa posizione centrale e lasciando inalterata la coordinata verticale (Y), si è proceduto ad una serie di traslazioni lungo X del rivelatore (passi da 250 μm), acquisendo ad ogni passo il profilo a 32 canali del fascio, e calcolandone in tempo reale il centroide. Qui di seguito sono riportati i diversi profili acquisiti, ciascuno caratterizzato da uno spostamento relativo rispetto alla posizione centrale.



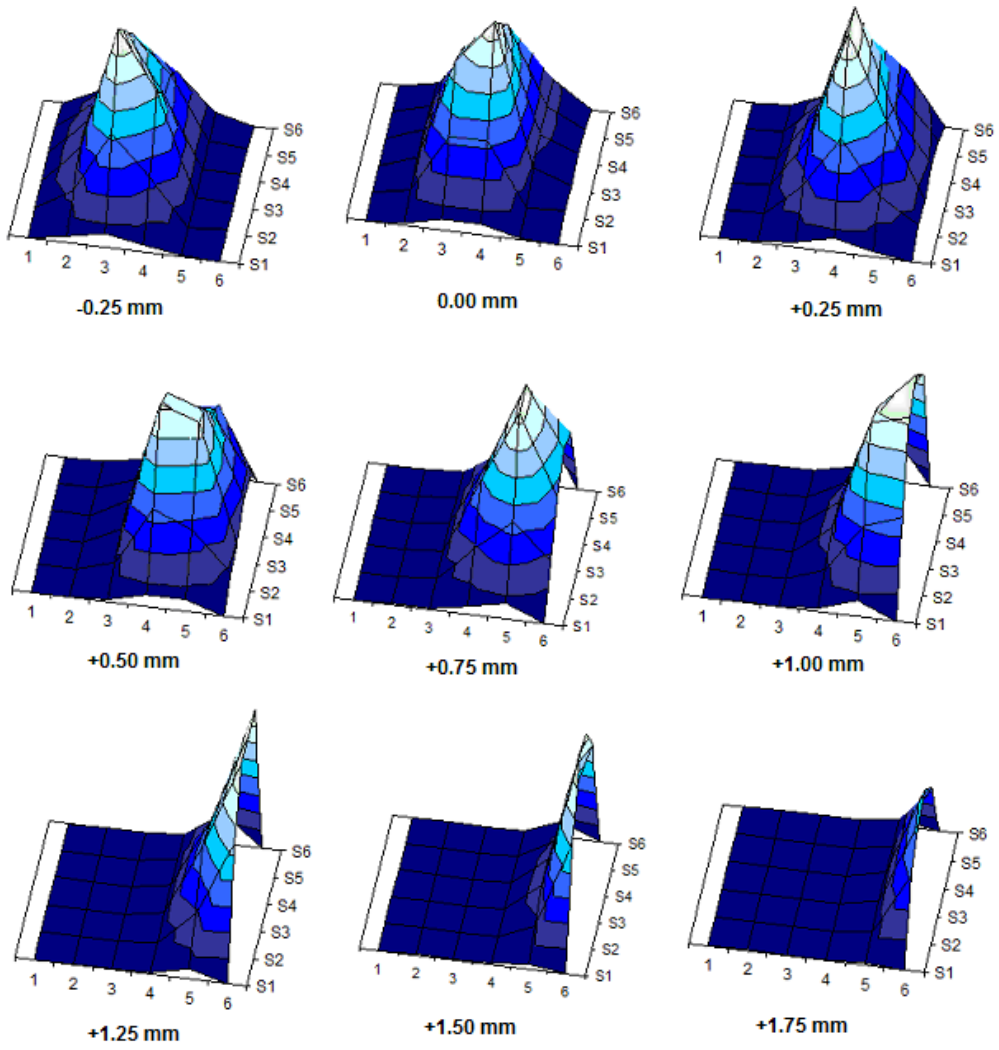
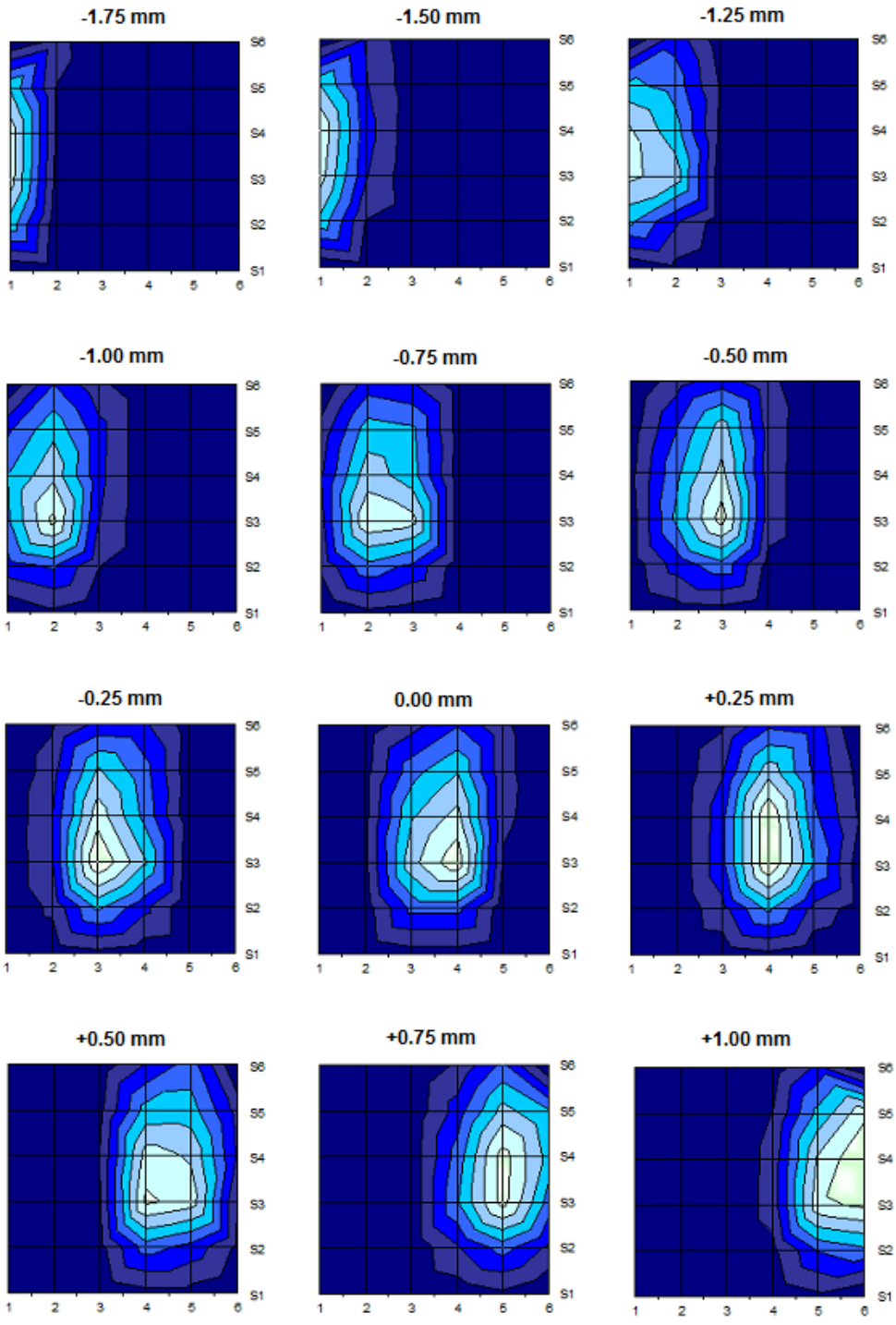


Fig. 5.13 *Profili UV acquisiti per successivi spostamenti orizzontali del rivelatore*

Per meglio apprezzare lo spostamento del centroide del fascio nelle diverse acquisizioni successive, vengono riportate qui di seguito anche una serie di distribuzioni spaziali dell'intensità del fascio sul piano d'incidenza (fig. 5.14): è evidente lo spostamento, da sinistra a destra lungo l'asse orizzontale dell'area attiva del rivelatore, del centroide del fascio.


-1.75 mm
-1.50 mm
-1.25 mm
-1.00 mm
-0.75 mm
-0.50 mm
-0.25 mm
0.00 mm
+0.25 mm
+0.50 mm
+0.75 mm
+1.00 mm
S6
S5
S4
S3
S2
S1
1
2
3
4
5
6

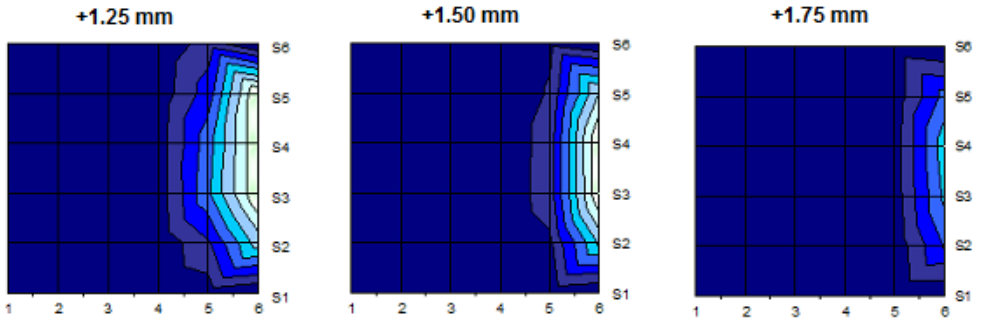


Fig. 5.14 *Variazione della distribuzione d'intensità UV sul piano del rivelatore*

Infine, per completezza, nella tabella 5.1 vengono riportate, per ogni acquisizione di profilo compiuta nell'esperimento descritto in precedenza, le coordinate del centroide del fascio in termini di pixel (numerati da 1 a 6 sia sull'asse X che sull'asse Y).

Spostamento	**Coordinata X centroide**	**Coordinata Y centroide**
−1.75 mm	1.36	3.57
−1.50 mm	1.49	3.59
−1.25 mm	1.69	3.58
−1.00 mm	2.02	3.56
−0.75 mm	2.40	3.52
−0.50 mm	2.82	3.58
−0.25 mm	3.25	3.54
0.00 mm	3.56	3.52
+0.25 mm	4.11	3.56
+0.50 mm	4.49	3.58
+0.75 mm	4.94	3.57
+1.00 mm	5.33	3.54
+1.25 mm	5.57	3.50
+1.50 mm	5.71	3.49
+1.75 mm	5.80	3.56

Tab. 5.1 *Coordinate del centroide del fascio UV al variare della posizione del rivelatore*

Lo spostamento è riferito rispetto al centro dell'area attiva del rivelatore: il segno "–" indica uno spostamento verso sinistra, il segno "+" verso destra. Si può notare come, nei diversi spostamenti, la coordinata Y del centroide rimanga pressoché costante, mentre la coordinata X varii linearmente con lo spostamento, come peraltro ben evidenziato in fig. 5.15. Ovviamente, si perde in linearità agli estremi della regione attiva del rivelatore: come si può visivamente notare dai profili di fig. 5.13 e 5.14, per spostamenti dal centro superiori (in modulo) a 1.25 mm, il centroide del fascio scompare dalla zona attiva del rivelatore, ed è dunque ovvio che l'informazione sul centroide sia destinata ad una naturale saturazione.

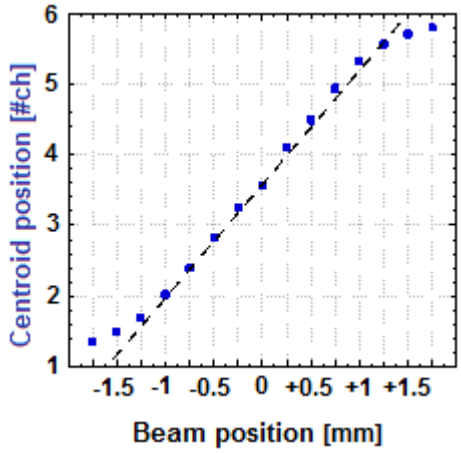


Fig. 5.15 *Andamento della coordinata X del centroide in funzione dello spostamento relativo*

5.4 *Imaging* di laser UV a eccimeri

Dopo la verifica del corretto funzionamento del sistema nell'analisi del profilo di sorgenti UV in continua, sono state condotte misure in condizioni di illuminazione da sorgenti UV impulsate, nella fattispecie da un laser UV a eccimeri ArF* con lunghezza d'onda λ = 193 nm.

Il *setup* di misura è analogo a quello utilizzato per le misure di fotoconducibilità UV impulsata per rivelatori a singolo pixel (vedi par. 3.6): l'unica differenza è l'eliminazione di ogni forma di attenuazione del fascio laser, allo scopo di verificare la robustezza del rivelatore (in particolar modo dei contatti metallici) in condizioni di irraggiamento ad alta energia. Il rivelatore utilizzato è ancora una volta quello di fig. 5.4 (rivelatore 2D a 36 pixel, di cui solo 32 connessi).

Una verifica preliminare è stata effettuata focalizzando lo *spot* del laser nella zona centrale del rivelatore, facendo in modo che andasse ad illuminare i 4 pixel centrali (fig. 5.16a): dopo di che si è proceduto all'analisi dell'evoluzione temporale del segnale di fotocorrente proveniente da ognuno di questi pixel. I risultati sperimentali ottenuti sono mostrati in fig. 5.16b.

In ascissa è riportata la successione dei campioni acquisiti (interpretabile dunque come una variabile temporale), mentre in ordinata sono riportati i codici in uscita dall'elettronica di *front-end* dopo la conversione ADC. Si può notare innanzitutto la variazione a gradino dei 4 segnali subito dopo l'accensione del laser, o meglio l'abilitazione del laser in uscita: dopo di che, è evidente il diverso andamento dei 4 segnali di fotocorrente, dovuto al fatto che lo *spot* del laser UV non illumina in maniera uniforme tutti e 4 i pixel. I segnali, dopo il transitorio mostrato in figura, vanno incontro ad una situazione stazionaria in cui si assestano ad un valor medio costante.

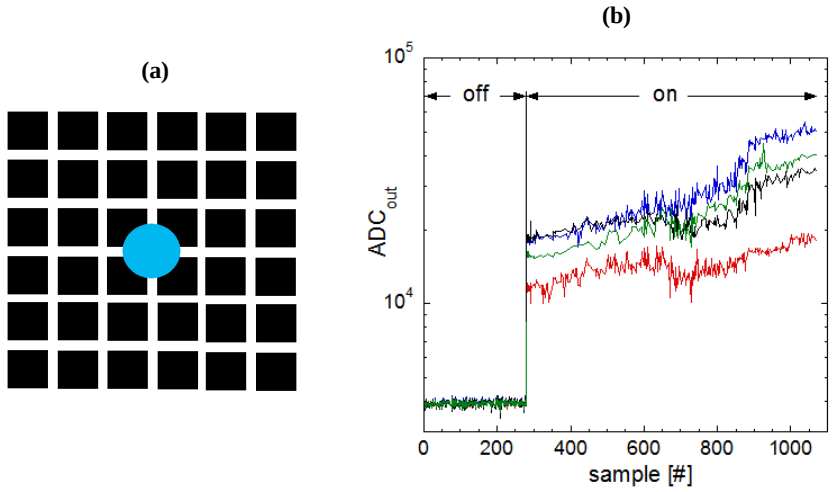


Fig. 5.16 *Posizione spot UV sul rivelatore (a) e segnali acquisiti dai 4 pixel centrali (b)*

Il comportamento rumoroso delle risposte dopo l'abilitazione in uscita del laser è imputabile a due fattori:

- l'intensità della radiazione incidente è leggermente differente ad ogni impulso, specialmente ad alti *repetition rates* (100 Hz), situazioni in cui l'efficienza di emissione laser diminuisce;
- non si può trascurare il contributo al rumore introdotto da un'interferenza elettromagnetica dovuta al generatore ad alta frequenza (*thyratron*) interno alla cavità laser.

Dopo questa verifica preliminare, sono state effettuate una serie di acquisizioni a 32 canali del profilo del fascio laser utilizzato. Per cercare di attenuare il più possibile il contributo al rumore dovuto al *thyratron* della cavità laser, il sistema (rivelatore ed elettronica di lettura) è stato posto ad una distanza di 3 m dalla fenditura di uscita del laser. Una prima acquisizione è stata compiuta con il laser in funzione, ma non abilitato in uscita (*shutter* chiuso), ed è riportata in fig. 5.17.

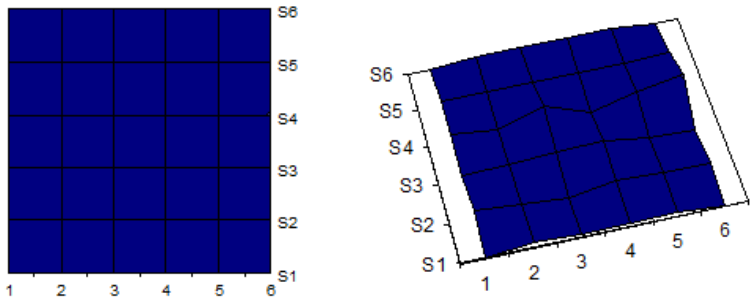


Fig. 5.17 *Acquisizione di profilo a shutter chiuso*

Dopo di che, lo *shutter* è stato aperto, ed è stato acquisito il profilo riportato in fig. 5.18. Il profilo è relativo ad un singolo impulso laser: le proprietà del diamante in termini di velocità di fotorisposta e l'alto *throughput* dell'elettronica di lettura (fino a 2 kHz) hanno infatti permesso di raccogliere ed elaborare i singoli impulsi del laser a eccimeri, anche in caso di massimo *repetition rate* (100 Hz).

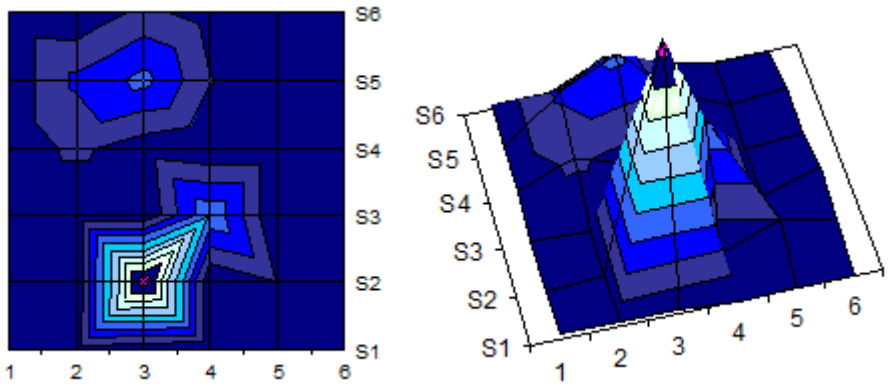


Fig. 5.18 *Profilo 2D di un impulso di laser a eccimeri ArF**

Il profilo in fig. 5.19, acquisito a distanza di tempo rispetto alle misure precedenti, presenta un doppio picco: molto probabilmente, questo sdoppiamento dell'impulso laser è dovuto ad un disallineamento della cavità, come riportato in letteratura per laser a eccimeri ArF* [73].

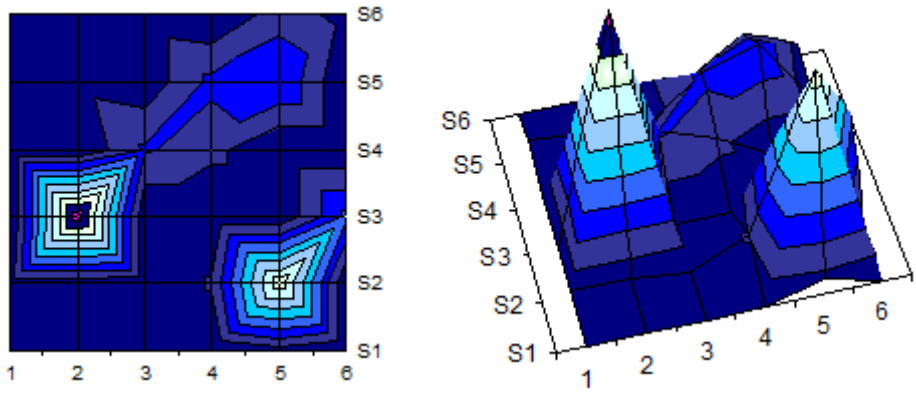


Fig. 5.19 *Profilo 2D di un impulso di laser a eccimeri ArF* in caso di disallineamento della cavità*

5.5 *Imaging* di sorgenti X

Per quel che riguarda l'analisi del profilo di una sorgente X, è stato utilizzato un tubo di Coolidge (vedi par. 1.5.1) con *target* in molibdeno, che produce (oltre a un fondo di radiazione di *Bremsstrahlung*) due intense righe a 17.5 keV (K_α) e 19.6 keV (K_β) [74]. La tensione di accelerazone del tubo a raggi X è stata fissata a 45 kV, per una corrente complessiva nel cannone elettronico di 1.1 mA.
Il rivelatore utilizzato per le misure è il sensore 2D a 36 pixel (vedi fig. 5.4). Per le procedure di calibrazione, nonché per poter effettuare delle acquisizioni di profilo al variare della posizione, il rivelatore (comprensivo dell'elettronica di *front-end* del sistema ECSIC32) è stato montato su un sistema di movimentazione submicrometrica XYZ. Nelle tre direzioni dello spazio, questo sistema di movimentazione passo-passo permette di compiere spostamenti per una lunghezza complessiva di 2.5 cm: tale distanza è coperta compiendo un numero totale di 620000 passi, pertanto la risoluzione del sistema di movimentazione (ovvero il singolo passo) è di soli 40 nm. Si è infine utilizzato un *chopper* come semplice otturatore.
In fig. 5.20 sono riportate due immagini del *setup* di misura: il rivelatore è stato inizialmente posizionato ad una distanza di 5 cm dalla sezione d'uscita del collimatore del tubo a raggi X: a questa distanza lo *spot-size* del fascio, misurato in precedenza tramite uno schermo a fluorescenza, è di circa 3 x 3 mm^2. Il rivelatore è stato successivamente polarizzato con una tensione di *bias* di 40 V.

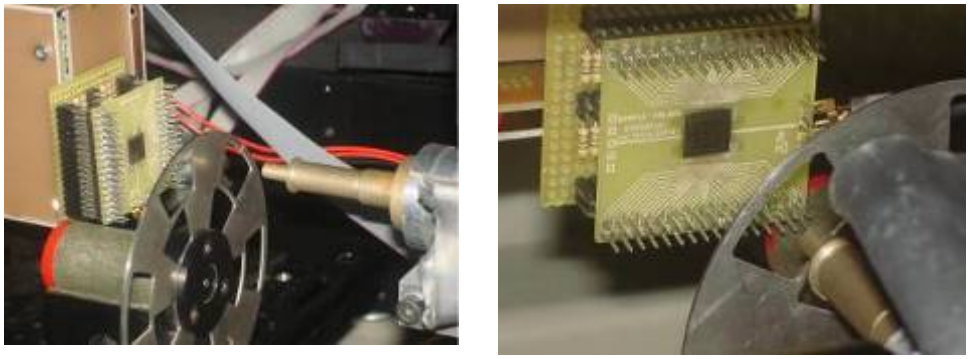


Fig. 5.20 *Setup di misura per l'imaging della sorgente X utilizzata*

Una prima serie di acquisizioni è stata compiuta per valutare il profilo del fascio X in diverse posizioni lungo la direzione orizzontale. In fig. 5.21 sono riportati tre diversi profili (3D e sul piano d'incidenza), ottenuti traslando ad ogni acquisizione il rivelatore di 22500 passi (pari ad uno spostamento di circa 900 μm, ovvero la distanza interasse fra due pixel adiacenti): dalla valutazione dei profili, è evidente come i risultati siano coerenti con gli spostamenti compiuti.

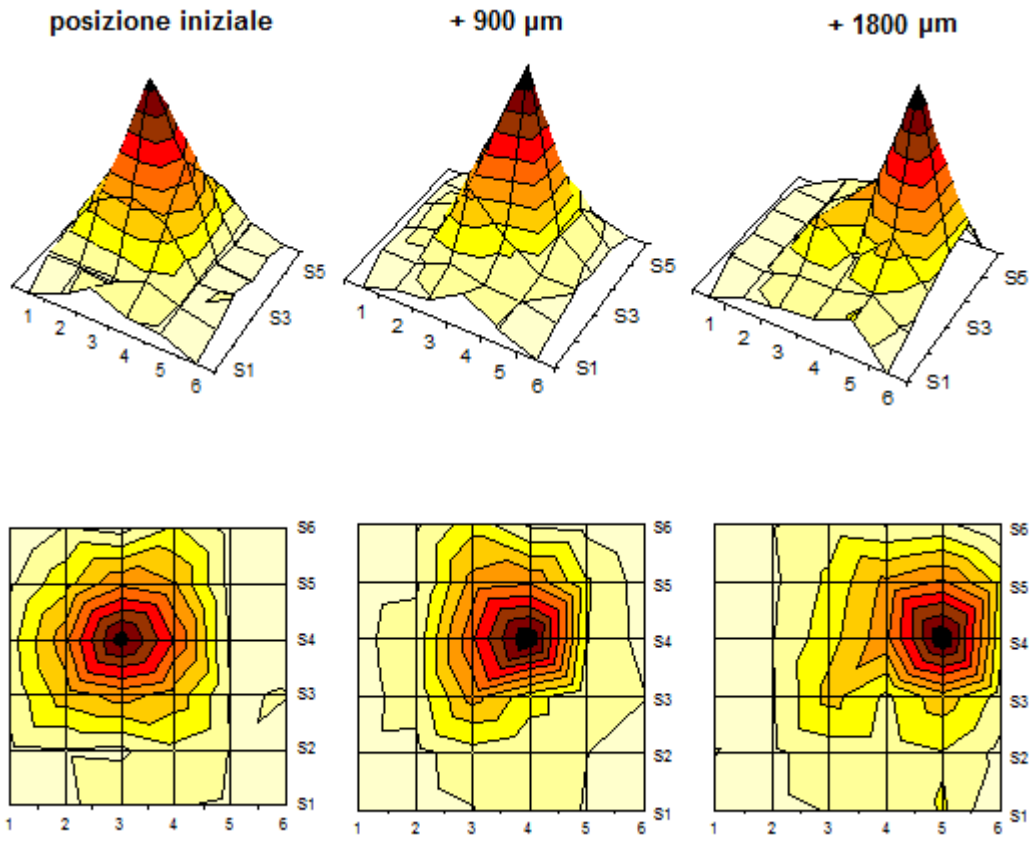


Fig. 5.21 *Profili del fascio X acquisiti per successivi spostamenti orizzontali del rivelatore*

Infine, lasciando il rivelatore in una posizione fissa riguardo alle coordinate trasversali (X e Y), è stata compiuta una serie di acquisizioni per diverse posizioni lungo l'asse Z, allontanando progressivamente il rivelatore dalla sezione d'uscita del collimatore del tubo a raggi X. In fig. 5.22 sono riportati tre profili significativi (3D e sul piano d'incidenza) delle misure effettuate, ottenuti traslando il rivelatore ad ogni acquisizione di 150000 passi (pari ad uno spostamento di circa 6 mm dal piano di partenza). Si può notare come l'allontanamento del rivelatore dalla sorgente X determini un leggero decremento del valore di picco dell'intensità sul piano d'incidenza, nonché un allargamento dello *spot-size* del fascio: entrambi questi effetti sono ovviamente dovuti alla divergenza subita dal fascio all'aumentare della distanza dalla sorgente.

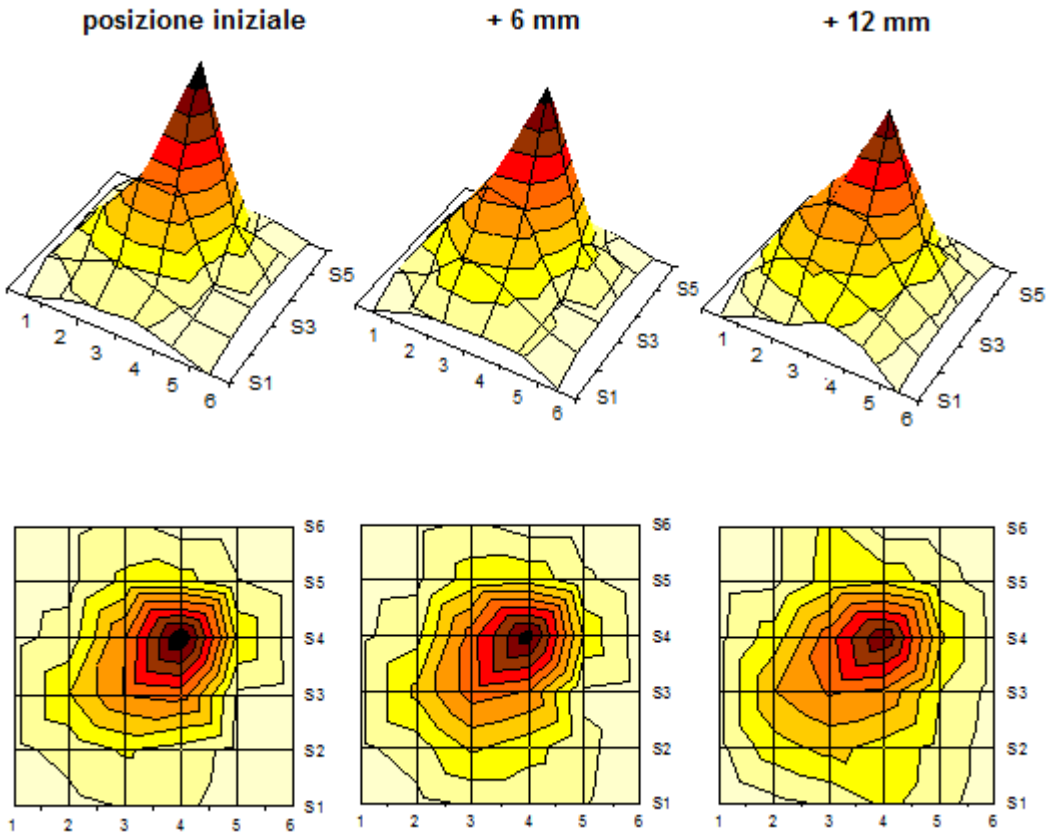


Fig. 5.22 *Profili del fascio X acquisiti per successivi spostamenti del rivelatore lungo l'asse Z*

Conclusioni

L'attività svolta per la presente tesi di dottorato ha avuto carattere prevalentemente sperimentale, e si è concentrata sui seguenti punti:

- *Realizzazione e caratterizzazione di rivelatori in diamante a singolo pixel per radiazioni UV e X*. Le strutture, realizzate con elettrodi metallici in Ag su diamante policristallino cresciuto con tecnica MWCVD, presentano basse correnti di buio (dell'ordine dei pA), e assicurano un comportamento ohmico fino a circa $3 \cdot 10^4$ V/cm di campo elettrico applicato (corrispondente ad una tensione di *bias* di circa 800 V) in configurazione *sandwich*. Le misure di fotoconducibilità spettrale hanno evidenziato l'ottima qualità del diamante impiegato, soprattutto in termini di discriminazione UV-VIS, pari a 4 ordini di grandezza. Le misure di fotoconducibilità impulsata UV con laser a eccimeri hanno messo in evidenza l'ottima velocità di risposta dei rivelatori (dell'ordine dei ns, paragonabile a quella di un tubo a vuoto), nonché un comportamento lineare della fotocorrente fino ad energie incidenti dell'ordine di $5 \cdot 10^{-3}$ mJ. Le misure di fotocorrente sotto raggi X hanno evidenziato una buona proporzionalità con l'intensità del fascio incidente, con un comportamento leggermente sub-lineare dovuto alla prevalenza di meccanismi di ricombinazione bimolecolare. Infine, è stata mostrata la possibilità di utilizzare i rivelatori per applicazioni di spettroscopia a raggi X, ottenendo risoluzioni in energia inferiori a 1 keV.

- *Progettazione e realizzazione di un sistema elettronico multicanale per l'acquisizione ed il condizionamento di deboli segnali in corrente*. Il sistema, denominato ECSIC, è caratterizzato da una testa di misura (elettronica di *front-end*) per rivelatori fino a 64 pixel, e da un'elettronica di elaborazione ad alta velocità, il tutto gestito da un microcontrollore. Il sistema ECSIC presenta una sensibilità (LSB) inferiore a 15 pC (a 150 pC di fondo-scala), una risoluzione massima nella conversione A/D pari a 20 *bit*, un alto *throughput* di dati (valore massimo 2 kHz), e una frequenza di trasmissione dati al PC di 460800 *bit/s*, consentendo di garantire un aggiornamento in tempo reale sull'analisi delle sorgenti monitorate. L'utente esterno può selezionare da remoto parametri di misura come il tempo d'integrazione (50 μs ÷ 1 s), il *range* di carica (50 ÷ 350 pC) e il numero di acquisizioni (1 ÷ 1024), ed ha inoltre la possibilità di richiedere al sistema di calcolare e restituire in tempo reale parametri caratteristici del fascio sotto analisi (centroide, valore massimo, valore minimo, valore atteso, scarto quadratico medio, spostamento).

- *Realizzazione e caratterizzazione di rivelatori in diamante multipixel per imaging di sorgenti UV e X*. Sono stati realizzati e testati con successo rivelatori 1D (*strip detectors*) e 2D (*pixel detectors*) come sensori di posizione UV, ottenendo una risoluzione spaziale inferiore a 10 μm. Sono stati acquisiti con un

rivelatore 2D a 32 pixel diversi profili di radiazioni UV, sia di debole intensità (lampade Xe-Hg o a deuterio), sia di potenza (laser a eccimeri), evidenziando in entrambi i casi la possibilità di effettuare un *imaging* in tempo reale della sorgente sotto esame: in particolare, le proprietà del diamante in termini di velocità di fotorisposta e l'alto *throughput* dell'elettronica di lettura hanno permesso di raccogliere ed elaborare i singoli impulsi del laser a eccimeri, anche in caso di massimo *repetition rate* (100 Hz). Infine, è stata verificata l'acquisizione di profili di fasci prodotti da un tubo a raggi X di elevata intensità: anche in questo caso il rivelatore 2D a 32 pixel ha dimostrato la possibilità di ottenere un'immagine 3D della sorgente monitorata, permettendo tra l'altro di effettuare un'analisi in tempo reale della divergenza del fascio all'aumentare della distanza del rivelatore.

Bibliografia

[1] M.S. Shur, A. Zukauskas, "*Basic Device Issues in UV Solid-state Emitters and Detectors*", in M.S. Shur, A. Zukauskas, "*UV Solid-State Light Emitters and Detectors*", Kluwer Academic Publishers, pp. 1-15 (2004)

[2] B. La Fontaine, "*Lasers and Moore's Law*", SPIE Professional, Oct.2010, p.20 (2010)

[3] I.G. Pallikaris, D.G. Siganos, "*Excimer Laser in situ Keratomileusis and Photorefractive Keratectomy for Correction of High Myopia"*, Journal of Refractive Corneal Surgery, 10, pp. 498-510 (1994)

[4] A. Richards, "*Digital Reflected-Ultraviolet Imaging*", Advanced Imaging, Apr. 2006, pp. 18-20 (2006)

[5] Y. Morimoto, T. Sumitomo, M. Yoshioko, T. Takemura, "*Recent Progress on UV Lamps for Industries*", IAS (IEEE Industry Application Society) 2004 Annual Meeting, pp. 24-31 (2006)

[6] http://sales.hamamatsu.com/en/products/solid-state-division/si-photodiode-series/si-photodiode/part-s1227-1010bq.php

[7] http://sales.hamamatsu.com/info/ims/#ccd

[8] M. Whitfield, S. Lansley, O. Gaudin, R. McKeag, N. Rizvi, R. Jackman, "*Diamond Photoconductors: Operational Lifetime and Radiation Hardness under Deep-UV Excimer Laser Irradiation*", Diamond and Related Materials, 10, pp. 715-721 (2001)

[9] http://www.ophiropt.com/user_files/laser/beam_profilers/UV-Lasers.pdf

[10] W.R. Leo, "*Techniques for Nuclear and Particle Physics Experiments: a How-to Approach*", 2nd Edition, Springer, p.175 (1994)

[11] P. Silvestroni, "*Fondamenti di chimica*", Masson (1992)

[12] C.D. Clark, P. J. Dean, P.V. Harris, "*Intrinsic Edge Absorption in Diamond*", Proceedings of the Royal Society A, 277, pp. 312-329 (1964)

[13] F.P. Bundy, "*Pressure-Temperature Phase Diagram of Elemental Carbon*", Physica A: Statistical Mechanics and its Applications, 156, pp. 169-178 (1989)

[14] L.S. Plano, "*Growth of CVD Diamond for Electronic Applications*", in "*Diamond: Electronic Properties and Applications*", L.S. Pan e D.R. Kania ed., Kluwer Academic Publishers (1995)

[15] B.S. Park, Y.J. Baik, "*Nucleation Enhancement Behavior of Diamond on Si Substrate According to Surface Treatment Materials*", Diamond and Related Materials, 6, pp. 1716-1721 (1997)

[16] S. Zhao, "*Characterization of the Electrical Properties of Polycrystalline Diamond Films*", PhD Thesis of the Graduate School of The Ohio State University (1994)

[17] K. Kobashi, K. Nishimura, Y. Kawate, T. Horiuchi, "*Synthesis of Diamond by use of Microwave Plasma Chemical Vapour Deposition: Morphology and Growth of Diamond Films,* Physical Review B, 38, pp. 4067-4084 (1988)
[18] J.E. Field, "*Properties of Diamond*", Academic Press, London (1979)
[19] K. Ueda, M. Kasu, Y. Yamauchi, T. Makinoto, M. Schwitters, D.J. Twitchen, G.A. Scarsbrook, S.E. Coe, "*Diamond FET using High-quality Polycrystalline Diamond with f_T of 45 GHz and f_{max} of 120 GHz*", IEEE Electron Device Letters, 27, pp. 570-572 (2006)
[20] M. Kasu, K. Ueda, H. Ye, Y. Yamauchi, N. Maeda, S. Sasaki, T. Makimoto, "*2W/mm Output Power Density at 1 GHz for Diamond FETs*", Electronics Letters, 41, pp. 1249-1250 (2005)
[21] S.A. Kajihara, A. Antonelli, J. Bernholc, R. Car, "*Nitrogen and Potential n-type Dopants in Diamond*", Physical Review Letters, 66, pp. 2010-2013 (1991)
[22] S. Koizumi, M. Kamo, Y. Sato, H. Ozaki, T. Inuzuka, "*Growth and Characterization of Phosphorous Doped {111} Homoepitaxial Diamond Thin Films*", Applied Physics Letters, 71, pp. 1065-1067 (1997)
[23] M.C. Rossi, S. Salvatori, F. Galluzzi, "*Gap Density of states in CVD Diamond Films from Photoconductivity and Photoluminescence Data*", Diamond and Related Materials, 6, pp. 712-716 (1997)
[24] A. Rose, "*Photoconductivity and Related Processes*", Interscience, New York (1963)
[25] F. Peter, "*Über Brechungsindizes und Absorptionskonstanten des Diamanten zwischen 644 und 226 mμ"*, Zeitschrift für Physik, 15, pp. 358-368 (1923)
[26] W. Shockley, "*Problems Related to p-n Junctions in Silicon*", Solid State Electronics, 2, pp. 35-67 (1961)
[27] C. Canali, E. Gatti, F. Kozlov, F. Manfredi, C. Manfredotti, F. Nava, A. Quirini, "*Electrical Properties and Performances of Natural Diamond Nuclear Radiation Detectors*" Nuclear Instruments and Methods in Physics Research Section A, 160, pp.73-77 (1979)
[28] D.R. Kania, L.S. Pan, "*Diamond: Electronic Properties and Applications"*, Kluwer Academic Publishers, Boston - Dordrecht - London (1993)
[29] S.M. Sze, "*Semiconductor Devices: Physics and Technology*", Wiley, New York (1985)
[30] "*Shipley Microposit® S1800® Series Photoresists*", Datasheet, Shipley Company, Marlborough, MA, USA
[31] "*Keithley Model 6517A Electrometer User's Manual*", Keithley Instruments Inc., Cleveland OH, USA (2004)
[32] S. Ashok, K. Srikanth, A. Badzian, T. Badzian, R. Messier, "*Space-Charge-Limited Current in Thin-Film Diamond*", Applied Physics Letters, 50, pp. 763-765 (1987)
[33] B. Huang, D.K. Reinhard, J. Asmussen, "*Electrical Properties of Undoped Large-Grain and Small-Grain Diamond Films*", Diamond and Related Materials, 2, pp. 812-815 (1993)
[34] C. Jany, F. Foulon, P. Bergonzo, A. Brambilla, A. Gicquel, T. Pochet, "*Influence of CVD Diamond Film Textures on the Electrical Response of Radiation*

Detectors", Nuclear Instruments and Methods in Physics Research Section A, 380, pp. 107-111 (1996)
[35] J.L. Pankove, "*Optical Processes in Semiconductors*", Dover Publications Inc., New York (1971)
[36] http://sales.hamamatsu.com/assets/pdf/parts_H/H8497-11.pdf
[37] J.D. Brownridge, S. Raboy, "*Investigation of Pyroelectric Generation of X-rays*", Journal of Applied Physics, 86, pp.640-647 (1999)
[38] http://www.amptek.com/pdf/coolx.pdf
[39] H. Spieler, "*Semiconductor Detector Systems*", Oxford Science Publications (2005)
[40] A. Thompson et al., "*X-Ray Data Booklet*", Lawrence Berkeley National Laboratory, University of California (2009)
[41] R.A. Pease, "*Troubleshooting Analog Circuits*", EDN Series for Design Engineers, Newnes (1991)
[42] "*Low Level Measurements Handbook*", 6th edition, Keithley Instruments Inc., Cleveland OH, USA (2004)
[43] "*OPA128 Difet® Electrometer Grade Operational Amplifier*", Datasheet, Burr Brown Corp., Tucson AZ, USA (1995)
[44] "*AD549 Ultralow Input Bias Current Operational Amplifier*", Datasheet, Analog Devices Inc. (2008)
[45] I. Sinclair, "*Passive Components for Circuit Design*", Newnes (2001)
[46] S. Salvatori, G. Conte, "*Precision Integrator Sparks Current-Ratio-to-Frequency Converter*", EDN'S Design Ideas, May 24th 2007 (2007)
[47] "*DDC101 20-bit Analog-to-Digital Converter*", Datasheet, Burr Brown Corp., Tucson AZ, USA (2000)
[48] "*DDC112 Dual Current Input, 20-bit Analog-to-Digital Converter*", Datasheet, Rev. B, Burr Brown Corp., Tucson AZ, USA (2004)
[49] "*DDC118 Octal Current Input, 20-bit Analog-to-Digital Converter*", Datasheet, Rev. B, Texas Instruments Inc., Dallas TX, USA (2009)
[50] "*DDC232 32-Channel, Current Input Analog-to-Digital Converter*", Datasheet, Rev. D, Texas Instruments Inc., Dallas TX, USA (2010)
[51] "*Model 6220 DC Current Source and Model 6221 AC and DC Current Source*", Datasheet, Keithley Instruments Inc., Cleveland OH, USA (2005)
[52] "*UM10161: LPC2101/2102/2103 User Manual*", Rev. 4, NXP Semiconductors, Eindhoven, the Netherlands (2009)
[53] "*LPC2101/2102/2103: Single-chip 16-bit/32-bit microcontrollers; 8 kB/16 kB/32 kB flash withISP/IAP, fast ports and 10-bit ADC*", Preliminary datasheet, Rev. 01, Koninklijke Philips Electronics N.V., Eindhoven, the Netherlands (2006)
[54] S. Salvatori, G. Di Nucci, "*Two-Wire Four-by-Four-Key Keyboard Interface Saves Power*", EDN's Design Ideas, August 17th 2006 (2006)
[55] "*W24257A*: *32K × 8 High Speed CMOS Static RAM*", Datasheet, Revision A14, Winbond Electronics Corp., Hsinchu, Taiwan (1997)
[56] "*ARM7TDMI Technical Reference Manual*", Revision: r4p1, ARM Inc., Cambridge, UK (2001)
[57] S. Furber, "*ARM System-on-Chip Architecture*", 2nd edition, Pearson Education Limited (2000)

[58] D. Seal, "*ARM Architecture Reference Manual*", 2nd edition, Addison-Wesley Professional (2000)
[59] "*PrimeCell® Vectored Interrupt Controller (PL190): Technical Reference Manual*", Revision: r1p2, ARM Inc., Cambridge, UK (2004)
[60] "*ispMACH™ 4A CPLD Family: High Performance E²CMOS® In-System Programmable Logic*", Datasheet, Rev. M, Lattice Semiconductors Corp., Hillsboro OR, USA (2006)
[61] J. Todsen, "*Retrieving Data from the DDC112*", Burr Brown Application Bulletin, doc.SBAA026 (2000)
[62] J. Todsen, "*The DDC112's Test Mode*", Burr Brown Application Bulletin, doc.SBAA025 (2000)
[63] J. Todsen, "*Understanding the DDC112's Continuous and Non-Continuous Modes*", Burr Brown Application Bulletin, doc.SBAA024 (2000)
[64] "*LM111, LM211, LM311 Differential Comparator with Strobes*", Datasheet, doc. SLCS700H, Texas Instruments Inc., Dallas TX, USA (2003)
[65] "*HD44780 (LCD-II): Dot Matrix Liquid Crystal Display Controller/Driver*", Datasheet, ADE-207-272(Z), Rev. 0.0, Hitachi Ltd., Tokyo, Japan (1999)
[66] D. Bala, "*The HD44780 LCD Controller Application Notes*" (2001)
[67] "*PC1602ARU-HWB-Q*", Datasheet, Rev. 0, Powertip Tech. Corp., Taichung, Taiwan (2006)
[68] "*EIA Standard RS-232-C Interface Between Data Terminal Equipment and Data Communication Equipment Employing Serial Data Interchange*", Electronic Industries Association (1969)
[69] "*TRS232E, Dual RS-232 Driver/Receiver with IEC61000-4-2 Protection*", Datasheet, doc.SLLS791C, Texas Instruments Inc., Dallas TX, USA (2008)
[70] http://www.olimex.com/dev/lpc-p2103.html
[71] "*OPA350, OPA2350, OPA4350, High-Speed, Single-Supply, Rail-to-Rail Operational Amplifiers, MicroAmplifiers™ Series*", Datasheet, doc.SBOS099C, Burr Brown Corp., Tucson AZ, USA (2005)
[72] "*SN54HC04, SN74HC04 Hex Inverters*", Datasheet, doc.SCLS078D, Texas Instruments Inc., Dallas TX, USA (2003)
[73] P. Parvin, B. Jaleh, H.R. Zangeneh, Z. Zamanipour, Gh.R. Davoud-Abadi, "*Excimer Laser Beam Profile Recording Based on Electrochemical Etched Polycarbonate*", Radiation Measurements 43, pp. 617-622 (2008)
[74] J.E. Martin, "*Physics for Radiation Protection: a Handbook*", 2nd edition, Wiley-VCH, Weinheim, Germany (2006)

Partecipazione a conferenze nazionali e internazionali

- EMRS (European Materials Research Society) 2010 Spring Meeting – Strasburgo, Francia 7-11 giugno 2010:
    - Presentazione poster "Excimer laser beam analyzer based on CVD diamond", M. Girolami, S. Salvatori, G. Conte
    - Presentazione orale "UV-triggered MESFET receivers", P. Calvani, M. Girolami, G. Conte, E. Giovine, A.P. Bolshakov, V.G. Ralchenko

- Diamond 2009 - 20th European Conference on Diamond, Diamond-like materials, Carbon Nanotubes and Nitrides: Presentazione poster "Polycrystalline diamond UV laser beam profilers", M. Girolami, P. Allegrini, F. Sinisi, S. Salvatori, G. Conte – Atene, Grecia, 6-10 settembre 2009

- NDNC 2009 – 3rd International New Diamond and Nano Carbons Conference: Presentazione poster "UV Photodetection in Surface Channel Diamond MESFET" , M.C. Rossi, P. Calvani, M. Girolami, S. Carta, G. Conte – Traverse City, Michigan, USA, 7-11 giugno 2009

- NANOTEC2009.IT – Convegno Internazionale sulle Nanotecnologie: Presentazione poster "UV enhanced field effect transistors on H-terminated diamond", P. Calvani, M. Girolami, M.C. Rossi, G. Conte – Roma, CNR (Consiglio Nazionale delle Ricerche), 31 marzo-3 aprile 2009

- XIV Conferenza Nazionale AISEM (Associazione Italiana Sensori e Microsistemi) – Pavia, 24-26 febbraio 2009:
    - Presentazione orale "UV Laser Beam Profilers Based on CVD Diamond", M. Girolami, P. Allegrini, P. Calvani, G. Conte, S. Salvatori
    - Presentazione poster "Photoconductive Position Sensitive CVD Diamond Detectors", P. Allegrini, M. Girolami, P. Calvani, G. Conte, S. Salvatori
    - Presentazione poster "UV Schottky Sensors Based on Wide Bandgap Semiconductors", P. Calvani, M. Girolami, P. Allegrini, G. Conte, M.C. Rossi
    - Presentazione poster "Phototransistors on H-Terminated Diamond", A. Del Monte, P. Allegrini, P. Calvani, M. Girolami, G. Conte, M.C. Rossi

- IEEE Sensors 2008 – The 7th IEEE Conference on Sensors: Presentazione poster "CVD Diamond Detectors for Real-Time Beam Profile Measurements", M. Girolami, P. Allegrini, G. Conte, S. Salvatori – Lecce, 26-29 ottobre 2008

- IEEE Dresden 2008 – NSS-MIC-RTSD, Nuclear Science Symposium, Medical Image Conference and 16th Room Temperature Semiconductor Detectors Workshop: Presentazione poster "X-Ray Spectroscopy Based on Polycrystalline Diamond", M. Girolami, P. Allegrini, S. Salvatori, G. Conte, E. Spiriti, V.G. Ralchenko – Dresda, Germania, 19-25 ottobre 2008

- SPIE Optics+Photonics 2008, Optical Engineering+Applications Symposium (section: X-ray, Gamma Ray and Particle Technologies) – 3rd Conference on Advances in X-ray/EUV Optics and Components: Presentazione poster "Diamond detectors for X-ray spectroscopy", M. Girolami, P. Allegrini, G. Conte, A. Fabbri, E. Spiriti, S. Salvatori – San Diego, California, USA, 10-14 agosto 2008

- NDIP 2008 – 5th International Conference on New Developments In Photodetection: Presentazione poster "Optoeletronic performance of GaN based UV photodetectors", P. Calvani, M. Girolami, S. Carta, M.C. Rossi, G. Conte – Aix-les-Bains, Francia, 15-20 giugno 2008

- XIII Conferenza Nazionale AISEM (Associazione Italiana Sensori e Microsistemi): Presentazione poster "Energy-resolving diamond detectors for X-ray spectroscopy", P. Allegrini, M. Girolami, G. Conte, S. Salvatori – Roma, 19-21 febbraio 2008

- Diamond 2007 - 18th European Conference on Diamond, Diamond-like materials, Carbon Nanotubes and Nitrides: Presentazione poster "Energy-resolved X-ray diamond detectors", M. Girolami, G. Conte, M.C. Rossi, S. Salvatori, V.G. Ralchenko – Berlino, Germania, 9-14 settembre 2007

- XII Conferenza Nazionale AISEM (Associazione Italiana Sensori e Microsistemi): Presentazione poster "Diamond detectors for X-ray sources monitoring", G. Conte, M. Girolami, S. Salvatori, R. Scafè, F. Pisacane, D. Della Sala – Napoli, 12-14 febbraio 2007

Pubblicazioni

2011

1. **M. Girolami**, P.Allegrini, S. Salvatori, G. Conte – **"Pixel diamond detectors for excimer laser beam diagnostics"** – Proceedings of SPIE (*submitted for publication*)

2010

2. **M. Girolami**, S. Salvatori, G. Conte – **"Excimer Laser Beam Analyzer Based on CVD Diamond"** – 2010 Wide Bandgap Cubic Semiconductors: From Growth to Devices: Proceedings of the E-MRS (European Materials Reserach Society) Symposium F, AIP Conference Proceedings, vol. 1292, Issue 1, American Institute of Physics, pp. 145-148, ISBN 978-0-7354-0847-0 (2010)
3. **M. Girolami**, P. Allegrini, S. Salvatori, G. Conte – **"UV laser beam profilers based on CVD diamond"** – Sensors and Microsystems, AISEM 2009 Proceedings, Series: Lecture Notes in Electrical Engineering Vol.54, Springer, pp. 101-104, ISBN 978-90-481-3605-6 (2010)
4. **M. Girolami**, P. Allegrini, S. Salvatori, G. Conte **– "Photoconductive position sensitive CVD diamond detectors"** – Sensors and Microsystems, AISEM 2009 Proceedings, Series: Lecture Notes in Electrical Engineering vol.54, Springer, pp. 105-108, ISBN 978-90-481-3605-6 (2010)
5. P. Allegrini, P. Calvani, **M. Girolami**, G. Conte, M. C. Rossi – **"UV Schottky Sensors Based on Wide Bandgap Semiconductors"** – Sensors and Microsystems, AISEM 2009 Proceedings, Series: Lecture Notes in Electrical Engineering vol.54, Springer, pp. 137-142, ISBN 978-90-481-3605-6 (2010)

2009

6. S.Salvatori, **M.Girolami** – **"Compact, four-quadrants lock-in amplifier generates two analog outputs"**, EDN's Design Ideas 15/12/2009, pp. 45-47 (2009)
7. P. Calvani, M. Girolami, S. Carta, M.C. Rossi, G. Conte "**Optoelectronic performance of GaN-based UV photodetectors**", Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipments, Vol. 610, Issue 1, pp.311-313 (2009)
8. P.Calvani, A.Corsaro, **M.Girolami**, F.Sinisi, D.M.Trucchi, M.C.Rossi, G.Conte, S.Carta, E.Giovine, S.Lavanga, E.Limiti, V.Ralchenko – "**DC and RF Performance of surface channel MESFETs on H-terminated polycrystalline diamond**" – Diamond and Related Materials, Vol. 18, pp. 786-788 (2009)

2008

9. **M.Girolami**, P.Allegrini, A.Fabbri, S.Salvatori, G.Conte – **"Energy-resolving diamond detectors for X-ray spectroscopy"** – Sensors and Microsystems, Proceedings of the 13th Italian Conference, World Scientific Publishing, pp. 307-312, ISBN 978-981-283-597-0 (2008)

10. **M.Girolami**, P.Allegrini, S.Salvatori, G.Conte, E.Spiriti, V.Ralchenko – **"X-Ray Spectroscopy Based on Polycrystalline Diamond"** – IEEE Dresden 2008 Conference Record (NSS-MIC-RTSD: Nuclear Science Symposium, Medical Image Conference and 16th Room Temperature Semiconductor Detectors Workshop), pp. 227-234 (2008)

11. **M.Girolami**, P.Allegrini, G.Conte and S.Salvatori – **"CVD Diamond Detectors for Real-Time Beam Profile Measurements"** – Proceedings of The 7th IEEE Conference on Sensors, pp. 270-273 (2008)

12. P.Allegrini, **M.Girolami**, P.Calvani, G.Conte, S.Salvatori, E.Spiriti, V.Ralchenko – **"Diamond detectors for X-ray spectroscopy"** – Proceedings of SPIE, Vol. 7077, 70771P (2008)

13. G.Conte, **M.Girolami**, S.Salvatori, R.Scafè, F.Pisacane, D.Della Sala - **"Diamond detectors for X-ray sources monitoring"** – Sensors and Microsystems, Proceedings of the 12th Italian Conference, World Scientific Publishing, pp. 515-519, ISBN 978-981-283-358-7 (2008)

2007

14. G.Conte, **M.Girolami**, S.Salvatori, V.Ralchenko – **"X-ray diamond detectors with energy resolution"** – Applied Physics Letters, Vol. 91, Issue 18, 183515 (2007)

15. **M.Girolami**, G.Conte, M.C.Rossi, S.Salvatori, V.Ralchenko – **"Energy-resolved X-ray diamond detectors"** – Diamond 2007 Conference Proceedings, Elsevier (2007)