UNIVERSITÀ DEGLI STUDI DI PADOVA

Facoltà di Ingegneria

Corso di Laurea Magistrale in Ingegneria delle Telecomunicazioni

Tesi di Laurea

ELETTRONICA INTEGRATA PER LA COMUNICAZIONE QUANTISTICA

Relatore:
Prof. PAOLO VILLORESI

Laureando:
ENRICO BALLARIN

Correlatore:
Ing. DANIELE VOGRIG

ANNO ACCADEMICO 2010-2011

Alla mia famiglia

Sommario

Questo progetto intende fornire una soluzione di natura elettronica al problema dell'individuazione di due eventi che si manifestano in istanti temporali molto ravvicinati tra loro. Più precisamente, parliamo di segnali elettrici con determinate caratteristiche di intensità e durata, provenienti, dopo la necessaria conversione, da un sistema di tipo quantistico.
Il lavoro svolto è la continuazione e l'ampliamento di quanto precedentemente realizzato dai colleghi Marco Pezzini e Andrea Merlin nell'ambito della loro tesi di laurea; l'ambiente in cui è stato sviluppato è il Laboratorio Luxor dell'Università di Padova, all'interno del quale è stato possibile accedere a tutta la strumentazione necessaria per i test sperimentali.

In questo elaborato si vuole dare una visione il più possibile ad ampio raggio dell'argomento di interesse. Introdurremo i concetti di meccanica quantistica, fondamentali per comprendere il contesto in si opera, e quindi passeremo a descrivere il fenomeno fisico, l'entanglement, di cui ci proponiamo di realizzare uno strumento di misura. Dalla dimensione ottica, nota la procedura di conversione dei fotoni in segnali elettrici, sposteremo la nostra attenzione alle metodologie che permettono la rivelazione di intervalli temporali molto brevi e, dopo aver motivato la scelta dell'hardware adatto alle nostre esigenze e averne elencato le funzionalità, partiremo con la descrizione del progetto vero e proprio. Seguirà un capitolo dedicato ai test sperimentali eseguiti per valutare qualità e punti deboli del dispositivo realizzato e, infine, verranno tratte le dovute conclusioni prospettando nuovi scenari che permettano di risolvere eventuali problemi rimasti irrisolti e migliorare le prestazioni complessive del sistema sviluppato.

Indice

Elenco delle figure

Introduzione

AGLI INIZI del ventesimo secolo lo studio della struttura dell'atomo e delle sue componenti ha indotto la scienza a riconsiderare gran parte dei punti fermi su cui si basavano le discipline fisiche classiche. Da un lato la scoperta del quanto di luce, elemento fondamentale per la formulazione della teoria della meccanica quantistica, ha riportato in vita la discussione sulla natura della luce, mentre dall'altro, la concezione ondulatoria della materia ha costretto gli scienziati a trattare radiazione e materia allo stesso modo attraverso un modello duale, ammettendo che entrambe presentano comportamenti ondulatori o particellari a seconda dell'esperimento condotto. Se, dunque, per la fisica classica di Newton e Maxwell onde e particelle rappresentavano entità distinte con proprietà differenti, è con la meccanica quantistica che si è dedotto come un elettrone rimbalzasse allo stesso modo di una particella e interferisse come un'onda.

Nel momento in cui, grazie a questa nuova disciplina, è stato possibile determinare l'evoluzione di sistemi microscopici costituiti, ad esempio, da fotoni o atomi in grado di fornire informazione tramite un particolare stato di polarizzazione, si è assistito alla comparsa e al successivo sviluppo della cosiddetta *quantum information theory*.

Uno dei fenomeni più importanti che riguardano lo studio di tale materia è l'*entanglement*, concetto che può essere assimilato all'idea di non separabilità dello stato di un sistema fisico composto, e che rende conto della correlazione presente tra i sistemi elementari da cui deriva. Volendo dare una definizione più rigorosa, due o più particelle vengono dette entangled se il loro stato di sovrapposizione non è fattorizzabile nello stato delle singole particelle, anche se queste risultano spazialmente separate.
Sorge allora il problema di come generare e rivelare particelle con le caratteristiche poc'anzi descritte. Il processo di generazione al momento più affidabile prende il nome di *parametric down conversion* e consiste nell'irradiazione di un cristallo non lineare da parte di un fascio laser, detto di pompa, che decade spontaneamente in coppie di fotoni, appunto, entangled. Per quanto riguarda la rivelazione di fotoni, l'affidabilità sempre maggiore degli SPAD (*Single Photon Avalanche Diode*) ha dato un forte impulso allo sviluppo di applicazioni che sfruttano tali coppie, consentendo la misura di intensità luminose molto deboli proprie della singola particella.

All'interno di questo contesto, il mio lavoro di tesi si propone di realizzare un sistema di rivelazione di fotoni entangled provenienti da quattro canali distinti. Il

criterio per determinare il verificarsi di questo fenomeno è, in questo caso, di tipo temporale: individuata la presenza di un fotone in un canale, si determina l'istante in cui si è manifestato e lo si confronta, all'interno di un'opportuna finestra temporale di osservazione, con gli eventuali istanti di arrivo accertati negli altri canali; qualora l'intervallo che intercorre tra due di questi sia sufficientemente piccolo, i due fotoni associati alle misure vengono considerati entangled. Il suddetto procedimento definisce il sistema di *time tagging* oggetto di questa pubblicazione.
La piattaforma hardware sulla quale poggia il progetto è una FPGA ML605 Virtex-6 prodotta dalla Xilinx; si può dunque intuire come la soluzione al problema in questione venga spostata dall'aspetto ottico a quello elettronico, provvedendo alla conversione dei fotoni da rivelare in impulsi elettrici misurabili dalla board in uso. La scelta di questa tecnologia può essere motivata dalla possibilità di sfruttare le molteplici risorse hardware disposizionibili e dalla facilità di riconfigurazione del software che implementa la funzione richiesta, caratteristiche non riscontrabili nei dispositivi ASIC (*Application Specific Integrated Circuits*) dedicati, le cui prestazioni sono ottimizzate rispetto ai task del sistema a scapito della flessibilità progettuale post-fabbricazione.

Senza dilungarci oltre, terminiamo questa breve sezione introduttiva elencando i capitoli nei quali questa tesi è suddivisa:

- Cap.1: trattazione delle nozioni di base della meccanica quantistica per introdurre formalmente il concetto di entanglement;
- Cap.2: definizione di entanglement dal punto di vista fisico e distinzione tra possibili metodi di generazione di coppie di fotoni entangled;
- Cap.3: excursus sul time interval measurement: metodologie di misura e differenza tra acquisizione analogica e digitale;
- Cap.4: descrizione dell'FPGA utilizzata: motivazioni della scelta, architettura interna e ambiente di sviluppo software;
- Cap.5: descrizione dettagliata del design del TDC realizzato con analisi dei singoli blocchi che lo costituiscono;
- Cap.6: stima delle prestazioni del sistema mediante test ed elaborazione dei risultati delle misure;
- Cap.7: valutazione del lavoro svolto, prossimi passi, conclusioni.

Capitolo 1

Meccanica Quantistica

NEL SEGUENTE CAPITOLO vengono riportati alcuni concetti fondamentali per introdurre l'universo della meccanica quantistica. Lo scopo è quello di fornire le basi necessarie per comprendere il fine ultimo di questo progetto di tesi: per maggiori approfondimenti si rimanda alla letteratura dedicata.

1.1 Nozioni preliminari e notazione

La teoria quantistica è un modello matematico del mondo fisico. Per definire questo modello, per prima cosa è necessario specificare i formalismi matematici su cui si basa e, quindi, definire i principi fondamentali che lo caratterizzano.

Ricordiamo brevemente alcune definizioni che saranno alla base di tutti i concetti e le relazioni che seguiranno.

Definizione 1.1.1 (Spazio di Hilbert)**.** Uno spazio di Hilbert è uno spazio vettoriale $\mathcal{H}$ nel campo dei numeri complessi dotato di *prodotto scalare interno*[1] $(\mathbf{x} \mid \mathbf{y})$. Tale spazio risulta essere completo[2] secondo la norma indotta dal prodotto scalare: $||\mathbf{x}|| = \sqrt{\mathbf{x} \mid \mathbf{x}}$.

Definizione 1.1.2 (Raggio vettore)**.** Per ogni vettore $\mathbf{x}$ dello spazio di Hilbert si definisce raggio vettore la classe di equivalenza dei vettori che hanno la stessa direzione di $\mathbf{x}$:

$$\forall \mathbf{x} \in \mathcal{H} \Longrightarrow r_x = c\mathbf{x},\ c \in \mathbb{C} \quad \text{e} \quad c \neq 0 \tag{1.1}$$

[1] In uno spazio vettoriale complesso V, una funzione da $V \times V \to \mathbb{C}$ si dice prodotto interno (o scalare) su V se verifica le seguenti proprietà:

1. $(a\mathbf{x} + b\mathbf{y} \mid \mathbf{z}) = a(\mathbf{x} \mid \mathbf{z}) + b(\mathbf{y} \mid \mathbf{z}) \quad \forall a, b \in \mathbb{C},\ \forall \mathbf{x}, \mathbf{y}, \mathbf{z} \in V$;
2. $(\mathbf{x} \mid \mathbf{y}) = \overline{(\mathbf{y} \mid \mathbf{x})} \quad \forall \mathbf{x}, \mathbf{y} \in V$;
3. $x \neq 0 \Longrightarrow (\mathbf{x} \mid \mathbf{x}) > 0$.

[2] Uno spazio di Cauchy V si dice *completo* se le successioni di Cauchy in esso definite sono convergenti.

Definizione 1.1.3 (Stato)**.** Uno stato è un raggio vettore di uno spazio vettoriale di Hilbert. Solitamente viene limitata la definizione ai raggi vettori con norma unitaria $||\psi|| = 1$. In meccanica quantistica, per rappresentare gli stati si usa solitamente la notazione *braket*. Lo stato ψ del sistema viene rappresentato con un vettore colonna di dimensione finita o infinita:

$$|\psi\rangle \tag{1.2}$$

che esprime in modo compatto le componenti dello stato nello spazio (tale scrittura viene chiamata *ket*). In modo analogo si introduce il simbolo *bra*:

$$\langle\psi| \tag{1.3}$$

che rappresenta il vettore trasposto e coniugato di $|\psi\rangle$; tale elemento risulta essere, perciò, un vettore riga.

La definizione di spazio vettoriale di Hilbert e le relative proprietà del prodotto scalare nella notazione presentata diventano:

- I vettori dello spazio vengono rappresentati con la scrittura $|\psi\rangle$;
- Il prodotto scalare assume la forma $\langle\psi|\psi\rangle$[3] e le sue proprietà vengono riscritte come segue:
  - $\langle\varphi \mid (a|\psi_1\rangle + b|\psi_2\rangle)\rangle = a\langle\varphi|\psi_1\rangle + b\langle\varphi|\psi_2\rangle$
  - $\langle\varphi|\psi\rangle = \langle\psi|\varphi\rangle^*$
  - $\psi \neq 0 \implies \langle\psi|\psi\rangle > 0$;
- La norma indotta dal prodotto scalare diventa $||\psi|| = \sqrt{\langle\psi|\psi\rangle}$.

Definizione 1.1.4 (Sovrapposizione)**.** Un sistema fisico che può assumere due stati distinti $|\psi_1\rangle$ e $|\psi_2\rangle$ può trovarsi anche nello stato:

$$|\psi\rangle = a_1|\psi_1\rangle + a_2|\psi_2\rangle \quad , \text{ dove } a, b \in \mathbb{C}. \tag{1.4}$$

che corrisponde alla condizione di sovrapposizione dei due stati $|\psi_1\rangle$ e $|\psi_2\rangle$. Il corrispondente vettore nella notazione *bra* sarà:

$$\langle\psi| = a_1^*\langle\psi_1| + a_2^*\langle\psi_2| \quad , \text{ dove } a, b \in \mathbb{C}. \tag{1.5}$$

Nel caso in cui $|\psi_1\rangle$ e $|\psi_2\rangle$ siano ortonormali e se $|\psi\rangle$ risulta normalizzato si può dimostrare che:

$$|a_1|^2 + |a_2|^2 = 1. \tag{1.6}$$

I termini $|a_1|^2$ e $|a_2|^2$ rappresentano le probabilità di trovare il sistema nello stato $|\psi_1\rangle$ o $|\psi_2\rangle$. Infatti, sebbene per il sistema sia contemplata una sovrapposizione di stati, un eventuale misura porterebbe il suo stato a collassare in $|\psi_1\rangle$ o $|\psi_2\rangle$ con probabilità, rispettivamente, $|a_1|^2$ e $|a_2|^2$.

[3] Pur essendo simile alla scrittura usata all'inizio del capitolo per rappresentare il prodotto scalare $(\mathbf{x} \mid \mathbf{y})$, in questo caso si vuole intendere il prodotto tra il vettore *bra* e il vettore *ket*.

Quanto fin qui enunciato per un sistema a due stati, può essere esteso ad un sistema ad n stati generalizzando le relazioni precedenti come segue:

$$|\psi\rangle = \sum_n a_n |\psi_n\rangle, \tag{1.7}$$

con $|\psi_n\rangle$ ortonormali e

$$\sum_n |a_n|^2 = 1. \tag{1.8}$$

Definizione 1.1.5 (Operatori)**.** Un operatore $\mathbf{A}$ è una funzione lineare dello spazio vettoriale di Hilbert V in V:

$$\mathbf{A}\colon V \longrightarrow V \tag{1.9}$$

$$|\psi\rangle \longrightarrow \mathbf{A}|\psi\rangle \tag{1.10}$$

che gode delle seguenti proprietà, legate alla linearità:

$$\mathbf{A}(|\psi\rangle + |\varphi\rangle) = \mathbf{A}|\psi\rangle + \mathbf{A}|\varphi\rangle \quad \forall |\psi\rangle, |\varphi\rangle \in V \tag{1.11}$$

$$\mathbf{A}(c|\psi\rangle) = c\,\mathbf{A}|\psi\rangle \quad \forall |\psi\rangle \in V \quad \text{e} \quad c \in \mathbb{C} \tag{1.12}$$

Questo ci consente di dire che un operatore che agisce sullo stato di un sistema determina un nuovo stato per il sistema stesso.[4] Gli operatori sono solitamente rappresentati in forma matriciale o attraverso il *prodotto esterno*. Quest'ultima rappresentazione sfrutta il prodotto interno. Siano, infatti, $|v\rangle$ e $|w\rangle$ vettori, rispettivamente, degli spazi vettoriali V e W dotati di prodotto interno. Si può definire $|w\rangle\langle v|$ come l'operatore lineare da V in W che mappa i vettori secondo la relazione:

$$(|w\rangle\langle w|)(|v'\rangle) = |w\rangle\langle v|v'\rangle = \langle v|v'\rangle|w\rangle. \tag{1.13}$$

Definizione 1.1.6 (Variabili dinamiche)**.** Alcune caratteristiche dei sistemi fisici possono essere descritte attraverso gli operatori; questo è il caso delle variabili dinamiche. Quanto detto amplia la definizione di operatore, che oltre a rappresentare un funzionale negli stati, diventa la descrizione di una variabile dinamica di un sistema quantistico. Ogni variabile dinamica sarà associata quindi a un operatore.

Definizione 1.1.7 (Osservabili)**.** Gli osservabili sono particolari variabili dinamiche che possono essere misurate. Un operatore lineare O associato ad una variabile dinamica è osservabile se esso coincide con il suo trasposto coniugato $\mathbf{O}^+$:

$$\mathbf{O}^+ = \mathbf{O}. \tag{1.14}$$

Nel caso degli spazi di Hilbert a dimensione finita, esso coincide con l'operatore aggiunto, per cui scrivendone l'equazione agli autovalori otteniamo:

$$\mathbf{O}|\psi\rangle = a\,|\psi\rangle, \tag{1.15}$$

[4] Va ricordato che la proprietà commutativa non vale per gli operatori lineari. Questa è una differenza sostanziale tra la fisica classica e la fisica quantistica.

che ci consente di scrivere l'osservabile come operatore definito da:

$$\mathbf{O}\colon |\psi_i\rangle = a_i\,|\psi_i\rangle. \tag{1.16}$$

I vettori $|\psi_i\rangle$, detti anche *autovettori*, nella notazione di Dirac prendono il nome di autoket[5], mentre gli scalari a_i sono detti *autovalori*.[6] Questi vettori rappresentano gli *autostati*. Nel nostro caso, trattandosi di un sistema fisico reale, gli autovalori saranno tutti reali. In seguito verranno usati arbitrariamente, con lo stesso significato, i termini autovettori e autostati.

Dalla teoria sugli spazi vettoriali si evince che gli autovettori di una variabile dinamica reale, associata ad autovalori diversi, sono tra loro ortogonali. Questa proprietà ci consentirà di considerare come base del nostro spazio quella costituita dagli autostati di un operatore. Inoltre, ricordando che gli autostati possono essere presi con norma unitaria, è lecito ottenere una base ortonormale (detta *spettro dell'operatore*). In maniera più formale, un vettore dello spazio degli stati del sistema può essere scritto come combinazione lineare degli autokets $\{|\psi_i\rangle\}$:

$$|\psi\rangle = \sum_i c_i\,|\psi_i\rangle. \tag{1.17}$$

Definizione 1.1.8 (Operatori unitari e trasformazione unitaria)**.** Considerando le trasformazioni lineari come matrici, possiamo introdurre le trasformazioni unitarie.

Un operatore lineare $U\colon V \longrightarrow V$, con V spazio di Hilbert, è unitario se soddisfa le seguenti relazioni:

$$\mathbf{U}^+\,\mathbf{U} = \mathbf{I} = \mathbf{U}\,\mathbf{U}^+ \Longleftrightarrow \mathbf{U}^{-1} = \mathbf{U}^+, \tag{1.18}$$

ovvero se il suo inverso coincide con il suo trasposto coniugato. Come vedremo, questi operatori descrivono la dinamica del sistema.

1.2 I postulati della meccanica quantistica

I principi della meccanica quantistica si basano sui seguenti postulati fondamentali:

Definizione 1.2.1 (Postulato 1)**.** A ogni sistema fisico isolato è associato uno spazio vettoriale complesso dotato di prodotto interno (spazio di Hilbert), chiamato *spazio di stato* del sistema. Il sistema è completamente descritto dal suo *vettore di stato*, che è un vettore unitario nello spazio di stato del sistema.

Definizione 1.2.2 (Postulato 2)**.** L'evoluzione di un sistema quantistico chiuso è descritta da una *trasformazione unitaria*. Lo stato $|\psi\rangle$ del sistema al tempo t_1 è legato allo stato $|\psi'\rangle$ del sistema al tempo t_2 da un operatore U che dipende solamente dagli istanti t_1 e t_2:

$$|\psi'\rangle = U|\psi\rangle. \tag{1.19}$$

[5] I vettori *bra* prendono il nome di *autobra*.

[6] Gli autovalori associati agli *autokets* sono gli stessi associati agli *autobras*.

Tale relazione non dà nessuna informazione sullo spazio di stato o su quale operatore U descrive il sistema, bensì assicura solamente che l'evoluzione di un sistema quantistico chiuso può essere descritta in questo modo.
Un'altra definizione più raffinata di questo postulato, che descrive l'evoluzione di un sistema quantistico considerando il tempo come variabile continua, è la seguente:

Definizione 1.2.3 (Postulato 2'). L'evoluzione nel tempo dello stato di un sistema quantistico chiuso è descritta dall'equazione di Schrödinger:

$$i\,\hbar\,\frac{d|\psi\rangle}{dt} = H\,|\psi\rangle. \tag{1.20}$$

Questo ci consente di dimostrare l'unitarietà di tale evoluzione, che si ricava dalle proprietà dell'operatore Hamiltoniano H.[7] Infatti, l'equazione dinamica di Schrödinger può essere espressa come:

$$\frac{d}{dt}\,|\psi(t)\rangle = -i\,H\,|\psi(t)\rangle, \tag{1.21}$$

che riscritta in forma infinitesimale porta alla relazione:

$$|\psi(t+dt)\rangle = (1 - iHdt)|\psi(t)\rangle. \tag{1.22}$$

Compattando la scrittura dell'operatore $1 - iHdt$ con $\mathfrak{R}$, osserviamo che esso risulta essere lineare e unitario, infatti:

$$\mathfrak{R}^{+}\mathfrak{R} = I = \mathfrak{R}\mathfrak{R}^{+}\,. \tag{1.23}$$

Ricordando che il prodotto di operatori lineari è finito, l'evoluzione temporale in un intervallo finito deve risultare, dunque, unitaria:

$$|\psi(t)\rangle = \mathfrak{R}(t)\,|\psi(0)\rangle\,. \tag{1.24}$$

Ne risulta, perciò, che la conoscenza dell'Hamiltoniano del sistema ne descrive completamente la dinamica.

Definizione 1.2.4 (Postulato 3). Le misure quantistiche sono descritte da un insieme $\{M_m\}$ di *operatori di sistema*. Questi operatori agiscono nello spazio di stato del sistema che sta per essere misurato. L'indice m si riferisce al possibile esito della misura. Se il sistema quantistico è nello stato $|\psi\rangle$ immediatamente prima della misura, la probabilità di ottenere il risultato m è data da:

$$p(m) = \langle\psi|\,M_m^{+}M_m\,|\psi\rangle \tag{1.25}$$

e lo stato del sistema dopo la misura risulta:

[7] L'Hamiltoniano è un operatore molto importante nella meccanica quantistica, il suo valore medio corrisponde all'energia del sistema.

$$\frac{M_m|\psi\rangle}{\sqrt{\langle\psi| M_m^+ M_m |\psi\rangle}}\,. \tag{1.26}$$

Gli operatori di misura soddisfano *l'equazione di completezza*:

$$\sum_m M_m^+ M_m = I\,, \tag{1.27}$$

la quale esprime il vincolo di unitarietà della somma delle probabilità:

$$1 = \sum_m p(m) = \sum_m \langle\psi| M_m^+ M_m |\psi\rangle\,. \tag{1.28}$$

Definizione 1.2.5 (Postulato 4)**.** Lo spazio di stati di un sistema fisico composto è il prodotto tensoriale degli spazi di stato dei componenti del sistema fisico. Pertanto, se consideriamo una serie di sistemi numerati da 1 a n, e l'i-esimo sistema è nello stato $|\psi_i\rangle$, allora lo stato complessivo del sistema globale è espresso dalla relazione:

$$|\psi_1\rangle \otimes |\psi_2\rangle \otimes \cdots \otimes |\psi_n\rangle. \tag{1.29}$$

Un concetto chiave nella trattazione dei sistemi composti è l'*entanglement*.

Definizione 1.2.6 (Entanglement)**.** Un sistema composto è definito *entangled* se non può essere scritto come la composizione degli stati dei sistemi che lo costituiscono.

Avremo modo più avanti di approfondire questa nozione, calandola nella realtà di nostro interesse.

1.3 Distinzione degli stati quantistici

Un importante applicazione del terzo postulato della meccanica quantistica è il problema di distinguere gli stati quantistici: supponiamo di avere a disposizione gli stati $|\psi_i\rangle$ con $(1 \leq i \leq n)$. Scegliamo casualmente uno di questi stati, e ipotizziamo di non sapere quale sia. Quello che vorremmo fare è misurare lo stato $|\psi\rangle$ per determinare l'indice i. Nell'ipotesi che gli stati siano ortonormali, potremmo procedere nel seguente modo: definiamo degli operatori di misura $M_i = |\psi_i\rangle\langle\psi_i|$ per ciascun valore dell'indice i, e un operatore M_0 definito come la radice quadrata dell'operatore positivo $I - \sum_{i\neq 0} |\psi_i\rangle\langle\psi_i|$.
Questi operatori soddisfano l'equazione di completezza, perciò se il sistema si trova nello stato $|\psi_i\rangle$ allora $p(i) = \langle\psi_i| M_i |\psi_i\rangle = 1$ e si otterrà il valore i senza possibilità d'errore. Possiamo quindi concludere che gli stati ortogonali possono essere distinti univocamente.
Viceversa, se gli stati non sono ortogonali è possibile dimostrare che non esiste nessuna misura quantistica in grado di distinguerli. Infatti, dati due stati $|\psi_1\rangle$ e $|\psi_2\rangle$ non ortogonali, potremmo scomporre $|\psi_2\rangle$ in una componente parallela e una ortogonale

a $|\psi_1\rangle$. La probabilità di ottenere come esito della misura lo stato $|\psi_1\rangle$ da un sistema nello stato $|\psi_2\rangle$ è non nulla: stati non ortogonali non sono, quindi, distinguibili.
Quanto appena dimostrato può essere riassunto sinteticamente dal seguente teorema:

Teorema 1.3.1 (Indistinguibilità di stati non ortogonali)**.** *Stati quantistici non ortogonali non possono essere distinti senza incertezza.*

Un altro teorema di una certa rilevanza è il teorema di *non clonazione*:

Teorema 1.3.2 (Non clonazione)**.** *Non è possibile eseguire una copia perfetta di ogni stato quantistico.*

1.4 Misure proiettive

Un caso particolare di operatori definiti nel Terzo Postulato riguarda la classe delle misure proiettive. Una misura proiettiva è descritta da un osservabile M, un operatore Hermitiano nello spazio di stato del sistema sotto esame. L'osservabile in questione ha la seguente scomposizione spettrale:

$$M = \sum_m m P_m, \tag{1.30}$$

dove P_m è il proiettore dell'autospazio di M con autovalore m. I possibili esiti della misura corrispondono agli autovalori m dell'osservabile.
Una misura dello stato $|\psi\rangle$ fornisce come risultato m con probabilità:

$$p(m) = \langle\psi| P_m |\psi\rangle. \tag{1.31}$$

Una volta ottenuto l'autovalore m come esito della misura, lo stato del sistema diviene:

$$\frac{P_m|\psi\rangle}{\sqrt{p(m)}}. \tag{1.32}$$

Il valore medio per le misurazioni proiettive può essere determinato come segue:

$$\begin{aligned} E(M) &= \sum_m m p(m) && (1.33)\\ &= \sum_m m\langle\psi| P_m |\psi\rangle && (1.34)\\ &= \langle\psi|\left(\sum_m m\, P_m\right)|\psi\rangle && (1.35)\\ &= \langle\psi| M |\psi\rangle. && (1.36) \end{aligned}$$

La deviazione standard associata all'osservazione di M (il valore medio è scritto nella forma $\langle M \rangle$) risulta quindi:

$$\Delta(M) = \langle (M - \langle M \rangle)^2 \rangle \tag{1.37}$$
$$= \langle M^2 \rangle - \langle M \rangle^2 \tag{1.38}$$

Si ricorda, per completezza, che questo valore descrive come saranno distribuiti gli esiti della misura di M.

1.5 Principio di indeterminazione di Heisenberg

Il principio di indeterminazione di Heisenberg, riveste un ruolo molto importante nella teoria della meccanica quantistica.

In parole povere, esso afferma che esistono coppie di grandezze fisiche che non possono essere misurate simultaneamente con precisione arbitraria (esempio: posizione e quantità di moto di un elettrone).
La trattazione formale si sviluppa analizzando gli operatori associati alle variabili dinamiche.
Due osservabili si definiscono *compatibili* se commutano:

$$AB = BA. \tag{1.39}$$

Definiamo come *commutatore* l'elemento che soddisfa la seguente relazione:

$$[A, B] = AB - BA. \tag{1.40}$$

Per entrambi gli operatori esprimiamo gli scarti della media[8] come segue:

$$\Delta A = A - \langle A \rangle \tag{1.41}$$
$$\Delta A = B - \langle B \rangle. \tag{1.42}$$

La deviazione standard per l'operatore A è data da $\langle (\Delta A)^2 \rangle$, e in modo analogo quella relativa all'operatore B risulta $\langle (\Delta B)^2 \rangle$. Sfruttando la disuguaglianza di Cauchy-Schwarz si ottiene:

$$\langle (\Delta A)^2 \rangle \langle (\Delta B)^2 \rangle \geq ||\langle \Delta A \Delta B \rangle||^2 , \tag{1.43}$$

e sviluppandola in funzione del commutatore si ricava il principio di indeterminazione di Heisenberg:

$$\langle (\Delta A)^2 \rangle \langle (\Delta B)^2 \rangle \geq \frac{1}{4} ||\langle [A, B] \rangle||^2. \tag{1.44}$$

[8] Il valore medio di un operatore O è espresso dalla relazione $\overline{O} = \langle \psi | O | \psi \rangle$; per motivi di compattezza grafica verrà indicato con $\langle O \rangle$.

Se i due operatori non commutano, cioè non sono compatibili, allora $[A, B] \neq 0$; pertanto per entrambe le grandezze non sarà possibile eseguire una misura simultanea con precisione arbitraria. Vale ovviamente il viceversa se i due operatori risultano compatibili.

1.6 Misurazioni POVM

In molte applicazioni è di particolare interesse conoscere la probabilità dell'esito di una misura, soprattutto nei casi in cui è possibile effettuare una sola misurazione. Per l'analisi di queste problematiche risulta conveniente adottare il formalismo matematico della POVM (*Positive Operator Valued Measure*), anch'essa conseguenza del Terzo Postulato della meccanica quantistica.
Siano M_m gli operatori che descrivono una misurazione su un sistema quantistico nello stato $|\psi\rangle$. La probabilità di ottenere l'esito m è data da:

$$p(m) = \langle\psi| M_m^+ M_m |\psi\rangle. \tag{1.45}$$

L'operatore così definito:

$$E_m = M_m^+ M_m \tag{1.46}$$

è un operatore positivo e si può dimostrare che soddisfa le seguenti uguaglianze:

$$\sum_m E_m = I \quad \text{e} \quad p(m) = \langle\psi| E_m |\psi\rangle .$$

L'insieme degli operatori E_m sono sufficienti per determinare le probabilità dei differenti esiti della misura e vengono detti *elementi POVM* associati alla misura stessa. L'insieme completo $\{E_m\}$ è noto come POVM.

Una delle proprietà delle misure di tipo proiettivo è la *ripetibilità*. Per ripetibilità intendiamo il mantenimento dello stesso esito di una misura anche se questa viene ripetuta più volte; ciò implica la non alterazione dello stato in seguito a essa.
Supponiamo che $|\psi\rangle$ sia lo stato di partenza del sistema. Dopo la prima misura lo stato diventerà $|\psi_m\rangle = (P_m|\psi\rangle)/\sqrt{\langle\psi| P_m |\psi\rangle}$. Applicando nuovamente P_m allo stato $|\psi_m\rangle$, esso non verrà perturbato, pertanto avremo $\langle\psi| P_m |\psi\rangle = 1$. Misurazioni ripetute, avranno quindi sempre esito m e non altereranno lo stato.

1.7 Qubit

Il qubit, abbreviativo di *quantum bit*, rappresenta il bit quantistico. Concettualmente a tale definizione si associa il *quanto di informazione*, ovvero la parte elementare più piccola in cui può essere codificata un'informazione. Così come i sistemi classici di computazione si basano sul bit, i sistemi quantistici hanno come unità basilare il qubit.

Sebbene per analogia di nome, bit e qubit possano sembrare intuitivamente simili, questi due concetti si differenziano, in realtà, in modo sostanziale.

Formalmente con qubit si intende lo stato di uno spazio vettoriale di Hilbert bidimensionale V. Scegliendo la base $\{|0\rangle, |1\rangle\}$ per lo spazio V, con $|0\rangle$ e $|1\rangle$ ortogonali, allora un generico qubit $|\psi\rangle$ può essere rappresentato come:

$$|\psi\rangle = a|0\rangle + b|1\rangle \,. \tag{1.47}$$

Un qubit può essere quindi definito come un vettore unitario di uno spazio di Hilbert bidimensionale, in cui $|a|^2+|b|^2 = 1$. Come si può notare, a differenza del bit classico che può assumere solo due valori, a un qubit può essere attribuito un valore qualsiasi nell'infinità di valori intermedi ottenuti a partire dagli stati di base $|0\rangle$ e $|1\rangle$.
Il qubit, essendo codificato attraverso gli stati quantistici di un sistema, gode di tutte le proprietà viste precedentemente per la meccanica quantistica. L'anomalia rispetto al bit classico consiste nella possibilità di assumere infiniti valori derivanti dalla sovrapposizione degli stati di base senza però conoscere, in accordo con i teoremi della meccanica quantistica, i valori di a e b. Una misura sul qubit restituirà lo stato $|0\rangle$ con probabilità a^2 e lo stato $|1\rangle$ con probabilità b^2, e successivamente lo stato del sistema collasserà su uno dei possibili stati di base, divenendo noto.

A livello pratico, alcuni esempi di qubit effettivamente impiegati sono:

- la polarizzazione dei fotoni;
- lo spin degli elettroni;
- lo spin dei nuclei atomici;
- ioni in cavità risonanti;
- componenti microelettronici superconduttori.

1.8 Qubit multipli

Come naturale estensione alla trattazione dei qubit, introduciamo i qubit multipli. A differenza dei sistemi classici di informazione, il passaggio a più qubit non è del tutto intuitivo. Consideriamo inizialmente un sistema di due qubit, dal quale poi ricaveremo le relazioni valide per il caso generale. Come considerazione preliminare, è necessario distinguere due casi:

- i due qubit sono *incorrelati*: operare su uno di essi non influisce sullo stato dell'altro;
- i due qubit sono *correlati*: gli stati sono legati tra loro; inoltre, dalla conoscenza di uno è possibile ottenere informazioni sull'altro.

Da un punto di vista formale, la discussione si sviluppa introducendo il concetto di *prodotto tensoriale* per la descrizione di sistemi composti da più sottosistemi, secondo il postulato (4).
Presi due spazi vettoriali V e W, con rispettive basi $\{v_1, v_2\}$ e $\{w_1, w_2\}$, il prodotto tensoriale di V e W genera uno spazio con la seguente base:

$$\big\{v_1 \otimes w_1,\, v_1 \otimes w_2,\, v_2 \otimes w_1,\, v_2 \otimes w_2\big\}. \tag{1.48}$$

La dimensione di tale spazio risulta quindi:

$$\dim(V \otimes W) = \dim(V) \times \dim(W)\,. \tag{1.49}$$

Conseguentemente, un sistema formato da due qubit, ognuno con la sua base $\{|0\rangle, |1\rangle\}$, ha uno spazio di stati di base:

$$\big\{|0\rangle \otimes |0\rangle,\, |0\rangle \otimes |1\rangle,\, |1\rangle \otimes |0\rangle,\, |1\rangle \otimes |1\rangle\big\}, \tag{1.50}$$

che può essere riscritto nella forma[9]:

$$\big\{|00\rangle, |01\rangle, |10\rangle, |11\rangle\big\}. \tag{1.51}$$

Quanto detto può essere applicato a sistemi di n qubit. Continuando a considerare un sistema quantistico a due qubit, la sovrapposizione diventa:

$$|\psi\rangle = c_0|00\rangle + c_1|01\rangle + c_2|0\rangle + c_3|11\rangle, \tag{1.52}$$

dove, ovviamente, vale la relazione: $|c_0|^2 + |c_1|^2 + |c_2|^2 + |c_3|^2 = 1$.
Proviamo ora ad estendere il processo di misura, ricordando che la misura proietta lo stato di un qubit in uno dei due stati di base, associato al processo di misura stesso. Il risultato ottenuto è probabilistico, e lo stato del sistema dopo la misura diventerà noto, come già detto, con probabilità unitaria. La misura del primo qubit con base $\{|0\rangle, |1\rangle\}$ può essere ottenuto considerando lo stato di partenza espresso nel seguente modo:

$$|\psi\rangle = a|00\rangle + b|01\rangle + c|10\rangle + d|11\rangle, \tag{1.53}$$

e quindi mettendo in evidenza il primo qubit attraverso il prodotto tensoriale, ottenendo:

$$|\psi\rangle = |0\rangle \otimes \big(a|0\rangle + b|1\rangle\big) + |1\rangle \otimes \big(c|0\rangle + d|1\rangle\big) = \tag{1.54}$$

$$\begin{aligned} &= \sqrt{|a|^2 + |b|^2}\,|0\rangle \otimes \left(\frac{a}{\sqrt{|a|^2 + |b|^2}}\,|0\rangle + \frac{b}{\sqrt{|a|^2 + |b|^2}}\,|1\rangle\right) + \\ &+ \sqrt{|c|^2 + |d|^2}\,|1\rangle \otimes \left(\frac{c}{\sqrt{|c|^2 + |d|^2}}\,|0\rangle + \frac{d}{\sqrt{|c|^2 + |d|^2}}\,|1\rangle\right). \end{aligned} \tag{1.55}$$

[9] Nella forma compatta si sostituisce la notazione $|0\rangle \otimes |1\rangle$ con $|01\rangle$.

I vettori dentro parentesi tonde hanno norma unitaria, pertanto la probabilità che una misura del primo qubit dia lo stato $|0\rangle$ è pari al termine sotto radice che precede lo stato $|0\rangle$, e analogamente per l'altro stato di base.

Riassumendo, una misura del primo qubit può dare i seguenti esiti:

- $|0\rangle$ con probabilità $|a|^2 + |b|^2$. Lo stato finale dopo la misura diventa: $|0\rangle \otimes \left(\frac{a}{\sqrt{|a|^2+|b|^2}}|0\rangle + \frac{b}{\sqrt{|a|^2+|b|^2}}|1\rangle\right)$;
- $|1\rangle$ con probabilità $|c|^2 + |d|^2$. Lo stato finale dopo la misura diviene: $|1\rangle \otimes \left(\frac{c}{\sqrt{|c|^2+|d|^2}}|0\rangle + \frac{d}{\sqrt{|c|^2+|d|^2}}|1\rangle\right)$.

1.9 Entanglement di qubit

Nelle pagine precedenti si è messa in luce l'esistenza di sistemi fisici che non possono essere completamente descritti attraverso lo stato dei singoli qubit. Questo è il caso dei *qubit entangled*.
Consideriamo nuovamente un sistema a due qubit, la sua base sarà:

$$\left\{|0\rangle \otimes |0\rangle,\, |0\rangle \otimes |1\rangle,\, |1\rangle \otimes |0\rangle,\, |1\rangle \otimes |1\rangle\right\}. \tag{1.56}$$

Si potrebbe pensare che un generico stato si possa sempre ottenere come prodotto tensoriale delle sovrapposizioni degli stati dei due qubit:

$$\left(a_1|0\rangle + b_1|1\rangle\right) \otimes \left(a_2|0\rangle + b_2|1\rangle\right). \tag{1.57}$$

Se così fosse, un generico stato del sistema sarebbe perciò del tipo:

$$|\psi\rangle = a_1a_2|00\rangle + a_1b_2|01\rangle + b_1a_2|10\rangle + b_2a_2|11\rangle. \tag{1.58}$$

In realtà, questa condizione non è sempre verificata, e un'esempio concreto della non generalità di queste relazioni è descritto dall'entanglement.

Definizione 1.9.1 (Entanglement di qubit)**.** Due qubit si definiscono entangled se non è possibile risalire ai valori dei parametri a_1, a_2, b_1, b_2 che descrivono lo stato di un sistema secondo la relazione:

$$|\psi\rangle = \left(a_1|0\rangle + b_1|1\rangle\right) \otimes \left(a_2|0\rangle + b_2|1\rangle\right). \tag{1.59}$$

Un caso di qubit entangled è il seguente:

$$|\phi\rangle = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle). \tag{1.60}$$

Si noti come lo stato del primo qubit sia strettamente legato a quello del secondo, e viceversa.
Considerando sempre questo caso particolare, se pensiamo di misurare il primo qubit

senza avere in precedenza misurato il secondo, seguendo una procedura formale otteniamo le seguenti probabilità:

$$P[qubit_1 = |0\rangle] \mid qubit_2 \text{ non misurato}] = \frac{1}{2}, \tag{1.61}$$

$$P[qubit_1 = |1\rangle] \mid qubit_2 \text{ non misurato}] = \frac{1}{2}. \tag{1.62}$$

La situazione cambia in modo radicale nel caso in cui il secondo qubit sia già stato misurato con un certo esito. Considerando $|0\rangle$ come risultato della misura, risulta:

$$P[qubit_1 = |0\rangle] \mid qubit_2 \text{ misurato}] = 1, \tag{1.63}$$

$$P[qubit_1 = |1\rangle] \mid qubit_2 \text{ misurato}] = 0. \tag{1.64}$$

Tali espressioni, invertite, permettono di determinare le probabilità relative al primo qubit nel caso in cui il risultato della misura del secondo sia, invece, $|1\rangle$.

Questa premessa teorica legata alla meccanica quantistica ci ha permesso finalmente di introdurre il processo che, tramite questo lavoro di tesi, vogliamo verificare. Se però, grazie a questa base di nozioni e formalismi, tale concetto è stato finora espresso in termini matematici, nel prossimo capitolo verrà trattato il fenomeno fisico e si darà un nome al sistema quantistico descritto in questa sezione.

Capitolo 2

Entanglement

L'ENTANGLEMENT è un fenomeno quantistico che non ha un corrispettivo nella fisica classica. Questa nozione è stata tradotta dal termine tedesco "Verschränkung", introdotto dal fisico e matematico austriaco Erwin Schrödinger nel trattato *The Essence of Quantum Physics* del 1935. L'entanglement si basa sul concetto di *sovrapposizione degli stati*, ovvero quel particolare stato nel quale il sistema in esame si trova in due stati diversi contemporaneamente, non permettendo quindi di determinare quale caratteristica sia predominante. Tale concetto, inserito nel contesto dell'interazione fra due particelle, afferma che il loro stato di superposition non può essere espresso attraverso il prodotto dei singoli stati, dal momento che nella loro descrizione quantistica questi risultano inseparabili. Il comportamento di una coppia di fotoni con la suddetta proprietà, di cui non sia noto lo stato di polarizzazione nè dell'uno nè dell'altro, è tale per cui, appena uno dei due è soggetto a misura, il risultato della stessa risulta completamente casuale e la polarizzazione dell'altro sarà opposta. Appare chiaro, dunque, che si parla di un fenomeno che implica una forte correlazione tra le entità che ne vengono interessate.

2.1 Entanglement in polarizzazione

Sebbene il concetto di entanglement sia generale ed esuli, in linea di principio, dalla forma che assume lo stato rappresentativo delle particelle, è l'entanglement in polarizzazione il fenomeno di gran lunga più interessante, per la sua facilità di realizzazione e l'ampia varietà di strumenti in grado di monitorare questa caratteristica fisica.
Successivamente verranno trattati i principali metodi che permettono di generare fotoni entangled in polarizzazione, tuttavia va fatto preliminarmente uno studio teorico sul loro stato, riallacciandoci in qualche modo all'introduzione formale dell'argomento fatta nel capitolo precedente.
Consideriamo un sistema quantistico costituito da due fotoni, denominati *signal* e *idler*, che si muovono lungo direzioni opposte. Questa assunzione non è casuale perchè permette di trattare i due fotoni come particelle distinte e determina la *non località*

del processo, concetto che approfondiremo più tardi. Il loro stato di polarizzazione sia definito dalla relazione:

$$|\varphi_{EPR}\rangle = \frac{1}{\sqrt{2}}(|V\rangle_s|V\rangle_i + |H\rangle_s|H\rangle_i), \tag{2.1}$$

dove $|V\rangle$ e $|H\rangle$ indicano l'asse di polarizzazione mentre i pedici distinguono tra signal e idler. Lo stato complessivo $|\varphi_{EPR}\rangle$ non può essere fattorizzato come prodotto degli stati di signal e idler, cioè:

$$|\varphi_{EPR}\rangle \neq |A\rangle_s|B\rangle_i, \tag{2.2}$$

con $|A\rangle_s$ e $|B\rangle_i$, rispettivamente, stato di signal e stato di idler qualsiasi.
Questa non fattorizzabilità indica che lo stato di ciascuna delle due particelle non può essere descritto in modo indipendente bensì richiede la conoscenza dell'altro. Due fotoni con tali caratteristiche vengono detti *entangled*, e $|\varphi_{EPR}\rangle$ rappresenta il loro *stato entangled*.
La misura della posizione di signal e idler nella base H, V restituisce una probabilità del 50% nel caso in cui i segnali siano polarizzati entrambi verticalmente o orizzontalmente.
Impiegando invece due polarizzatori ruotati di un certo angolo α, la nuova base di misura risulta:

$$\begin{cases} |V_\alpha\rangle = \cos\alpha|V\rangle - \sin\alpha|H\rangle \\ |H_\alpha\rangle = \cos\alpha|H\rangle + \sin\alpha|V\rangle \end{cases} \tag{2.3}$$

Lo stato complessivo $|V\rangle$ e $|H\rangle$ assume quindi la forma:

$$|\varphi_{EPR}\rangle_\alpha = \frac{1}{\sqrt{2}}(|V_\alpha\rangle_s|V_\alpha\rangle_i + |H_\alpha\rangle_s|H_\alpha\rangle_i). \tag{2.4}$$

Analizzando l'equazione (2.4) si nota che la probabilità che entrambi gli stati siano verticali od orizzontali non è cambiata rispetto al caso precedente, ovvero è ancora pari a $\frac{1}{2}$. Se ne deduce allora che la misura del fotone signal permette di determinare in maniera univoca la polarizzazione dell'idler.
E' rilevabile una certa indeterminazione fra le polarizzazioni nelle differenti basi: la conoscenza della polarizzazione di un fotone in una base fissata (descritta dall'angolo α) non implica, infatti, che questa sia nota in una base differente. Va precisato, tuttavia, che non è la scelta di α a determinare la polarizzazione del fotone idler nello stato $|H_\alpha\rangle_i$ o $|V_\alpha\rangle_i$ ma il risultato casuale della misura della polarizzazione del fotone signal.
Come poc'anzi anticipato, il processo descritto è *non locale*: infatti, sebbene i due fotoni si trovino a grande distanza tra loro, la misura dello stato del fotone signal porta all'immediato collasso del fotone idler in uno stato ortogonale.

2.2 Metodi di generazione di fotoni entangled

Si è visto in precedenza che non esiste un solo tipo di entanglement perchè lo stato che caratterizza un fotone può essere relativo alla frequenza, al numero d'onda, alla polarizzazione. Il fenomeno di nostro interesse è l'entanglement in polarizzazione, e per questo andremo in seguito ad analizzare la sorgente che permette la generazione di fotoni con tale caratteristica. La suddivisione delle possibili sorgenti in base al tipo di entanglement realizzato, tuttavia, è riduttivo dal momento che nel processo di creazione delle coppie di fotoni entangled entrano in gioco altri parametri che non possono essere trascurati. Prima di prendere in considerazione queste differenze, vale allora la pena descrivere il processo generale che prende il nome di *OPDC* (*Optical Parametric Down Conversion*).
Quando un fascio di luce intenso (*pompa*) con frequenza ω_p incontra un cristallo non lineare birifrangente, una parte dei fotoni che lo compongono vengono convertiti in coppie di fotoni (signal e idler) di frequenza ω_s e ω_i. Poichè l'energia e il momento sono conservativi, è necessario che le componenti lungo i tre assi dei campi elettromagnetici in gioco nel processo rispettino le seguenti due relazioni:

$$\omega_s + \omega_i = \omega_p \tag{2.5}$$

$$k_s + k_i = k_p, \tag{2.6}$$

dove k_p rappresenta il vettore d'onda del campo di pompa nel cristallo mentre k_s e k_i sono i vettori d'onda relativi a signal e idler. Queste condizioni descrivono l'entanglement in termini di frequenza e di momento dello stato quantistico dei due fotoni generati. Il fenomeno risulta tale in quanto definisce una sovrapposizione coerente di valori di ampiezza continui per le due particelle che non può essere fattorizzata nei singoli valori relativi a ciascuna di esse.
Introduciamo ora la prima fondamentale distinzione tra le caratteristiche dei fotoni entangled generati tramite OPDC:

- OPDC di *tipo I*, nella quale i fotoni entangled emessi hanno polarizzazione parallela;
- OPDC di *tipo II*, che ha la peculiarità di emettere fotoni entangled con polarizzazione ortogonale.

La seconda differenza sostanziale riguarda la direzione di uscita dei fotoni dal cristallo (vedi figura 2.2). In base all'orientazione di quest'ultimo si distingue tra:

- OPDC *collineare*, caratterizzata dal fatto che signal e idler escono nella stessa direzione;
- OPDC *non collineare*, nella quale signal e idler viaggiano lungo direzioni distinte.

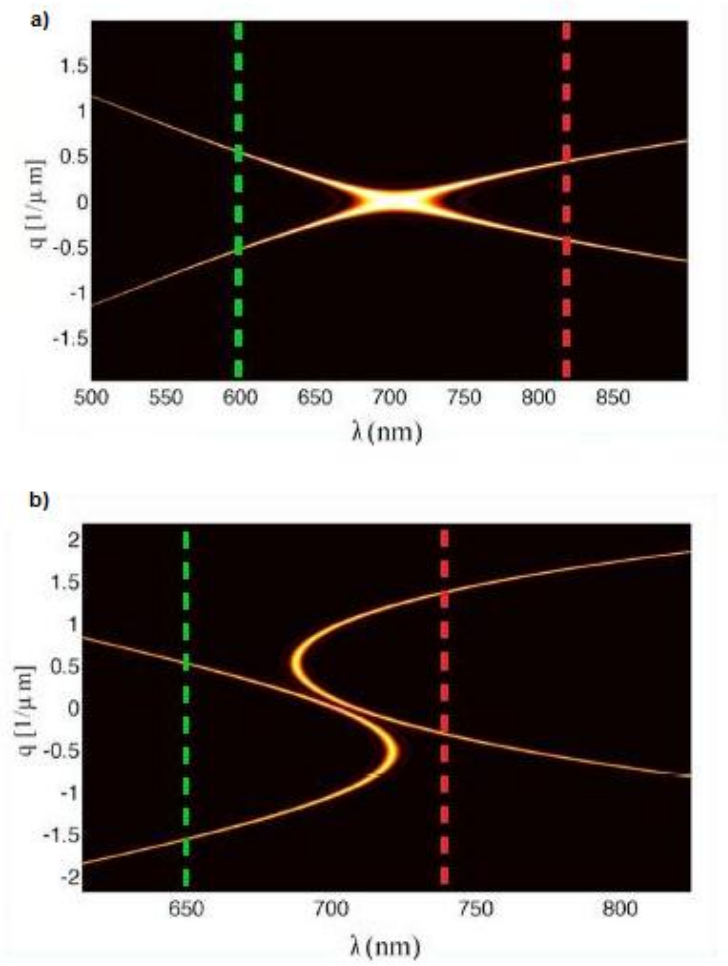


Figura 2.1: Tipi di OPDC. a) Fotoni a polarizzazione parallela (tipo I). b) Fotoni a polarizzazione ortogonale (tipo II).

Figura 2.2: Differenza tra OPDC collineare (a sinistra) e non collineare (a destra).

In figura 2.3 è rappresentato un esempio di setup ottico che realizza il processo di generazione di fotoni entangled mediante un cristallo BBO (*Beta Barium Borate*)

di tipo II. Il fascio di luce generato dal laser incide sul cristallo BBO, il quale induce una frazione di fotoni a decadere in accordo con il principio della parametric down conversion, dando luogo a coppie di fotoni entangled.

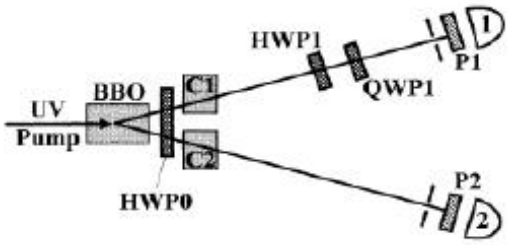


Figura 2.3: Schema a blocchi di una sorgente di fotoni entangled.

2.3 Single Photon Avalanche Diode

L'osservazione di fenomeni di estrema brevità temporale ed energia infinitesima ha come limite ultimo di risoluzione il fotone. L'esigenza di rivelare singoli fotoni trovò supporto strumentale nei primi anni '60 grazie agli APD (*Avalanche Photon Diode*), che dimostrarono una particolare sensibilità nella rivelazione di sorgenti luminose di intensità anche molto bassa. Fu però negli '80, con lo sviluppo delle tecniche di produzione e soprattutto l'avvento degli SPAD (*Single Photon Avalanche Diode*), che gli studi in materia si poterono avvalere di dispositivi di rivelazione disponibili su larga scala.
Con il termine SPAD si indica tipicamente uno strumento in grado di fornire in uscita una corrente alternata in risposta a un fotoelettrone generato all'interno del dispositivo stesso. Dal punto di vista strutturale, uno SPAD non differisce da un qualsiasi altro fotodiodo a valanga, ciò che cambia è la modalità di funzionamento: mentre l'APD opera in regime lineare e alimentazione costante, lo SPAD è composto da un sensore monolitico e da un particolare circuito, detto di *quencing*. L'accoppiamento con un circuito di quencing permette al fotodiodo di operare in configurazione digitale sul singolo fotoelettrone (acceso o spento), consentendo, una volta eseguita la rivelazione, di riprodurre in uscita un segnale in tensione osservabile. Questa modalità di funzionamento viene detta geiger mode.
Da questa breve introduzione si può dedurre facilmente che il ruolo dello SPAD è assimilabile a un contatore di fotoni.

2.3.1 Principio di funzionamento

Gli SPAD sono sostanzialmente giunzioni p-n polarizzate a una tensione V_A superiore alla tensione di breakdown V_Z. Un potenziale del genere per la polarizzazione del dispositivo implica la presenza di un campo elettrico così elevato ($3 * 10^5$ V/cm)

da innescare un *effetto valanga* di portatori nel substrato non appena una singola carica viene iniettata nella regione di svuotamento. Quando ciò avviene, la corrente aumenta molto rapidamente (tempo di salita del fronte nell'ordine del ns) fino a raggiungere un livello costante. Il portatore primario che dà il via al processo viene generato attraverso *fotoionizzazione* e la valanga di portatori che ne scaturisce indicherà, con un ritardo di alcuni ps, l'arrivo di un fotone sul componente sensibile (il fotodiodo) dello SPAD.
Ovviamente, una volta che l'effetto valanga si è verificato, è necessario riportare il dispositivo alla condizione di quiescenza, in modo che il processo possa essere riprodotto nuovamente. Tale compito è affidato al circuito di *quencing* (vedi figura 2.4) che esegue il ripristino dello stato pre-valanga dello SPAD.
Come anticipato, questo regime di funzionamento, illustrato in figura 2.5, è definito geiger mode.

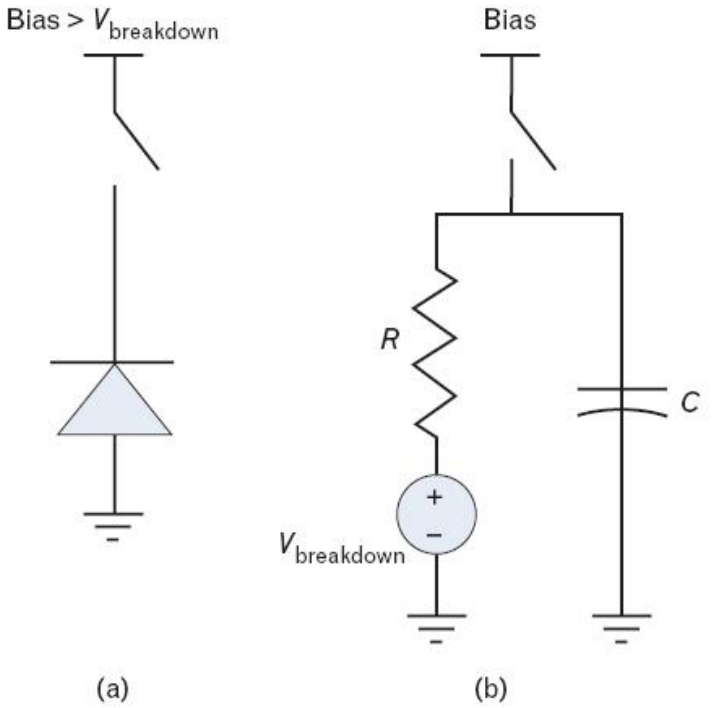


Figura 2.4: Circuito di quencing passivo. (a) In modalità geiger, l'APD viene caricato con una polarizzazione superiore alla tensione di breakdown. (b) Quando l'effetto valanga è stato innescato, l'APD si comporta come il circuito a destra.

Va sottolineato che lo stato di equilibrio raggiunto dalla corrente di portatori liberi (attorno a qualche decina di μA) tende idealmente a mantenersi per un tempo indefinito quando la tensione della regione di carica spaziale si riduce, senza poi subire ulteriori variazioni di rilievo, al valore di breakdown della giunzione, e il rate

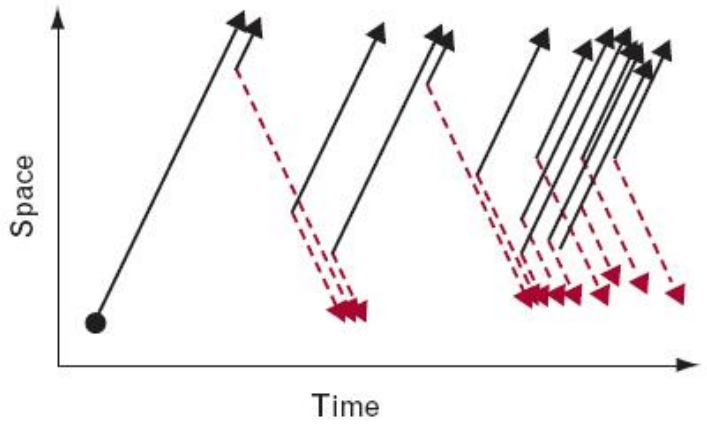


Figura 2.5: Rappresentazione concettuale dell'effetto valanga in modalità geiger.

di generazione di coppie elettrone-lacune liberi coincide con il rate di ricombinazione delle stesse. Infatti, se la corrente per qualche motivo decresce, la caduta di tensione sulla resistenza in serie intrinseca del diodo tende a sua volta a ridursi portando a un incremento del potenziale ai capi della regione di carica spaziale, e di conseguenza un nuovo aumento della stessa corrente dovuto al crescente tasso di fotoionizzazione.

2.3.2 Fotodiodo a valanga

Si è visto nella sezione precedente che APD e SPAD presentano alcune differenze sostanziali. Ciò che li accomuna è l'elemento sensibile che opera la conversione dei fotoni incidenti in una grandezza elettrica misurabile (tensione, corrente, ecc.), poi amplificata per facilitarne la successiva analisi.
L'APD viene polarizzato inversamente applicando una tensione V_A (maggiore della tensione nominale di breakdown V_Z) con un valore di tensione eccedente V_{ecc}, in genere, scelto tra il 10% e il 30% del valore di breakdown, in grado di preservare il dispositivo dagli effetti distruttivi tipici dell'applicazione di una tensione inversa.
L'utilizzo dei circuiti di quencing permette di mantenere il dispositivo in uno stato di attesa prima dell'innesco dell'effetto valanga, permettendo il successivo ripristino delle condizioni iniziali di tensione.
La giunzione p-n che sta alla base del funzionamento del fotorivelatore presenta caratteristiche abbastanza particolari rispetto a una normale giunzione p-n che costituisce un diodo. Le regioni n^+ e p^+ sono molto drogate e presentano campi elettrici di notevole entità a causa dello strozzamento delle bande garantendo le condizioni ideali per l'instaurarsi della valanga di portatori.
Quando un fotone incide e viene assorbito nel semiconduttore, si forma una coppia elettrone-lacuna libera. Ciascun portatore di carica, detto *primario*, tende quindi a diffondere nella zona in cui è maggioritario attraversando la regione di carica spaziale, e viene rapidamente accelerato dall'elevato campo elettrico presente cau-

sando l'effetto valanga di cariche. La capacità di giunzione a questo punto si scarica verso l'esterno, producendo una corrente macroscopica che si mantiene per un tempo estremamente basso, in corrispondenza del ritorno della tensione al valore di breakdown V_Z.

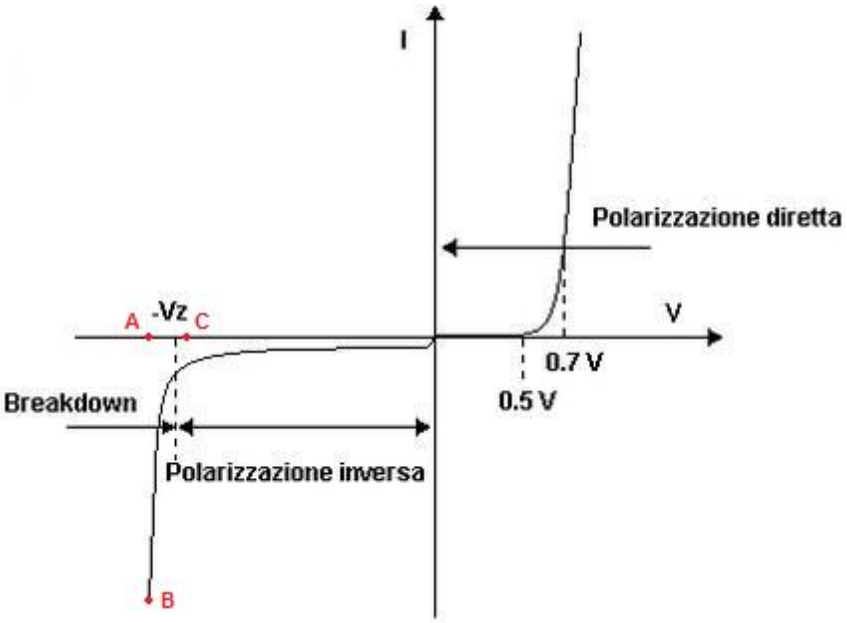


Figura 2.6: Caratteristica V-I di una giunzione p-n.

La caratteristica V-I (tensione-corrente) della giunzione p-n descritta è illustrata in figura 2.6. La transizione dalla regione di breakdown (coincidente con l'applicazione di una tensione inversa in modulo superiore a V_Z, punto B) e la regione di polarizzazione inversa (punto C) rappresenta la fase di quenching o spegnimento della valanga mentre la transizione intermedia tra C e A è detta di *hold-off* ed è caratterizzata dal fatto che la tensione applicata al diodo non è sufficiente a provocare la moltiplicazione di portatori di carica liberi. L'altro passaggio parziale dal punto A a l punto B viene detto, infine, transizione di *hold-on* e comporta il ripristino della tensione del diodo al valore V_A.
I tempi di commutazione della grandezze elettriche in gioco dipendono strettamente dai parametri della giunzione (in particolare, capacità e resistenza interna) e del circuito di quenching.

2.3.3 Caratterizzazione degli SPAD

Per descrivere le prestazioni di uno SPAD vi sono alcuni parametri che vanno sicuramente tenuti in considerazione. A tale scopo, mettiamo sotto la lente di ingrandimento le seguenti caratteristiche:

- efficienza di rivelazione;

- risposta temporale;
- rumore intrinseco;
- afterpulsing.

L'*efficienza di rivelazione* descrive la capacità del dispositivo di emettere un flusso di corrente di breve durata non appena un fotone incide sulla giunzione p-n, in maniera da certificarne l'arrivo per il corretto conteggio. Affinchè un fotone venga rivelato non è sufficiente che sia assorbito nella regione attiva bensì è necessario che la coppia elettrone-lacuna generata sia in grado di dare vita all'effetto di moltiplicazione a valanga. Per questo motivo, l'efficienza di rivelazione dello SPAD può essere pensata come prodotto tra l'efficienza di assorbimento e la probabilità d'innesto della valanga. Se la luce non è perfettamente focalizzata nella zona sensibile l'efficienza di raccolta dei fotoni all'interno dello SPAD si riduce in modo drastico a causa della scarsa efficienza geometrica, data dal rapporto tra l'area sensibile e l'area soggetta a illuminazione.
Una volta raggiunta l'area attiva del dispositivo, il numero di fotoni viene scalato naturalmente in base a un coefficiente di assorbimento η che è funzione della temperatura di lavoro e dell'energia del fotone incidente. Tale coefficiente è definito dalla seguente relazione:

$$\eta = (1 - R)(1 - e^{-\alpha W})e^{-\alpha D}, \tag{2.7}$$

dove $1 - R$ è il coefficiente di trasmissione che presenta la superficie sensibile dello SPAD, $e^{-\alpha D}$ descrive la frazione di fotoni soggetta ad assorbimento geometrico, $1 - e^{-\alpha W}$ rappresenta la percentuale di fotoni realmente assorbiti in uno spessore W di carica spaziale.
Se si utilizzano diodi con bassa tensione di breakdown l'efficienza di innesco tende a crescere molto rapidamente con l'aumentare del potenziale applicato al fotodiodo. Gli SPAD con volume attivo sottile vengono mantenuti in condizioni operative tali da ottenere un'efficienza di rivelazione attorno al $40 \div 50\%$ per lunghezze d'onda comprese tra i 400 e 500 nm, 30% a 650 nm, e poi via via a scalare verso zero tendendo al lontano infrarosso. Per SPAD con regione di svuotamento più ampia (profondità media di giunzione pari a 150 μm) si può arrivare a un valore di efficienza quantica superiore al 60% per una lunghezza d'onda di 500 nm, con banda spettrale che supera il μm.

L'indeterminazione intrinseca introdotta dagli SPAD nelle distribuzioni temporali è legata alla diversa concentrazione di carica nel volume attivo, che può innescare processi di moltiplicazione a valanga differenti a seconda del punto in cui i fotoni vengono assorbiti. Considerando, infatti, una sorgente di luce impulsata ideale, i tempi di risposta di ciascuno SPAD cambiano leggermente a causa della distribuzione di carica non uniforme nella regione di svuotamento della giunzione, dovuta ai difetti di produzione del fotodiodo. In genere, nella zona centrale dell'area attiva si ha una concentrazione maggiore rispetto alle superifici laterali; ciò comporta un tempo di innesco dell'effetto valanga più rapido nel caso in cui il fotone incida nel punto di

mezzo.
Il ritardo tra lo stato di quiescenza e l'innesco dell'effetto valanga coincide con il fronte di salita del segnale di corrente e marca il tempo di arrivo del fotone. I portatori generati al di fuori dell'area attiva si muovono verso la regione di carica spaziale o per diffusione o per minore effetto del campo elettrico, ed è per questo motivo che impiegano un certo tempo per innescare la valanga. Un portatore di carica primario che viaggia con velocità inferiore, quindi, entra in correlazione con l'impulso ottico che lo ha generato con un certo ritardo, introducendo una componente più lenta della distribuzione temporale.

La generazione di coppie elettrone-lacuna liberi, come si è visto, può essere generata dall'assorbimento di un fotone e successivamente amplificata dall'effetto valanga. Nei semiconduttori, tuttavia, lo stesso fenomeno avviene in misura minore naturalmente per effetto termico ed è compensato da un successivo processo opposto di ricombinazione di portatori.
In uno SPAD polarizzato con una tensione prossima al valore di breakdown, i portatori generati per processo termico, se non celermente ricombinati, potrebbero causare un effetto valanga indesiderato. Nonostante il dispositivo sia mantenuto il più possibile lontano da sorgenti luminose, è dunque possibile osservare un certo numero di conteggi spuri, detti *dark count*. L'emissione termica di portatori nella zona di carica spaziale differisce da quella delle zone neutre dal momento che tale probabilità è legata alla densità di carica nel semiconduttore e ai livelli energetici in cui i centri di generazione si collocano. Nelle zone quasi neutre la densità di carica risulta molto minore rispetto agli strati che formano la zona sensibile. Per un semiconduttore intrinseco si assume una probabilità di emissione (di elettroni) pari a:

$$P[emissione_{e^-}] = Ane^{\frac{Egap}{KT}}, \tag{2.8}$$

dove A è un termine supposto indipendente dalla temperatura, n è la densità di carica di un semiconduttore intrinseco, K è la costante di Boltzmann, T la temperatura assoluta. Questa espressione non risulta più valida nel caso di zone fortemente drogate, e questo scostamento rispetto alla formula precedente viene ancor di più accentuato dagli eventuali difetti presenti nel semiconduttore (impurità o droganti che provocano irregolarità nella geometria del reticolo cristallino), i quali fungono da centri di generazione di carica.
Il dark count di uno SPAD può essere sensibilmente ridotto operando a temperature molto basse e riducendo al minimo le imperfezioni del semiconduttore; è possibile, inoltre, contenere il numero di conteggi spuri applicando all'APD un valore molto limitato di tensione in eccesso V_{ecc}; in questo caso però ne risente l'efficienza di rivelazione.

Mentre ha luogo la moltiplicazione a valanga dei portatori liberi, come detto, un'elevata quantità di carica attraversa la regione attiva cosicchè alcuni portatori possono essere catturati dalle trappole energetiche e rilasciati dopo un certo intervallo di tempo. Questo meccanismo di rilascio da parte delle trappole avviene in tempi variabili più o meno rapidi, ed è correlato alla valanga precedentemente in-

nescata. Se un portatore libero trova una caduta di tensione sufficiente può causare nuovamente la valanga: questo fenomeno prende il nome di *afterpulsing*. Esso causa un incremento del dark count e, nel caso di segnale ottico, provoca anche una correlazione spuria con l'impulso elettrico fotogenerato in precedenza, che si ripercuote sulla distribuzione temporale distorcendone la forma. Il fenomeno dell'afterpulsing dipende dalla tensione in eccesso sul fotodiodo, in quanto la probabilità di "intrappolamento"dipende dalla quantità di carica che attraversa la giunzione.
Una soluzione per contenere questo processo indesiderato consiste nel ridurre il tempo di spegnimento della valanga così da diminuire il numero di cariche che attraversano la regione di svuotamento del dispositivo, e successivamente mantenere la tensione di polarizzazione al valore minimo per una durata tale da far tendere a zero la probabilità di innesco da parte di eventuali portatori intrappolati.

Capitolo 3

Time Interval Measurement

A LIVELLO CONCETTUALE, la rivelazione di fotoni entangled generati attraverso il setup ottico descritto nel capitolo 2 risulta abbastanza semplice, dal momento che essi sono correlati e generati nello stesso istante temporale. Nella realtà, individuare con assoluta precisione due eventi che si verificano contemporaneamente, in un determinato istante t, è un problema tutt'altro che banale. Queste difficoltà sono legate soprattutto alle limitazioni fisiche intrinseche degli strumenti di misura. Occorre quindi fissare un margine temporale entro cui considerare i fotoni ricevuti effettivamente entangled.

Prima di entrare "nel vivo"del progetto è utile allora fornire un'infarinatura sui principi e le tecniche alla base della misura di intervalli temporali, detta *Time Interval Measurement* (TIM).
Questa tipologia di misura ha luogo al verificarsi di due eventi: un evento di $START$, che dà inizio al conteggio, e un evento di $STOP$, che ne determina l'esito finale. L'intervallo temporale viene convertito in una word binaria, ed è per questo motivo che spesso ci si riferisce a essa anche con la denominazione di *time-to-digital conversion*.
Originariamente la time-to-digital conversion descriveva misure a interpolazione su un range temporale piuttosto limitato, tipicamente non superiore a $100 \div 200$ ns. Quando il TIM a interpolazione avviene su un intervallo di tempo più ampio (fino alle decine di secondi), si usa spesso il termine *time counter* (TC).
All'interno di questo capitolo verranno presentate varie metodologie di misura basate sul time tagging in circuiti digitali, facendo un'analisi critica e un confronto con le tecniche di acquisizione analogica, ed evidenziando pro e contro che entrambi gli approcci presentano.

3.1 Parametri di misura

Quando si esegue una misura di intervalli temporali, alcune specifiche vanno senz'altro prese in considerazione:

- *minimo range temporale* (MTI): rappresenta il minimo intervallo di tempo (tra START e STOP) che il contatore è in grado di osservare. Nel caso della misura tra un singolo impulso e l'impulso di eco (la cosiddetta *single-shot measurement*), tale valore deve essere non inferiore al periodo di clock;

- *dead time minimo* (DT): è il tempo necessario affinchè il contatore sia in grado di ricevere un nuovo impulso di START dopo l'arrivo dello STOP della misura precedente. fornisce la massima frequenza di ripetizione di un segnale che il sistema è in grado di sostenere;

- *larghezza minima di impulso* (PW): descrive la durata del più breve impulso che il contatore può accettare sia per lo START sia per lo STOP;

- *range di misura* (MR): rappresenta il più lungo intervallo temporale misurabile.

- *non linearità della time-to-digital conversion*: *errore differenziale* (DNL) ed *errore integrale* (INL) introdotti dalla conversione analogico-digitale della misura temporale;

- *risoluzione incrementale* (r): detta anche passo di quantizzazione (q) o *least significant bit* (LBS), è la variazione minima di ampiezza temporale per cui cambia di un bit il risultato della parola binaria di misura;

- *velocità di lettura in uscita*: fornisce informazioni su quanto velocemente lo strumento può fornire un risultato; questo parametro ha particolare importanza quando vengono realizzate misure in modo continuo a un elevato rate e con lettura "al volo".

3.2 Tecniche di misura

Il più semplice metodo per misurare un intervallo temporale richiede l'uso di un contatore in grado di sfruttare un clock di riferimento a una data frequenza f_0 (periodo $T_0 = 1/f_0$). In questo caso la risoluzione (LSB) risulta pari appunto al periodo del clock utilizzato dal sistema. Dopo un iniziale reset, l'impulso di START abilita il contatore per la durata T stabilita dall'arrivo dello STOP, ottenendo la misura $T_p = nT_0$, dove n rappresenta l'equivalente decimale del numero di cicli di clock risultanti dal conteggio.
Quando si impiega la tecnica di misura mediante semplice contatore si assume che gli intervalli misurati T siano asincroni e dunque, nè inizio nè fine dell'intervallo siano correlati nel tempo con gli impulsi di clock. In altre parole, vi è una distribuzione di probabilità uniforme dell'intervallo temporale che intercorre tra il fronte attivo dell'impulso di clock e l'inizio (e la fine) del conteggio che dà T come risultato. In questo caso il massimo errore di quantizzazione di una singola misura è pari a $\pm T_0$, a seconda di T e della posizione dell'intervallo rispetto al riferimento del clock (vedi figura 3.1).

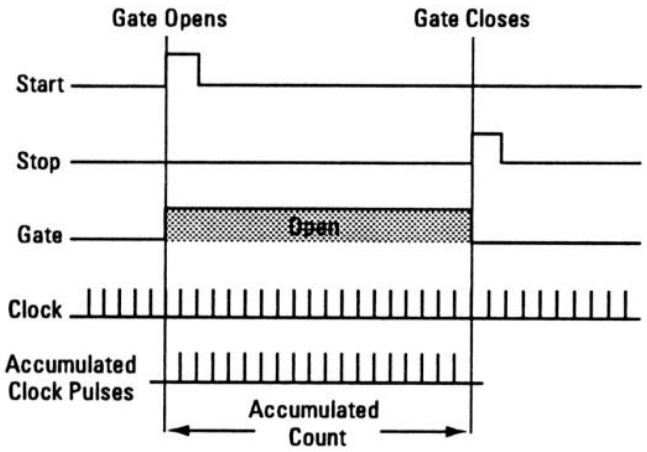


Figura 3.1: Principio di misura di un coarse counter.

In questa sezione verranno presentati i metodi di base (senza interpolazione) che impiegano TDC con un range di misura relativamente breve e con una precisione molto migliore rispetto a quanto si può ottenere con i cosiddetti *coarse counter*.
La precisione di questi TDC dipende principalmente dalla non linearità (integrale e differenziale) della conversione in digitale, analogamente a quanto succede nei classici ADC (*Analog-to-Digital Converter*). Come regola di base, tali sistemi sono progettati in modo tale da rispettare la seguente condizione:

$$INL_max < LSB. \tag{3.1}$$

Iterando la misura di un intervallo temporale T che rimane costante, l'errore casuale s osservato risulta causato soprattutto dal *jitter* associato al circuito elettronico utilizzato, ovvero dalla fluttuazione temporale degli impulsi di clock che non cadono a intervalli esattamente regolari come ci si potrebbe aspettare. Il valore di s può essere fissato per un dato T (tipicamente $s < 10$ ps) oppure essere presentato nel caso peggiore, con s_max contenuto nel range di misura del TDC.
Facciamo ora una carrellata dei metodi di misura più comuni, alcuni dei quali saranno poi approfonditi nei prossimi paragrafi:

- *time stretching* (analogico) seguito dal metodo contatore (digitale);
- doppia conversione tempo/ampiezza (analogico) con successiva conversione analogico/digitale;
- metodo *Vernier* con due oscillatori avviabili (digitale);
- conversione tempo/digitale utilizzando *tapped delay line* (digitale);

- metodo *Vernier* con una linea di ritardo differenziale costituito da due *tapped delay line* (digitale);

Come si può notare, per ciascuna tecnica di misura è stato indicato se appartiene alla categoria analogica o digitale. Ciascun metodo, tuttavia, può essere a sua volta applicato secondo due modalità: senza coarse counter, ottenendo un TDC con range di misura breve, e con coarse counter seguito da interpolatori (comunemente chiamato *time converter*), permettendo una misura di ampiezza temporale più ampia. Un time converter è composto da un coarse counter binario e uno o due TDC con basso range di misura ed elevata risoluzione, garantendo, quindi, una valutazione dell'intervallo "profonda "e allo stesso tempo molto precisa.
Un'eccezione alla regola è rappresentata dal metodo di Vernier, nel quale si fa uso di contatori in grado di assicurare entrambe le proprietà appena descritte. Di questo parliamo nel sottoparagrafo seguente.

3.2.1 Metodo Vernier

Il primo metodo di conversione tempo/digitale fu realizzato nei primi anni del diciassettesimo secolo e prese il nome di metodo di Vernier, dall'ingegnere francese Pierre Vernier che formalizzò per primo questa tecnica di misura di intervalli temporali. Attualmente questo metodo viene utilizzato per realizzare stretching temporale con sistemi digitali, e nella sua configurazione base consta di due oscillatori avviabili (*SG1 e SG2*), un circuito delle coincidenze (*CC*), due contatori (*CTR1* e *CTR2*) e due flip-flop di tipo D, come illustrato in figura 3.2.

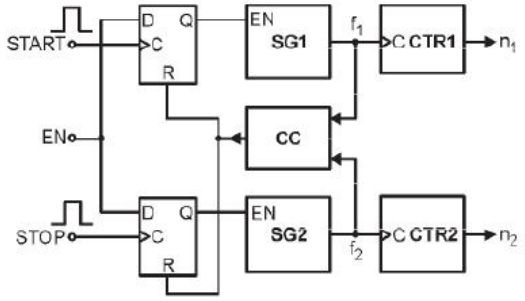


Figura 3.2: Principio di misura del metodo Vernier.

I due oscillatori avviabili generano segnali a frequenze, rispettivamente, $f_1 = 1/T_1$ e $f_2 = 1/T_2$ che differiscono solo leggermente l'una dall'altra. La risoluzione incrementale del sistema dipende proprio da tali frequenze, dal momento che vale la relazione:

$$r = t_1 - t_2. \tag{3.2}$$

L'inizio della forma d'onda prodotta da ciascun oscillatore è allineato con il fronte attivo del relativo segnale di ingresso (START e STOP). La conversione risulta completata quando la coincidenza dei fronti attivi degli impulsi prodotti dai generatori

è rilevata dal circuito delle coincidenze e i due contatori associati hanno immagazzinato i numeri n_1 e n_2. Ignorando l'errore di quantizzazione, il risultato della misura si ottiene dalla relazione:

$$T = (n_1 - 1)\,T_1 - (n_2 - 1)\,T_2 = (n_1 - n_2)\,T_1 + (n_2 - 1)r. \tag{3.3}$$

Il concetto, dunque, è quello di individuare l'istante in cui cadono entrambi i fronti attivi dei segnali prodotti dai due oscillatori e, noti i periodi di oscillazione e i risultati dei contatori, ricavare il valore della misura dalla differenza tra gli intervalli di tempo compresi tra i segnali di ingresso e il segnale prodotto dal circuito delle coincidenze. Quando $T < T_1$, allora $n_1 = n_2$ e $T = (n_2$ - $1)r$. L'uso di un solo contatore, CTR2, è in questo caso sufficiente. Il più lungo tempo di conversione risulta pari a $n_2 max\, T_2 = T_1\, T_2/r$.
Il metodo Vernier permette di ottenere risoluzioni inferiori a 100 ps, e, con particolari accorgimenti nel design, tale valore può essere drasticamente abbassato fino a meno di 1 ps. Punti focali per ottenere misure di buona qualità attraverso questa tecnica sono la stabilità e la precisione degli oscillatori avviabili, obiettivo spesso non semplice da raggiungere, specie quando si vogliono misurare intervalli di ampia durata.

3.2.2 Conversione time-to-digital mediante tapped delay line

Un altro metodo concettualmente semplice per la misura di intervalli temporali si basa sull'impiego di *tapped delay line*. Si tratta di una linea composta da un certo numero di celle di ritardo, ciascuna delle quali introduce idealmente lo stesso ritardo di propagazione τ. La misura temporale si ottiene campionando lo stato della linea durante la propagazione di un impulso di START; tante più uscite della linea verranno rilevate attive tanto maggiore sarà l'ampiezza dell'intervallo sotto test.
L'evoluzione della tecnologia dei semiconduttori ha permesso lo sviluppo di sistemi basati su linee di ritardo molto avanzati, già a partire dai primi anni '80. I nuovi TDC integrati utilizzano le catene di ritardo unitamente a *phase-locked loop* (PLL) o *delay-locked loop* per ottenere elevata stabilità e calibrazione.
Le tapped delay line possono essere usate in varie configurazioni. La scelta più semplice prevede che la linea di ritardo venga creata con una catena composta da N celle contenenti latch in modalità iniziale "trasparente"(STOP a livello logico alto) con START a '0'. Il fronte di salita dell'impulso di START si propaga attraverso la serie di latch consecutivi finchè non si verifica il fronte di discesa dell'impulso di STOP, il quale fissa lo stato di tutti gli elementi di memoria campionando lo stato corrente della linea e interrompendo la propagazione. La misura dell'intervallo temporale è data dalla somma dei ritardi introdotti da tutti i flip-flop che hanno immagazzinato il livello logico alto, ovvero è pari a $T = k\tau$, dove k è il latch più avanzato nella linea di ritardo ad aver immagazzinato lo stato logico HIGH. I dati ottenuti sono descritti da un codice termometrico e, dunque, è necessaria una conversione a codice naturale o binario BCD.
La linea di ritardo si può ottenere anche ponendo in cascata una serie di buffer, caratterizzati, come nel caso precedente, da ritardo costante τ. Come si può notare in

figura 3.3, lo stato della linea viene campionato in corrispondenza del fronte di salita dell'impulso di STOP e mantenuto nei flip-flop di tipo D edge triggered. Il risultato della misura è ancora determinato dalla posizione più alta del flip-flop contenente uno stato logico alto, e non può prescindere dal fatto che i buffer che compongono la catena devono introdurre tutti lo stesso ritardo temporale.

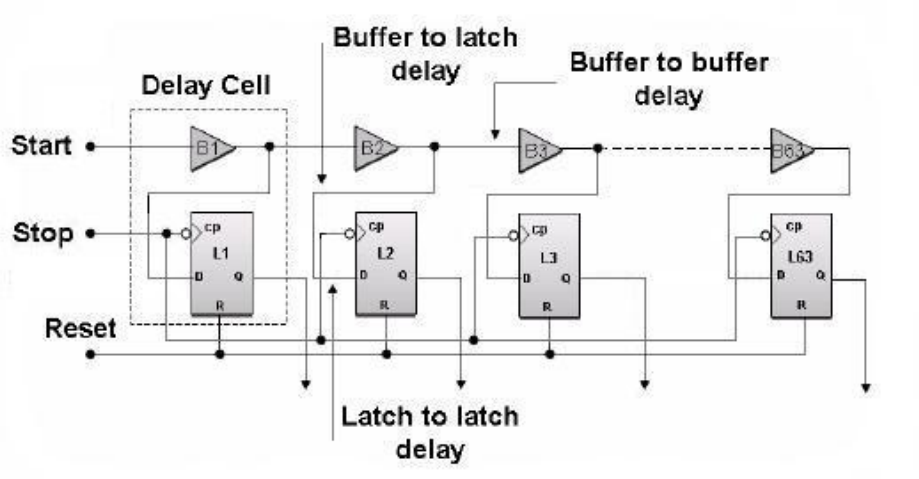


Figura 3.3: Tapped delay line differenziale composta da flip-flop di tipo D per il segnale di START e buffer per il segnale di STOP.

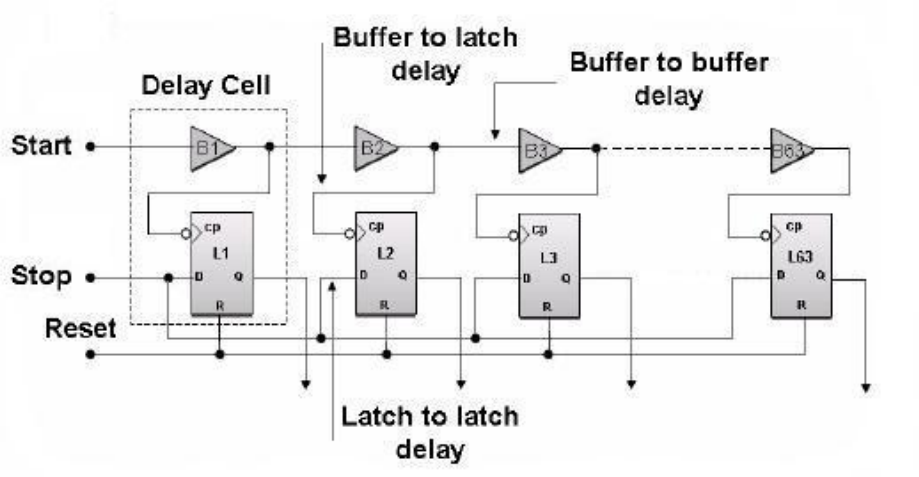


Figura 3.4: Tapped delay line differenziale composta da buffer per il segnale di START e flip-flop di tipo D per il segnale di STOP.

Se nella configurazione appena descritta gli ingressi del clock (C) e dei dati (D) dei flip-flop vengono scambiati tra loro, si ottiene il circuito mostrato in figura 3.4.

La linea, con questa modifica, opera come un clock multifase campionando lo stato dell'ingresso STOP. Quando si verifica un impulso di STOP, il primo fronte di clock successivo porta l'uscita del flip-flop allo stato HIGH e, finchè non si disattiva, viene innescato tramite logica addizionale anche il flip-flop successivo, portando anche la sua uscita a livello alto dopo un ritardo τ, e così via. In questo caso è la posizione più bassa della catena di flip-flop che ha immagazzinato un '1' logico a determinare il risultato della misura.
Le tecniche di "delay line"appena descritte rappresentano una conversione tempo/digitale diretta, ovvero non necessitano di processing intermedio. L'operazione di campionamento avviene in un tempo di conversione trascurabile e, per questo motivo, tali convertitori vengono definiti *flash TDC*.
Ignorando il tempo di lettura dell'uscita, il dead time del circuito (a) è uguale al tempo necessario per resettare tutti i latch della linea. Quando la linea è resettata in modo seriale (impostando a livello logico basso il segnale di START), il dead time è $N\tau$, ma quando si usa un reset parallelo (agendo separatamente sul reset di ciascun latch) il dead time diventa trascurabile. Usando gli ingressi di reset separati dei flip-flop dei circuiti, anche (b) e (c) presentano un dead time trascurabile.
Si potrebbe puntualizzare che l'uso di tapped delay line per la misura di intervalli temporali sia equivalente all'uso di veloci contatori pilotati da un clock stabile. Ad esempio, la linea composta da latch che introducono un ritardo $\tau = 2$ ns è equivalente a un contatore avente clock di frequenza 500 MHz. Il numero di latch N nella linea è comunque molto maggiore del numero n di flip-flop del contatore equivalente ($N = 2n$). Il range di misura può altresì essere allargato molto più facilmente usando l'approccio "contatore". Per raddoppiare tale range in questo caso bisogna aggiungere un solo flip-flop (passando da un numero totale di n a $n+1$), mentre la lunghezza della linea dovrebbe essere raddoppiata (da N a $2N$).
Una buona risoluzione del TDC può essere ottenuta usando due linee di celle aventi ritardo leggermente diverso, credo cioè una linea di ritardo differenziale. Questa soluzione è solitamente realizzata utilizzando dispositivi ASIC (*Application-Specific Integrated Circuit*), tuttavia la tecnologia FPGA CMOS risulta più adatta per questo tipo di applicazioni, anche nel rapporto prezzo/prestazioni.

Riferendoci a quest'ultima categoria di TDC, il circuito che assolve al ruolo di codificatore temporale (illustrato in figura 3.5) risulta composto da due linee di ritardo da N celle e da un decoder in uscita. Ogni cella di ritardo contiene nella prima catena un latch L, che introduce un ritardo τ_1, mentre nella seconda un buffer non invertente B, avente intervallo $\tau_2 < \tau_1$. L'intervallo temporale da misurare è, al solito, definito dal fronte di salita del segnale di START al fronte di salita del segnale di STOP, ed è codificato nella prima linea di ritardo impostando a livello logico alto l'uscita Q dell'ultima cella in cui il cambiamento dell'ingresso C (L $\rightarrow$ H) è davanti a un uguale transizione dell'ingresso D. Il massimo tempo di conversione con questa configurazione è pari a $N\tau_1$. Affinchè sia effettivamente solo l'ultima cella che ha subito una transazione a livello HIGH a fornire l'informazione sul risultato della misura, ogni cella il cui ingresso C cambia stato passando a '1' genera un segnale

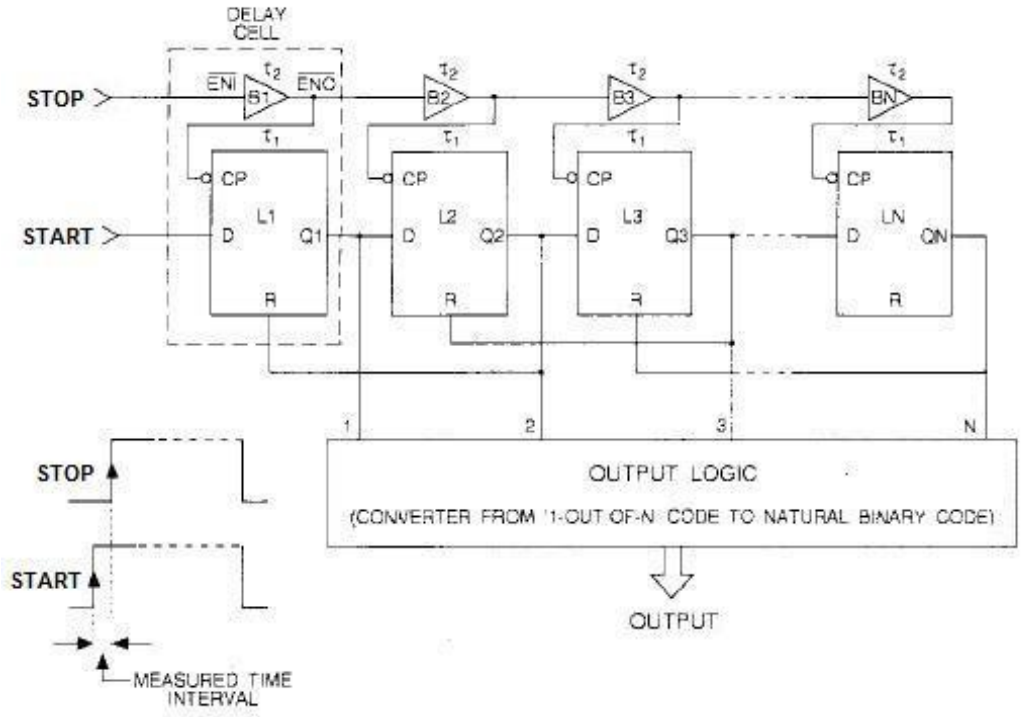


Figura 3.5: Schema di una tapped delay line di Vernier.

di reset collegato all'input della cella precedente all'interno di un anello di feedback locale. In questo modo, tutte le celle prima dell'ultima vengono "pulite"e l'uscita della linea è fornita come codice "*1-out-of-N*". Non è necessario un ingresso di reset separato perchè nello stato iniziale la linea, essendo costituita da latch aperti (STOP a livello logico basso), risulta trasparente all'ingresso START = LOW.
La progettazione di un TDC preciso implementato su tecnologia FPGA non è semplice perchè richiede, al fine di ottenere basso errore di linearità, il test di molti *trial-and-error* design prima di raggiungere risultati davvero soddisfacenti.
Si può notare che la tecnica appena descritta appare molto simile al metodo di Vernier con due generatori avviabili considerando come periodi di oscillazione T_1 e T_2 i ritardi τ_1 e τ_2 delle due linee. E' per questo motivo che la linea differenziale è anche chiamata *Vernier delay line*, e il convertitore associato TDC con linee di ritardo Vernier.
L'utilizzo dell'FPGA per la misura di intervalli temporali ha portato allo sviluppo di TDC con precisioni di centinaia di ps, che verranno di sicuro ulteriormente migliorate con l'incremento progressivo delle prestazioni di questi dispositivi. Il principio per raggiungere livelli di risoluzione così bassi consiste nello sfruttare il ritardo di riporto intrinseco della logica delle slice presenti in ciascun CLB dell'FPGA, creando un sistema di tapped delay line con campionamento simultaneo dello stato.

3.2.3 Metodi a interpolazione

I metodi a interpolazione vengono impiegati quando si vuole misurare un intervallo temporale esteso in durata con una risoluzione elevata. Dal momento che si vuole una misura con due caratteristiche che tipicamente non vanno a braccetto tra loro, entra in gioco il concetto di misura "grezza"e misura "fine"e il risultato finale è dato dalla combinazione di questi due valori: per ottenere un range di misura ampio si fa uso di un coarse counter pilotato da un clock di riferimento (LSB = T_0) mentre l'alta risoluzione è assicurata da interpolatori molto precisi.
Per definizione, l'interpolazione è una tecnica che permette di determinare un valore approssimato di una funzione all'interno di un range limitato da due valori della funzione stessa. Riferendoci all'argomento in questione, quando un evento temporale T_x avviene tra due stati successivi, genericamente n e $n + 1$, di un coarse counter, allora $T_x/T_0 = n_x + c_x$, dove il "contenuto"del coarse counter $n_x = Int(T_x/T_0)$ non è altro che la parte intera del rapporto T_x/T_0 e $c_x = Frc(T_x/T_0)$ rappresenta la rispettiva parte frazionaria, misurata attraverso l'interpolatore.
L'intervallo T misurato con l'uso del metodo a interpolazione viene scomposto in tre intervalli: un intervallo (che può essere abbastanza lungo) è misurato in real time dal coarse counter mentre gli altri due, di breve durata, sono individuati all'inizio e alla fine dell'intervallo (più precisamente, all'arrivo degli istanti di START e di STOP) e vengono misurati da uno o due interpolatori. Ogni interpolatore contiene un sincronizzatore che genera un breve intervallo (di solito compreso tra T_0 e $2\,T_0$) misurato da un TDC fine con breve range di misura (tipicamente $2\,T_0$). Il TDC adotta uno dei metodi di conversione precedentemente descritti e garantisce un'elevata risoluzione ($LSB = T_0/K$, dove K varia tra 10 e 10^4). Se viene utilizzato un oscillatore avviabile per il coarse counter, l'interpolazione relativa al segnale di START non è necessaria ($c_START = 0$). Nel caso in cui l'interpolazione avvenga sia all'inizio sia alla fine dell'intervallo temporale, si parla di *doppia interpolazione*. In alcuni casi tale termine viene impiegato per riferirsi a un singolo interpolatore che esegue l'operazione in due passi successivi facendo uso di circuiti elettronici distinti; per evitare ambiguità, a questo modo di procedere verrà dato il nome di *interpolazione a due stadi*. L'interpolazione a due stadi può essere realizzata consecutivamente attraverso due circuiti posti in cascata (*interpolazione seriale*) o in maniera simultanea utilizzando due circuiti in parallelo (*interpolazione parallela*).
Il metodo a interpolazione doppia venne introdotto inizialmente con l'uso dei metodi Vernier e del digital time stretching. Il metodo Baron, chiamato anche "dual Vernier", consisteva nell'impiego di due oscillatori free-running stabilizzati da un PLL. Gli oscillatori potevano essere fermati momentaneamente e poi fatti ripartire in fase con l'inizio degli intervalli temporali interpolati. I PLL venivano attivati automaticamente quando i loro phase-detector individuavano concidenze temporali, garantendo così che il sistema fosse in grado di fornire elevata precisione (20 ps) e esteso range di misura (10 s).
Un metodo divenuto molto popolare grazie alla sua relativa semplicità di progettazione e i bassi costi di realizzazione, senza tuttavia penalizzare la qualità del-

la misura, è il metodo di Nutt (vedi figura 3.6). L'intervallo temporale T viene scomposto in tre parti:

$$T = n_c T_0 + T_A - T_B, \tag{3.4}$$

dove n_c è il conteggio del coarse counter che opera in real time. Gli intervalli temporali T_A e T_B sono misurati tra il fronte dell'impulso in ingresso e l'impulso di clock appena seguente. I segnali interni ST e SP servono a generare il segnale di abilitazione al conteggio "grezzo". Quando si ricorre ai time stretcher, gli intervalli T_A e T_B vengono per così dire "stirati"di un numero totale di step di ritardo dato, rispettivamente, dai parametri K_A e K_B, e quindi contati per ottenere il numero effettivo di passi n_A e n_B percorsi dai segnali. Definendo le risoluzioni associate ai due stretching temporali come $\tau_A = T_0/K_A$ e $\tau_B = T_0/K_B$, otteniamo:

$$T_A = n_A \tau_A e T_B = n_B \tau_B. \tag{3.5}$$

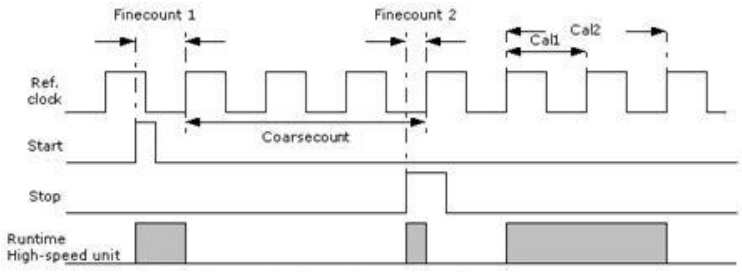


Figura 3.6: Metodo a interpolazione di Nutt.

3.2.4 Incertezza dei contatori a interpolazione

La misura di intervalli temporali attraverso contatori a interpolazione è sostanzialmente affetta da tre principali sorgenti di errori:

- non linearità degli interpolatori
- errore di quantizzazione
- jitter.

Molto spesso gli impulsi di START e STOP in ingresso sono asincroni o comunque non correlati nel tempo con il clock di riferimento. L'errore di non linearità, dunque, è una funzione dipendente dall'intervallo misurato T che al variare di T risulta periodica di periodo T_0 principalmente a causa della non linearità del range di misura breve degli interpolatori dei TDC.
L'errore di quantizzazione associato al metodo a interpolazione è dato dalla differenza tra gli errori di quantizzazione prodotti dagli interpolatori dei segnali di START e STOP. Per un dato intervallo T che giunge in maniera asincrona rispetto al clock,

l'errore indotto dall'interpolatore relativo allo START segue una distribuzione uniforme. Gli eventi di STOP sono tipicamente correlati nel tempo con gli eventi di START; considerando il caso teorico più semplice nel quale entrambi gli interpolatori abbiano gli stessi valori di conversione K, lineari e interi, l'errore di quantizzazione può assumere un valore positivo o negativo, con le seguenti probabilità normalizzate:

$$P_1(\eta_x \geq \eta_c) = 1 - \eta_c \tag{3.6}$$

e

$$P_2(\eta_x \leq \eta_c) = \eta_c, \tag{3.7}$$

dove $\eta_x = Frc(Kx)$ con $0 \leq \mathrm{x} \leq 1$, $\eta_c = Frc(Kc)$, $c = Frc(T/T_0)$ e $K = T_0/LSB$.

Ciò significa che la frazione c viene decomposta in K_c passi di quantizzazione di lunghezza pari all'LSB. L'errore di quantizzazione si verifica soltanto nell'ultimo step. L'errore ha media nulla, ma la sua deviazione standard dipende fortemente da η_c o dall'intervallo misurato T:

$$\sigma = LSB\sqrt{(1-\eta_c)\eta_c}. \tag{3.8}$$

La massima deviazione standard $\sigma = 0.5LSB$ si ottiene per $\eta_c = 0.5$, mentre la deviazione standard media risulta:

$$\sigma_a v = \frac{\pi LSB}{8} \simeq 0.39LSB. \tag{3.9}$$

Il *jitter* è un fenomeno di una certa rilevanza nella valutazione delle performance di un circuito e può essere definito come lo spostamento nel tempo dei fronti di salita e di discesa dei segnali che lo popolano rispetto alla situazione ideale (vedi figura 3.7). La sua misura per un segnale, denominata *jitter picco-picco*, si calcola considerando la differenza tra il massimo ritardo e il massimo anticipo del fronte.

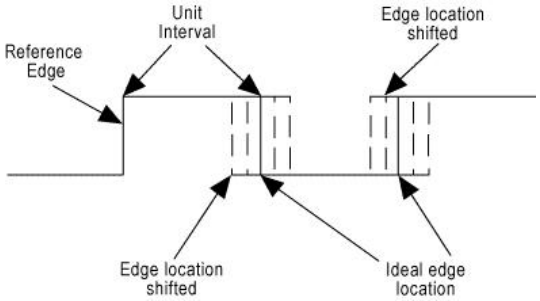


Figura 3.7: Jitter su un segnale periodico.

L'errore complessivo dovuto al jitter è causato dal rumore intrinseco ai componenti utilizzati, dal contributo di jitter dei segnali esterni (incluso il clock) e dal rumore indotto dall'ambiente (compresa l'alimentazione del sistema). Il contributo

del jitter generalmente non dipende da c e crea un "floor level"quasi sempre inferiore a 10 ps (RMS).
Quindi, quando si misura $\sigma(c)$, con $0 < c < 1$, si può distinguere il livello di jitter, pressochè costante, e l'errore variabile causato dalla non linearità e dalla quantizzazione.

3.3 Confronto tra acquisizione analogica e digitale

Come si è visto, metodi di acquisizione analogica e digitale presentano caratteristiche diverse; ciò fa sì che in taluni casi sia privilegiato l'uso dei primi e in altri quello dei secondi, in base all'applicazione a cui si fa riferimento.
Le tecniche di misura digitali sono in generale preferibili per la loro facilità di implementazione nella tecnologia integrata e per la loro stabilità dagli agenti esterni, ottenuta attraverso circuiti PLL o DLL. Al contrario, i TDC analogici soffrono maggiormente le variazioni di temperatura ambientale, richiedendo perciò calibrazioni frequenti, e hanno un tempo di conversione più alto, tuttavia garantiscono migliori risoluzioni rispetto ai digitali (ordine del ps).
Per entrambe le categorie la risoluzione può essere migliorata, cambiano, invece, le modalità affinchè ciò avvenga: nei sistemi analogici si mediano gli intervalli temporali mentre nei sistemi digitali si opera l'interpolazione delle misure. L'operazione di media negli intervalli temporali assume che i fattori che limitano la risoluzione dei sistemi analogici siano casuali e che misure multiple costanti tendano ad annullare tali fattori portando a un incremento della precisione. Il processo di interpolazione comporta il calcolo di valori stimati attraverso dati discreti e genera un effetto positivo sulla risoluzione intaccato solo dal rumore.
Complessivamente si può dire che, sebbene possano essere molto precisi e automatizzati usando algoritimi adattivi avanzati, i TDC analogici appaiono oggi dispositivi arretrati, ed è quindi prevedibile che nella stragrande maggioranza delle applicazioni saranno soppiantati dai più moderni contatori digitali.

Capitolo 4

FPGA

PER REALIZZARE il sistema di time-tagging richiesto dal progetto si è scelto di affidarsi alla tecnologia delle FPGA (*Field Programmable Gate Array*). Si tratta di schede programmabili prodotte su larga scala per le applicazioni più disparate; per meglio comprendere, dunque, i perchè dell'utilizzo di questi dispositivi, si intende fornire in questo capitolo una descrizione esaustiva delle loro caratteristiche hardware e della piattaforma software su cui basa la loro programmazione, nonchè fare un confronto con le altre soluzioni tecnologiche offerte dal mercato.

4.1 Programmable Logic Device

Le schede FPGA rientrano nella più vasta categoria dei PLD (*Programmable Logic Device*), ovvero quei componenti elettronici la cui funzione logica viene stabilita attraverso una procedura di programmazione post-fabbricazione. I PLD sono generalmente divisi in due sottoclassi: i CPLD (*Complex Programmable Logic Device*) e gli SPLD (*Simple Programmable Logic Device*), a seconda che abbiano un numero di pin rispettivamente maggiore o minore di 48. Il ruolo dell'FPGA è divenuto predominante con la crescente necessità di dispositivi che potessero rappresentare un buon compromesso tra le peculiarità dei circuiti integrati programmabili, versatili e veloci da progettare e realizzare ma limitati a funzioni relativamente semplici, e dei circuiti integrati per applicazioni specifiche, i cosiddetti ASIC (*Application Specific Integrated Circuits*), in grado di risolvere problemi di complessità molto maggiore a scapito però della fase di progetto e fabbricazione, più lunga e costosa, e della riconfigurabilità del dispositivo, non più possibile. Le prime FPGA furono proposte da Xilinx negli anni '80, adottavano la tecnologia CMOS con memoria di riconfigurazione SRAM e riprendevano l'architettura dei primi CPLD, utilizzando però un maggior numero di blocchi logici di complessità inferiore organizzati in una griglia di interconnessioni con matrici di commutazione agli incroci. La struttura base di FPGA e CPLD è tuttora abbastanza simile, tanto che genericamente i due termini vengono spesso confusi; ci sono in realtà alcune differenze a livello strutturale:

- *complessità*: per i CPLD tipicamente varia da 1 kgate a 100 kgate, per le FPGA da 10 kgate a 10 Mgate;
- *granularità*: nei CPLD è presente un numero limitato di blocchi logici programmabili (da 1 a 100) di dimensioni maggiori (alcune decine di ingressi), viceversa, nelle FPGA vi sono più blocchi logici (da 100 a 100000) meno complessi (da 2 a 6 ingressi e 1 o 2 registri) che richiedono, tuttavia, una griglia di interconnessioni più articolata;
- *ritardi*: i tempi di propagazione per quanto riguarda i CPLD dipendono in maniera preponderante dai blocchi logici mentre il contributo delle interconnessioni può essere calcolato in maniera abbastanza accurata a priori; al contrario, nel caso delle FPGA prevale il ritardo dovuto alle interconnessioni e quindi non può essere stimato in modo preciso in fase progettuale a causa della forte dipendenza dal piazzamento dei blocchi nel circuito;
- *tecnologia di programmazione*: per i CPLD va per la maggiore l'EEPROM/FLASH mentre per le FPGA è solitamente ad antifusibile o SRAM.

4.2 Motivazioni della scelta dell'FPGA

Nel valutare quale tecnologia fosse più adatta a realizzare il sistema, è stato necessario tener conto dei pro e dei contro che intrinsecamente ciascuna soluzione presentava. Concentrando la nostra analisi sui dispositivi ASIC e FPGA, è risultato chiaro come questi ultimi fossero maggiormente flessibili per le nostre esigenze, consentendo la riprogrammazione del chip a disposizione e la simulazione del suo comportamento in ogni fase del progetto. In questo modo è stato possibile apportare continue modifiche al software, modifiche divenute essenziali per correggere errori e malfunzionamenti imprevisti e affinare le funzionalità del sistema per ottimizzarne l'efficienza. Ovviamente questo ragionamento è stato fatto tenendo conto della natura provvisoria del progetto; diversa sarebbe stata la valutazione se il sistema definito in ogni suo aspetto fosse stato destinato alla produzione su larga scala: in tal caso l'impiego di circuiti integrati dedicati si sarebbe fatto preferire per l'abbattimento dei costi.

4.3 Xilinx Virtex-6 ML605

Dopo aver fatto una panoramica sulle tecnologie che consentono di implementare applicazioni come quella in discussione, e spiegati i perchè della scelta dell'FPGA come migliore soluzione al problema, andiamo nel dettaglio dell'hardware utilizzato per realizzare il TDC.
La board a cui si è fatto riferimento è la ML605 con chip Virtex-6 prodotta dalla Xilinx, azienda leader nella progettazione e vendita di dispositivi logici programmabili basati su semiconduttori. Questa piattaforma fornisce, oltre all'FPGA vera e propria, tutta una serie di componenti utili per l'acquisizione, la memorizzazione e

Numero	Componente
1	FPGA Virtex-6
2	DDR3 SODIMM
3	128 Mb Platform Flash XL
4	Linear BPI Flash
5	System ACE CF controller, connettore CF
6	Cavo JTAG (USB Mini-B)
7	Generatore di clock
8	Porte GTX RX/TX
9	PCIe Gen1 (8-lane), Gen2 (4-lane)
10	Porta e connettore SFP
11	Ethernet (10/100/1000) con interfaccia SGMII
12	USB Mini-B, USB-to-UART bridge
13	Connettori periferiche USB-A Host, USB Mini-B
14	Connettore Video - DVI
15	IIC NV EEPROM, 8 Kb
16	LED di stato
17	User I/O
17f	Display LCD a 16 caratteri x 2 linee
18	Switch
19	Connettore FMC - HPC
20	Connettore FMC - LPC
21	Gestione alimentazione
22	Connettore interfaccia System Monitor
23	System ACE Error DS30 LED

Tabella 4.1: Lista dei componenti hardware della board ML605.

l'elaborazione dei dati, nonchè l'hardware necessario per la comunicazione con altri dispositivi quali, ad esempio, un'altra scheda programmabile o un PC. Entrando più nel dettaglio, l'hardware messo a disposizione dalla board è illustrato in tabella 4.1 (vedi anche figura 4.1).

Questa ampia varietà di componenti permette all'FPGA di essere uno strumento in grado di adattarsi alle più disparate tipologie di applicazioni, secondo la filosofia propria dei dispositivi *general purpose*. Ovviamente le risorse funzionali al nostro progetto rappresentano solo una minima parte del potenziale disponibile. Il nucleo fondamentale della board Virtex-6 è il chip montato su di essa; è infatti grazie all'FPGA vera e propria che è possibile elaborare e immagazzinare i segnali provenienti dalle varie periferiche, sfruttando gli elementi che la costituiscono al suo interno. Andiamo allora ad approfondire questo aspetto, descrivendo in successione le caratteristiche e l'utilizzo dei blocchi logici, delle look-up table e degli elementi di memoria.

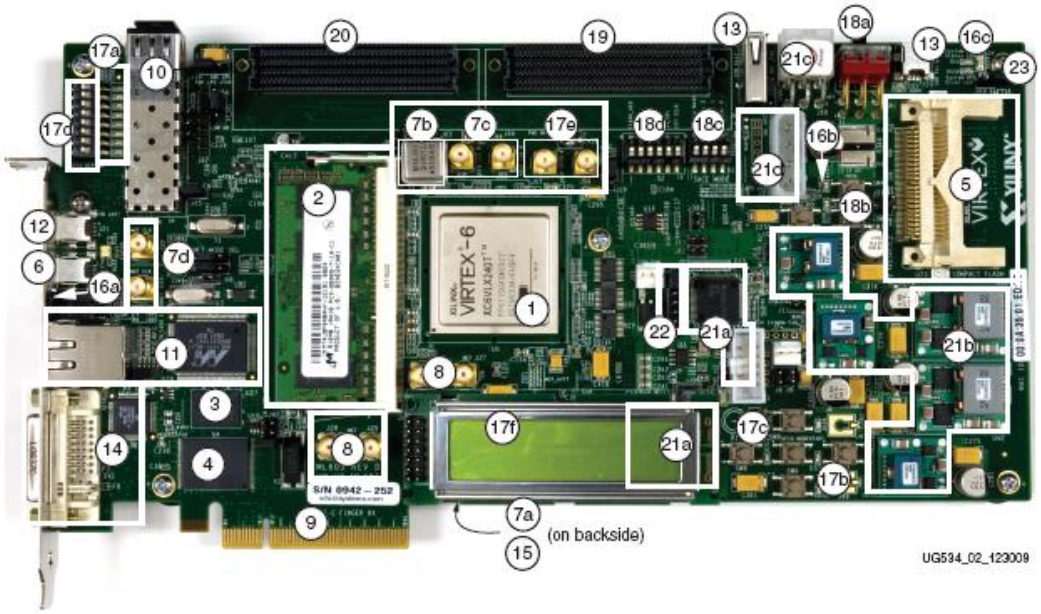


Figura 4.1: Vista dall'alto della board Virtex-6 ML605.

4.3.1 Configurable Logic Block

Con l'acronimo CLB (*Configurable Logic Blocks*) si fa riferimento alla risorsa indispensabile per l'implementazione di circuiti sequenziali e combinatori all'interno dell'FPGA: i blocchi logici. Un FPGA è sostanzialmente organizzata in una matrice quadrata centrale di blocchi logici e in una serie di blocchi periferici che si interfacciano con l'esterno. Un reticolo di piste elettriche garantisce che qualsiasi elemento sia raggiungibile da un altro; è il progettista che, in base alle proprie esigenze, decide come sfruttare blocchi e interconnessioni: in particolare, programmare un blocco logico significa determinare la funzione logica a cui sarà adibito, programmare un blocco di interfaccia equivale a specificare la direzione del traffico di segnali che lo coinvolgono mentre programmare una pista elettrica consente di realizzare la connessione fisica tra i blocchi appena citati.

Ogni blocco logico contiene due *slice*, organizzate in colonne e prive di connessioni dirette tra loro (vedi figura 4.2). Una slice presenta al suo interno quattro generatori di funzioni logiche (*look-up table*), otto elementi di memoria, una serie di multiplexer aventi funzioni diverse e la logica di carry. A queste risorse di base, proprie delle cosiddette *SLICEL* (la cui struttura interna è illustrata in figura 4.3), si aggiungono funzionalità supplementari per una categoria particolare di slice: le SLICEM (vedi figura 4.4). Si tratta di slice di complessità maggiore che supportano la memorizzazione dei dati tramite l'utilizzo di RAM distribuita e logica di scorrimento con registri a 32 bit.

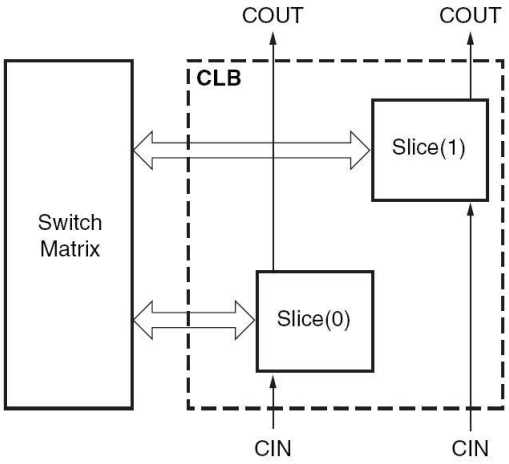


Figura 4.2: Struttura interna di un CLB.

4.3.2 Carry chain

Le slice sono provviste di logica di riporto (*carry logic*) dedicata al fine di ottenere veloci operazioni aritmetiche di addizione e sottrazione. I CLB della Virtex-6 dispongono di due catene di riporto (*carry chain*) distinte che possono essere poste in cascata al fine di ottenere logiche di addizione/sottrazione più estese.
Seguendo lo schema di figura 4.5, ogni carry chain opera dal basso verso l'alto, con un'ampiezza di 4 bit per slice. Per ogni bit di uscita è associato un multiplexer (MUXCY) e una porta XOR dedicata per la somma/sottrazione degli operandi. Il cammino di riporto dedicato e il multiplexer possono anche essere utilizzati in cascata alle LUT per implementare funzioni logiche più estese.
Un blocco di carry chain viene anche definito *CARRY4*. Una CARRY4 presenta 10 ingressi indipendenti (S0 ÷ S3, DI1 ÷ DI4, CYINIT e CIN) e 8 uscite indipendenti (O0 ÷ O3 e CO0 ÷ CO3). Gli ingressi S sono utilizzati per assegnare il segnale che dovrà propagarsi lungo la catena di carry; per consentire questa operazione l'uscita del generatore di funzioni O6 viene connessa a S. Gli ingressi DI vanno a uno dei due ingressi del multiplexer gestito dal segnale S, e i segnali che giungono a essi provengono a loro volta dall'uscita O5 di una LUT oppure dagli ingressi di bypass (AX, BX, CX, DX) di un'altra slice. CYINIT è l'ingresso CIN corrispondente al primo bit della carry chain e può assumere i valori '0' per il calcolo di una somma o '1' nel caso di differenza, oppure essere collegato all'ingresso AX in modo da rendere dinamico il primo bit della carry. Le uscite O contengono la sintesi delle operazioni di addizione e sottrazione, mentre i vari CO restituiscono il riporto di uscita per ogni bit. Per realizzare sequenze di carry più lunghe si utilizza l'ingresso CIN e l'uscita COUT: a ogni CIN di una CARRY4 è collegato il COUT della precedente mentre ciascun COUT viene posto in ingresso al CIN della carry successiva.
Il ritardo di propagazione per un sommatore aumenta linearmente in funzione del

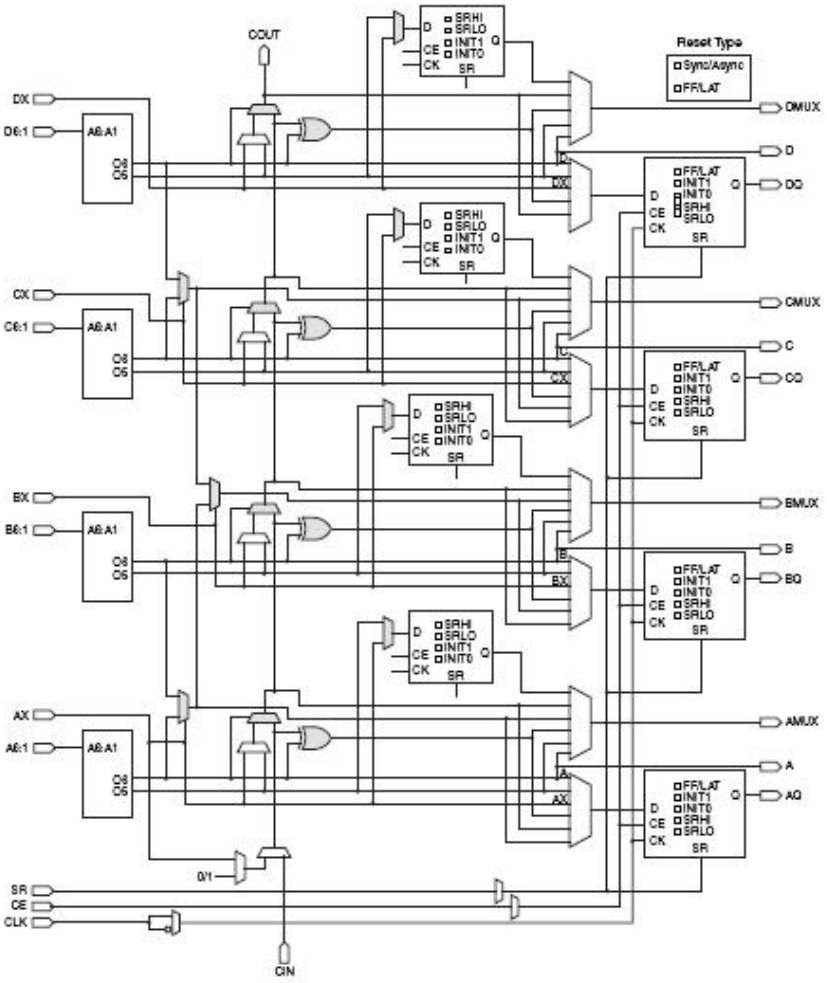


Figura 4.3: Struttura interna di una SLICEL.

numero di bit coinvolti e, dunque, al crescere del numero di slice connesse.

4.3.3 Look-Up Table

La generazione di funzioni logiche viene realizzata tramite look-up table, ovvero strutture dati la cui consultazione (in inglese, appunto, *lookup*) permette di risparmiare operazioni di calcolo a runtime che presentano una maggiore complessità temporale. Accedere a un valore salvato in memoria, infatti, risulta spesso più veloce rispetto all'elaborazione dati, garantendo così un guadagno temporale anche significativo. Il principio mediante il quale attraverso le look-up table vengono realizzate funzioni logiche consiste nell'associare ad ogni ammissibile combinazione di dati in ingresso una corrispondente configurazione di dati in uscita, non necessariamente univoca. Nel chip Virtex-6 questi elementi presentano 6 ingressi (A1 $\div$ A6) e 2 uscite (O5

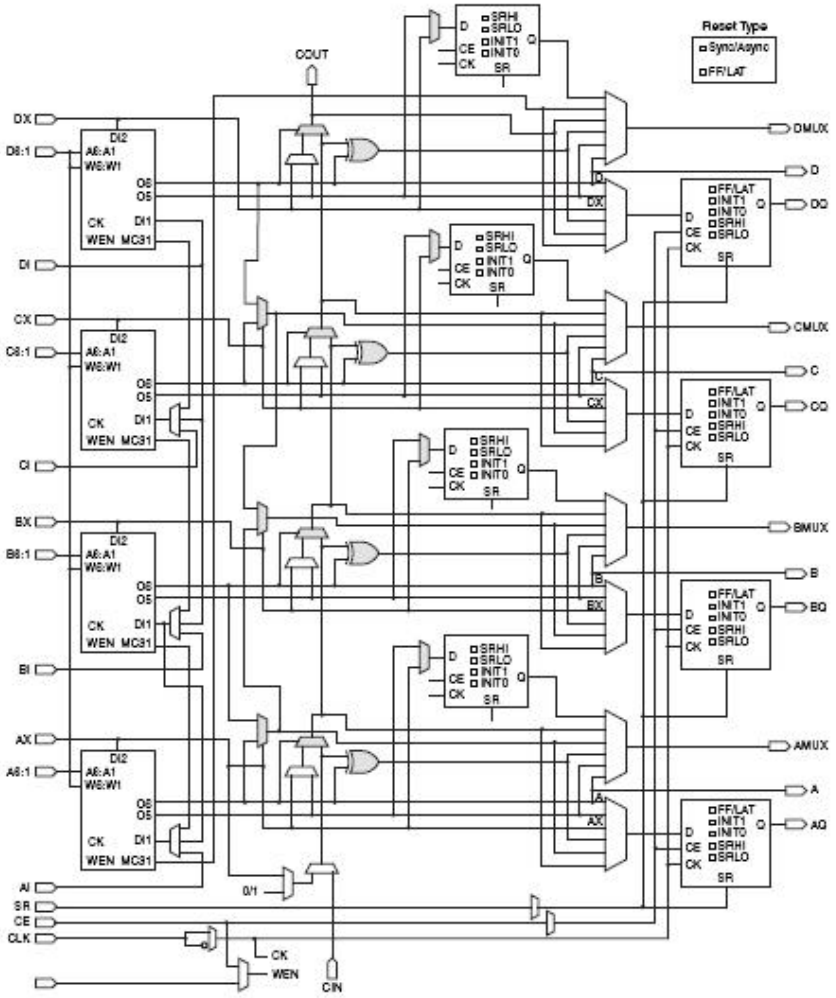


Figura 4.4: Struttura interna di una SLICEM.

e O6) indipendenti per ogni slice, permettendo di implementare qualsiasi funzione booleana a 6 ingressi o, in alternativa, coppie di funzioni a 5 ingressi che condividano però le entrate in comune (solo in questo caso le due uscite sono entrambe utilizzate). Il tempo di propagazione attraverso una LUT è indipendente dalla funzione che viene realizzata.
Come si può vedere in figura 4.3, i segnali provenienti dalle LUT possono essere prelevati direttamente dalle porte dedicate (A, B, C, D per l'uscita O6; AMUX, BMUX, CMUX, DMUX per l'uscita O5), entrare nella porta XOR dedicata dall'uscita O6, fungere da ingresso per il multiplexer della carry chain attraverso l'uscita O5 o ancora essere convogliati come ingresso D nell'elemento di memoria (flip-flop). Un'ulteriore via percorribile è rappresentata dai multiplexer F7AMUX e F7BMUX, che assieme a un terzo componente F8MUX consentono la combinazione di un massimo di 4 LUT

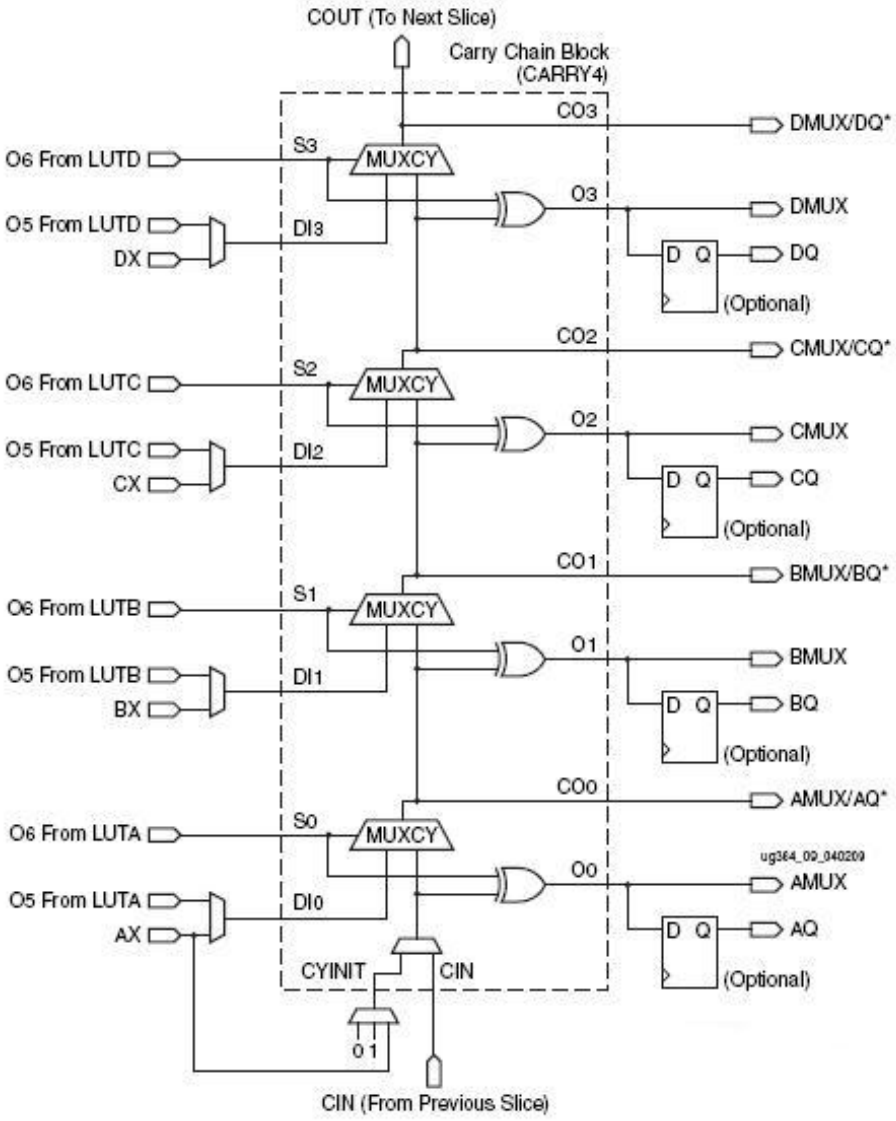


Figura 4.5: Struttura della carry chain di un CLB.

per l'implementazione di funzioni, rispettivamente, a 7 o 8 ingressi nella stessa slice. Non sono presenti connessioni dirette tra slice per formare generatori di funzioni a più di 8 ingressi all'interno del blocco logico, tuttavia è possibile realizzare comunque lo scopo utilizzando slice distinte.

4.3.4 Elementi di memoria

I quattro elementi di memoria presenti nelle slice della Virtex-6 (illustrati in figura 4.6) possono essere configurati come flip-flop di tipo D sensibili al fronte di salita o discesa (*edge-triggered*) oppure come latch sensibili al livello logico (*level-sensitive*). L'ingresso D può essere collegato direttamente all'uscita di una LUT attraverso i multiplexer AFFMUX, BFFMUX, CFFMUX e DFFMUX, oppure si può bypassare la logica interna della slice connettendolo direttamente con gli ingressi AX, BX, CX, DX. Quando vengono configurati come latch, tali elementi di memoria risultano *trasparenti* se il clock di sistema è nello stato logico basso. I segnali di controllo (CLK, CE, SR) sono comuni a tutti gli elementi di memoria all'interno della slice. Quando un flip-flop presenta il segnale SR o il segnale CE attivo, gli altri flip-flop utilizzati nella slice avranno anch'essi lo stesso segnale abilitato; solo il clock ha polarità indipendente e ogni inverter posizionato su tale segnale viene automaticamente assorbito. I segnali CE e SR sono attivi alti. Il primo forza l'elemento di memoria in uno stato specificato dall'attributo SRHIGH o SRLOW: SRHIGH a livello logico alto in corrispondenza di segnale SR attivo, mentre SRLOW a livello logico basso. SRHIGH e SRLOW possono essere settati individualmente per ogni elemento di memoria presente in una slice, mentre, al contrario, ciò non è possibile per la scelta del tipo di set/reset (SRTYPE), sincrono (SYNC) o asincrono (ASYNC).

Lo stato iniziale dopo la configurazione è definito dagli attributi distinti INIT0 e INIT1. Di default, attivando SRLOW viene settato INIT0, attivando SRLOW, invece, INIT1. Nel chip Virtex-6, tuttavia, è possibile slegare questa associazione tra attributi settando INIT0 e INIT1 in modo indipendente da SRLOW e SRHIGH.
Le opzioni di configurazione per le funzionalità di set/reset per i registri e gli elementi di memoria in grado di operare come latch sono:

- nessun set o reset;
- set sincrono;
- reset sincrono;
- set asincrono (*preset*);
- reset asincrono (*clear*).

4.4 Ambiente di sviluppo progettuale e tool di simluazione e analisi

Il software di programmazione per la board Virtex-6 è ISE (*Integrated Synthesis Environment*), un pacchetto fornito dalla stessa compagnia che produce il chip, la Xilinx. Grazie a esso è possibile usufruire di una serie di strumenti per la progettazione e l'analisi di circuiti digitali con l'implementazione su FPGA e CPLD, il tutto mediante un'interfaccia grafica semplice e intuitiva.

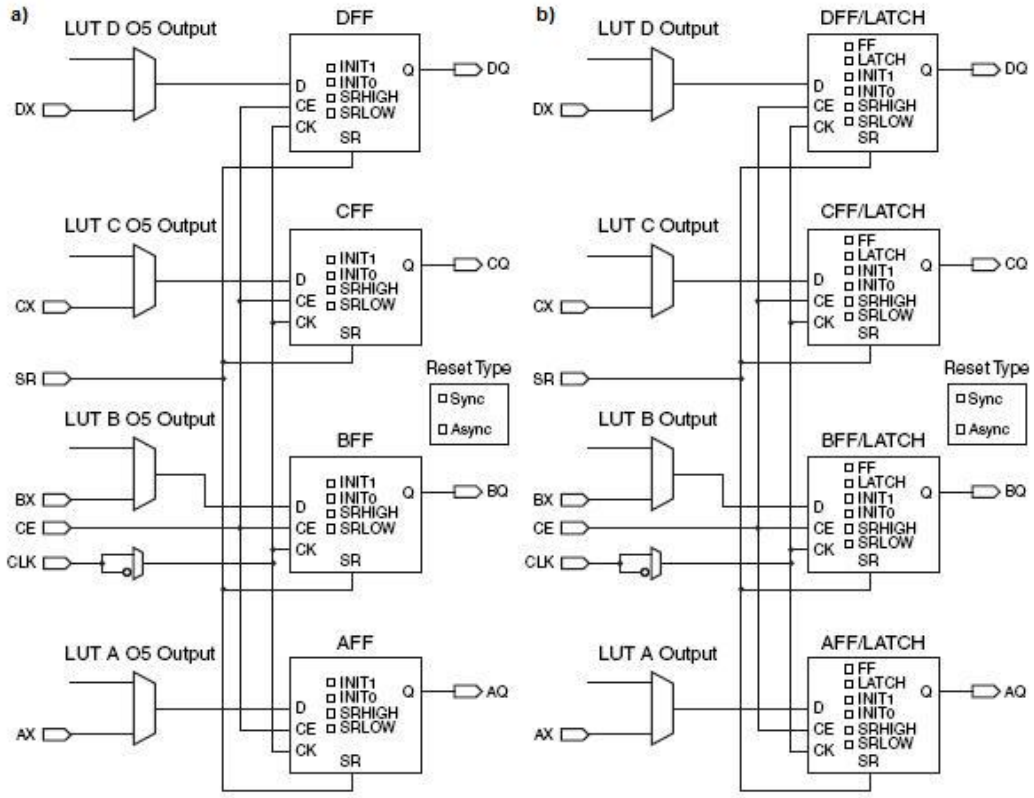


Figura 4.6: Configurazioni degli elementi di memoria in una slice: (a) soli registri, (b) registri/latch.

4.4.1 Linguaggio VHDL

Il flusso di progetto è articolato secondo una struttura a livelli e include la creazione di modelli cosiddetti HDL (*Hardware Description Language*), espressi in linguaggio VHDL o Verilog, la verifica di tali modelli tramite simulazioni a eventi e il trasferimento del programma nella piattaforma fisica. I linguaggi di descrizione hardware nominati permettono di definire i componenti funzionali al progetto e istanziarli in una struttura gerarchica. Il sistema che ne risulta può essere analizzato a diversi stadi di astrazione, dalla descrizione di sistema fino alla rete di porte logiche.
Nel livello di astrazione più alto (detto comportamentale o, in inglese, *behavioural*) vengono definite le operazioni che ISE provvederà poi a implementare sull'hardware nella sequenza corretta, senza però informazioni temporali.
A livello strutturale viene rappresentata, invece, l'architettura interna del sistema, formata dai componenti di più basso livello e dalle relative connessioni; questo stadio presuppone la scelta degli elementi che realizzeranno i task richiesti dal progetto.
In posizione intermedia tra queste due si trova l'astrazione di tipo RTL (*Register Transfer Level*), nella quale viene descritta la composizione del dispositivo sviluppa-

to in termini di registri, logica combinatoria, bus e unità di controllo, assegnando le operazioni a un determinato ciclo di clock (le specifiche relative alle caratteristiche temporali verranno descritte in modo approfondito in seguito).
Va precisato, tuttavia, che indipendentemente dal livello di astrazione scelto come punto di partenza per la progettazione, il software ISE si occupa dell'implementazione dei restanti passi necessari per raggiungere la descrizione dell'architettura a livello più basso, ovvero appunto delle porte logiche, permettendo l'inserimento manuale di vincoli che regolino ogni step di tale processo. La descrizione di un modulo VHDL consta di due elementi principali:

- un'interfaccia (*entity*) che descrive solamente i terminali di I/O e la denominazione del circuito;
- una o più implementazioni (*architecture*) nelle quali è definito il comportamento e la struttura interna dell'oggetto.

In ciascuna architecture vi è una prima parte, denominata *dichiarativa*, nella quale vengono elencati i segnali interni, specificando la tipologia di dato, e una seconda parte, detta *assertiva*, dove sono descritte le operazioni logiche tra segnali e specificate le istruzioni di assegnazione. Per comprendere la logica di un linguaggio di programmazione destinato all'hardware è necessario aver presente, anche al fine di fare proprio il meccanismo di simulazione che verrà trattato più tardi, due concetti fondamentali: quelli di segnale e di processo.
Un segnale è una struttura dati in grado di rappresentare forme d'onda nel tempo. E' identificato dalla parola chiave *signal*, a esso è associato un tipo e (opzionale) un valore iniziale. L'operazione tramite la quale si fa uso di segnali è detta *assegnazione* e ha una sintassi di questo tipo:

$$P \Leftarrow AxorB \tag{4.1}$$

A sinistra del segno di assegnazione (<=) c'è il *target* (P), nella parte destra c'è il *signal driver*, in questo caso costituito dall'operazione booleana tra due segnali. L'assegnazione stabilisce un legame tra i suoi ingressi (i segnali contenuti nel driver) e l'uscita (il target) e viene attivata soltanto quando si verifica un *evento*, ovvero un cambiamento di valore di uno dei segnali del driver. Grazie a questa proprietà il linguaggio VHDL riesce a riprodurre la simultaneità dell'esecuzione di operazioni propria dei circuiti elettronici. La disposizione delle assegnazioni all'interno del codice, dunque, non influenza l'ordine temporale con cui queste verranno effettivamente eseguite. Un altro costrutto fondamentale nella programmazione VHDL è il *processo*. Un processo è un blocco di codice, che descrive il funzionamento di un modulo di logica sequenziale o combinatoria, la cui peculiarità è quella di attivarsi esclusivamente quando accede un evento su particolari segnali, i segnali appartenenti alla *sensitivity list*. Quando ciò si verifica, si attiva, appunto, l'esecuzione del processo e vengono presi in considerazione eventuali eventi sui driver interni del blocco secondo una particolare procedura che caratterizza il procedimento di simulazione a eventi.

ISE permette l'impiego di *segnali*, ovvero strutture dati in grado di rappresentare forme d'onda variabili nel tempo. Inoltre, ogni modulo VHDL può essere espresso mediante la vista *schematic*, che dà una rappresentazione simbolica dell'oggetto considerato evidenziando i segnali di input e output e le interconnessioni con gli altri moduli. La programmazione in questa modalità permette di semplificare l'analisi di strutture complesse scomponendo intuitivamente il sistema nei suoi sottoblocchi costituenti. Lo stesso inserimento dei moduli IP (*Intellectual Property*) - i blocchi circuitali parzialmente o totalmente preprogettati e messi a disposizione del programmatore - risulta immediato grazie all'utilizzo di tale opzione.

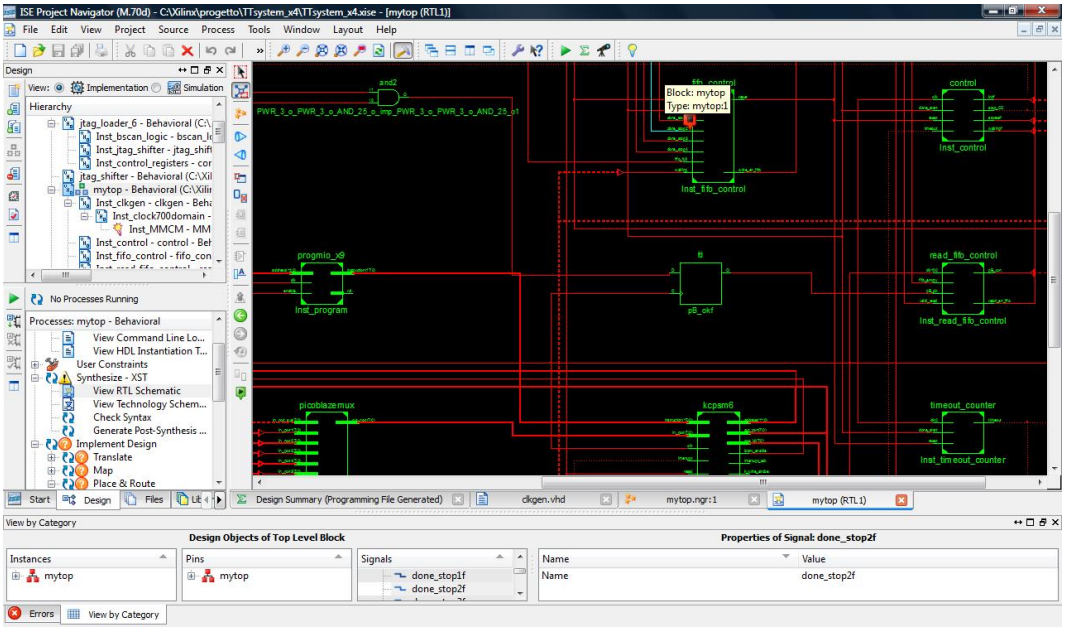


Figura 4.7: Interfaccia di programmazione ISE Project Navigator.

4.4.2 Debugging

ISE integra alla serie di strumenti forniti da Xilinx un tool di debug per la verifica del corretto funzionamento del progetto: il simulatore a eventi. Si tratta di un software in grado di applicare al sistema sotto analisi una successione temporale di ingressi e verificare il comportamento del circuito osservando i segnali che progressivamente vengono posti in uscita. E' possibile scegliere sia quali stimoli applicare al modulo VHDL compilato sia le caratteristiche temporali che questi presenteranno all'interno della simulazione del design. Le caratteristiche principali del simulatore sono:

- il modello interno del tempo;
- l'aggiornamento dei segnali;
- l'esecuzione dei processi e dei driver (cosiddetto *event processing*)

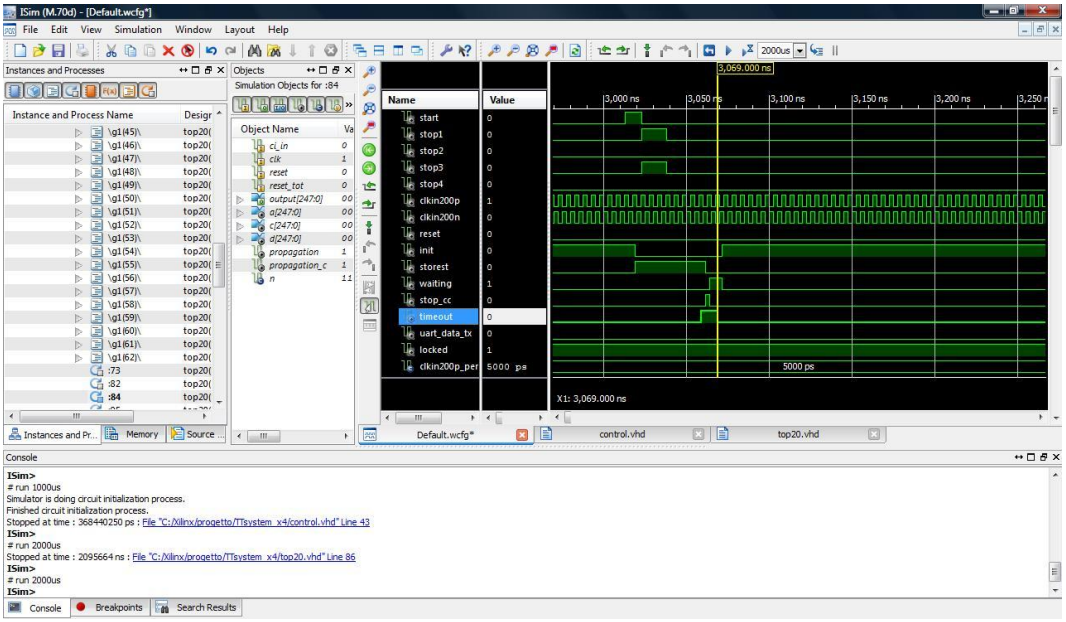


Figura 4.8: Ambiente ISim per la simulazione di progetti VHDL.

Come poc'anzi illustrato, un evento su un segnale appartenente alla sensitivity list di un processo ne attiva l'esecuzione. I segnali target che dipendono da tale segnale subiscono una *transazione* e il nuovo valore che devono assumere viene posto nella *coda degli eventi*. Una volta eseguiti tutti i driver sensibili all'evento, e inserite nella coda tutte le transazioni dei segnali target, il simulatore passa a considerare l'elemento successivo alla coda. Se l'aggiornamento del relativo segnale ne provoca il cambiamento di valore si ha un evento che, come più volte detto in precedenza, attiva i driver a esso sensibili facendo ripartire il processo descritto. L'esempio che segue fa riferimento a un semplice *full adder*, di cui si presenta il codice VHDL:

```
entity FULL_ADDER is
port(A, B, CI: in bit;
        CO, S: out bit);
end FULL_ADDER

architecture BHV of FULL_ADDER is
signal P, G: bit;
begin
P <= A xor B;
G <= A and B;
S <= P xor CI;
CO <= (P and CI) or G;
end BHV;
```

Impostando le commutazioni dei due segnali di ingresso è possibile osservare come il simulatore compie le operazioni di aggiornamento dei segnali ed esecuzione dei driver. La tabella seguente rappresenta la coda degli eventi che si aggiornerà nel corso della simulazione:

$5ns$	$10ns$				
$A \to 1$	$B \to 1$				

La commutazione di A a livello '1' è un evento dal momento che il suo valore di partenza era '0'. Se si considera il codice precedente, facendo particolare attenzione ai driver sensibili ad A:

```
P <= A xor B;
G <= A and B;
```

si può notare come le transazioni dei due segnali target vengano aggiunte alla coda e nel riferimento temporale del simulatore tali transazioni avvengano nello stesso istante della commutazione di A a '1'.

$5ns$	$5ns$	$5ns$	$10ns$		
$A \to 1$	$P \to 1$	$G \to 0$	$B \to 1$		

Giunti a questo punto, si passa a considerare il primo evento successivo nella coda degli eventi: il passaggio di P a '1'. Il nuovo valore viene confrontato con quello che precedentemente era stato assunto: poichè questi risultano differenti, P $\to$ 1 è davvero un evento e la forma d'onda del segnale P viene aggiornata al livello richiesto. Come è stato fatto per A, vengono ora analizzati tutti i driver sensibili a P e, dunque, aggiunte alla coda le transazioni dei target:

```
S <= P xor CI;
CO <= (P and CI) or G;
```

La nuova coda degli eventi risulta:

$5ns$	$5ns$	$5ns$	$5ns$	$5ns$	$10ns$
$A \to 1$	$P \to 1$	$G \to 0$	$S \to 1$	$CO \to 0$	$B \to 1$

La transazione successiva nella coda degli eventi è G $\to$ 0; visto che il valore precedente di G è '0' non si ha un evento, nessun driver sensibile a G viene attivato e la forma d'onda di G non necessita di essere aggiornata. Viene allora presa in considerazione la transazione S $\to$ 1, ripetendo il ciclo di operazioni finora eseguite.

Capitolo 5

Progetto di un Time-to-Digital Converter

Dopo aver descritto il fenomeno oggetto del nostro interesse (l'entanglement), analizzato alcune delle tecniche che permettono di osservarlo, almeno dal punto di vista temporale (il time interval measurement), e illustrato la piattaforma attraverso la quale concretizzare lo strumento di misura (l'FPGA), è giunto il momento di approfondire le caratteristiche del design sviluppato in questo lavoro.

Per comprendere appieno il funzionamento del TDC realizzato, si introdurranno dapprima le specifiche del sistema, si darà quindi una visione d'insieme della sua struttura interna e, infine, si andranno ad analizzarne i singoli elementi descrivendo il loro ruolo in riferimento al funzionamento globale.

5.1 Specifiche del sistema

Per rendere possibile la rivelazione di fotoni che nel caso ideale si generano e propagano contemporaneamente, mentre nella realtà, a causa dei limiti fisici degli strumenti a disposizione, giungono al dispositivo di misura con un ritardo il più possibile prossimo a 0, è necessario sfruttare tutte le capacità della board fornita al fine di ottenere la migliore risoluzione possibile.
Come richiesto dal progetto, si vuole che le coincidenze siano valutate da quattro possibili canali di trasmissione e, dunque, per ciascuno di essi dovrà essere eseguita la misura nel caso in cui si verifichi un arrivo. Il segnale di "inizio osservazione"verrà fatto coincidere con uno START manuale (settato, quindi, in base alle nostre esigenze) e innescherà il funzionamento del sistema per un certo intervallo di tempo, intervallo all'interno del quale il verificarsi di due segnali su canali distinti verrà interpretato come una coincidenza. E' stato supposto che all'interno di tale intervallo di osservazione, che in prima approssimazione considereremo pari a 40 ns, un solo arrivo (o nessuno) per canale sia possibile; questa ipotesi si può considerare plausibile tenendo presente che il segnale ottenuto in uscita dal rivelatore di singoli fotoni (lo SPAD, *Single Photon Avalanche Diode*) ha una durata dell'ordine della decina di ns

e lo strumento in questione presenta un dead time[1] successivo alla rivelazione che di fatto rende poco probabile l'identificazione di eventi multipli nello stesso lasso di tempo.
Possiamo riassumere le specifiche del TDC nei seguenti punti:

- massima risoluzione incrementale, determinata dall'hardware impiegato;
- intervallo di osservazione arbitrario (40 ns);
- segnale di START arbitrario;
- 4 canali di STOP;
- possibilità di arrivi (STOP) multipli, non più di uno per canale, all'interno dello stesso slot temporale;
- no arrivi (STOP) multipli nello stesso canale all'interno dello stesso slot temporale.

5.2 Architettura del TDC

Il TDC realizzato adotta come tecnica di misura uno schema di Nutt. Come visto in precedenza, questo metodo prevede per ogni canale l'analisi di tre sottointervalli, due dei quali misurati da interpolatori (detti *fine counter*) e il terzo convertito da un contatore grezzo (*coarse counter*) che opera alla frequenza di lavoro del sistema.
Le misure dei due fine counter si riferiscono alla durata temporale che intercorre tra i segnali di ingresso (START e STOP) e il primo fronte attivo di clock successivo, e, dunque, sono misure di breve range ed elevata risoluzione. Per eseguirle si fa uso delle carry chain ottenute dalla concatenazione della logica di riporto delle slice interne all'FPGA: l'uscita di un blocco viene posta in cascata collegandola all'ingresso della successiva e così via, creando una linea di lunghezza variabile in base all'intervallo complessivo da coprire.
Il coarse counter, invece, non è altro che un contatore sincrono con il clock di riferimento che tiene traccia del numero di cicli completi che intercorrono tra l'impulso di START e lo STOP associato al canale. In questo caso non è richiesta una risoluzione eccessivamente accurata - che infatti è pari al periodo del clock del sistema - bensì la possibilità di misurare intervalli di tempo arbitrariamente ampi.
Dal momento che le specifiche richiedono un tempo di osservazione delle coincidenze limitato, un contatore di timeout provvede a segnalare la fine dello slot di misura, imponendo a contatori e interpolatori di terminare il conteggio qualora non fosse giunto alcun impulso di STOP nel canale.

[1] Possiamo definire dead time il tempo necessario a un sistema di detection per essere in grado di individuare un nuovo evento dopo che ne ha appena rivelato uno. Tipicamente esso è costituito dal contributo intrinseco dell'elemento sensibile (nel nostro caso, per uno SPAD, il fotorivelatore) e dal tempo di conversione dello strumento.

Appare chiaro che la gestione delle varie fasi della misura ha una complessità tale da non poter essere assolta da una sequenza di istruzioni e task che si ripetono costantemente alla stessa maniera. Si pensi ad esempio al fatto che i vari conteggi devono avere inizio solamente quando viene azionato il comando arbitrario di START, mentre finchè ciò non si verifica il sistema deve rimanere in una sorta di "stato di attesa". Proprio questa situazione, ovvero il permanere del TDC in stati di funzionamento asincroni ai quali corrispondono azioni distinte, porta alla necessità di definire una *macchina a stati*.
Agli elementi centrali del TDC, adibiti alle operazioni di individuazione dei segnali di ingresso e misura dell'intervallo temporale, va aggiunta, inoltre, un'unità di memorizzazione temporanea e successiva trasmissione dei dati per permettere la visualizzazione dei risultati all'esterno. Per il primo task si è scelto di utilizzare una memoria FIFO (*First In First Out*) sfruttando i core IP (*Intellectual Property*) precompilati forniti dalla Xilinx, mentre per il trasferimento dei dati si è optato per la soluzione operativamente più semplice: la comunicazione seriale implementando il protocollo UART (*Universal Asynchronous Receiver/Transmitter*). Per coordinare questi processi, effettuando tra l'altro le azioni di controllo sullo stato della memoria e della trasmissione, e fornire il segnale di RESET alla macchina a stati una volta che la comunicazione è terminata, è parsa evidente la necessità di sfruttare un'altra risorsa messa a disposizione dalla piattaforma ML605: il microcontrollore *picoBlaze*. Il design sviluppato mette insieme questi moduli configurando per il TDC l'architettura mostrata in figura 5.1.

Va precisato che l'implementazione dei vari componenti del sistema è frutto di un'attenta analisi delle funzionalità e delle problematiche associate a ciascuno di essi che si è snodata nel lavoro presente e nei progetti precedenti dei quali questo è la naturale prosecuzione. Si andrà allora, ove possibile e ritenuto opportuno, a riprendere alcune delle valutazioni già consolidate al fine di giustificare le scelte progettuali fatte.

5.2.1 Coarse counter

L'idea alla base dello sviluppo del modulo coarse counter è piuttosto semplice e consiste nel contare il numero intero di cicli di clock del sistema contenuto all'interno dell'intervallo da misurare. Dal punto di vista della programmazione è stato dunque sufficiente un controllo del livello logico dei segnali di START e di STOP e l'incremento di una variabile di conteggio fino a che è verificata la condizione START = '1' and STOP = '0' oppure è terminata la finestra temporale di osservazione.
Ciò di cui si è dovuto tenere conto in un design apparentemente scontato è stata la stabilità dei registri che devono memorizzare i dati di questa misura grezza. Può infatti accadere che il fronte di salita di uno dei segnali di ingresso si verifichi a stretta corrispondenza con il fronte di clock del sistema, violando i tempi di setup o di hold e inducendo il campionamento di un valore non corretto.
Il *tempo di setup* è definito come il minimo intervallo temporale durante il quale il dato in ingresso al fllip-flop deve permanere in uno stato stabile prima che il clock

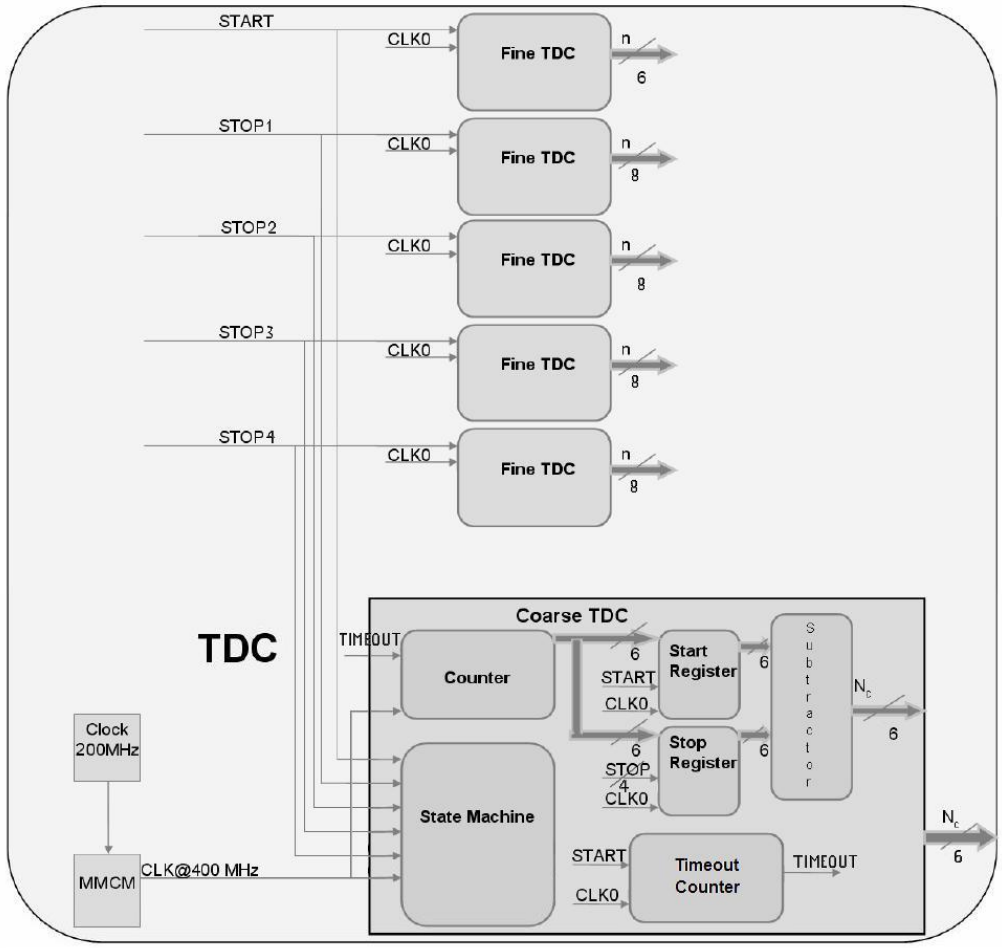


Figura 5.1: Schema semplificato dell'architettura del TDC.

compia una transizione valida.
Il *tempo di hold* rappresenta, invece, il minimo periodo di tempo per cui il dato deve mantenersi stabile dopo la transizione del clock affinchè il suo campionamento avvenga in maniera corretta.
Si tratta di due parametri che risultano molto importanti in fase di progettazione; il mancato rispetto di questi vincoli temporali è segnalato in fase di routing da ISE. Qualora un ingresso cambiasse valore all'interno di questi intervalli, infatti, esisterebbe la possibilità di indurre l'elemento di memoria in una posizione cosiddetta *metastabile*, ovvero non in equilibrio tra i livelli logici '0' e '1'. Ovviamente questa è una condizione che si vuole evitare perchè introdurrebbe un grado di imprevedibilità nel comportamento del sistema che non può essere accettato. I tempi di setup e hold sono consultabili nel datasheet dell'FPGA nella sezione "*DC and Switching Characteristics*"(se ne può vedere l'estratto in figura 5.2).

Setup and Hold Times of CLB Flip-Flops Before/After Clock CLK						
T_{DICK}/T_{CKDI}	A – D input to CLK on A – D Flip Flops	0.30/ 0.17	0.36/ 0.18	0.43/ 0.20	0.44/ 0.25	ns, Min
$T_{CECK\_CLB}/$ T_{CKCE_CLB}	CE input to CLK on A – D Flip Flops	0.20/ 0.00	0.25/ 0.00	0.32/ 0.00	0.32/ 0.01	ns, Min
T_{SRCK}/T_{CKSR}	SR input to CLK on A – D Flip Flops	0.39/ –0.07	0.44/ –0.07	0.52/ –0.07	0.58/ –0.08	ns, Min
T_{CINCK}/T_{CKCIN}	CIN input to CLK on A – D Flip Flops	0.16/ 0.12	0.19/ 0.14	0.24/ 0.16	0.23/ 0.22	ns, Min

Figura 5.2: Tempi di setup e hold dell'FPGA Virtex-6

La prima soluzione che era stata prospettata, tenendo conto di queste considerazioni, prevedeva l'utilizzo, per un singolo canale, di quattro coarse counter funzionanti alla stessa frequenza di clock ma sfasati tra loro di 90°. I quattro riferimenti *clock0*, *clock90*, *clock180* e *clock270* permettevano quindi di campionare l'intervallo temporale in quattro intervalli equamente divisi all'interno del range di incertezza T_0, garantendo così, sulla carta, la corretta misura del fronte del segnale di ingresso. Questa scelta, in realtà, non risolveva del tutto i problemi del conteggio grezzo, introducendone, invece, altri. Per prima cosa, sorgeva la questione di come assicurare un uguale percorso ai segnali di START e STOP che dovevano giungere a ben quattro oggetti distinti; inoltre, diventava necessario introdurre una logica addizionale per la decisione del contatore che aveva eseguito il campionamento corretto.
Per aggirare questa situazione si è preferito tornare all'impiego di un singolo clock di riferimento, alla frequenza di 400 MHz, e provvedere ad aggiungere nel design cinque segnali - *done_start*, *done_stop1*, *done_stop2*, *done_stop3*, *done_stop4* - che rappresentano il primo bit di ciascuna carry chain associata allo START e ai quattro segnali di STOP. Utilizzando questi nuovi segnali è possibile costruire contatori che danno inizio al conteggio all'attivazione del segnale done_start e ne pongono il termine quando passa ad HIGH il segnale di done_stop di quel canale. Questa architettura non esegue la misura in real time rispetto all'arrivo dei segnali di ingresso, bensì con un ritardo che nel peggiore dei casi è pari a due cicli di clock ($T_0 = 2.5$ ns). Tale latenza, tuttavia, non crea al sistema alcun tipo di problema in quanto i conteggi terminano prima che i codificatori termometrici inizino la conversione del fine counter associato a ogni segnale di STOP; l'elaborazione delle varie misure è, inoltre, gestita attraverso la macchina a stati finiti che può scandire le varie fasi del processo in maniera opportuna.

Viene ora riportato il codice VHDL che implementa il processo di conteggio e successiva assegnazione dei risultati della misura dei quattro coarse counter:

```
process(clk0,reset,reset_tot, done_start, done_stop1)
begin
if(reset = '1' or reset_tot = '1') then
count0_1 <= (others => '0');
elsif (rising_edge(clk0)) and (done_start='1')and(done_stop1='0')
```

```
then
count0_1  <= count0_1 + 1;
end if;
end process;

process(clk0,reset,reset_tot, done_start, done_stop2)
begin
if(reset = '1' or reset_tot = '1') then
count0_2 <= (others => '0');
elsif (rising_edge(clk0)) and (done_start='1')and(done_stop2='0')
then
count0_2  <= count0_2 + 1;
end if;
end process;

process(clk0,reset,reset_tot, done_start, done_stop3)
begin
if(reset = '1' or reset_tot = '1') then
count0_3 <= (others => '0');
elsif (rising_edge(clk0)) and (done_start='1')and(done_stop3='0')
then
count0_3  <= count0_3 + 1;
end if;
end process;

process(clk0,reset,reset_tot, done_start, done_stop4)
begin
if(reset = '1' or reset_tot = '1') then
count0_4 <= (others => '0');
elsif (rising_edge(clk0)) and (done_start='1')and(done_stop4='0')
then
count0_4  <= count0_4 + 1;
end if;
end process;

process(stop_CC)
begin
if(rising_edge(stop_CC)) then
output0_1 <= count0_1;
output0_2 <= count0_2;
output0_3 <= count0_3;
output0_4 <= count0_4;
end if;
```

```
end process;
```

Si può notare che vengono definiti cinque processi: quattro di questi sono sensibili al segnale di clock, ai reset di sistema, al segnale done_start e al done_stop relativo al canale, ed eseguono il conteggio vero e proprio dei cicli di clock che intercorrono tra i due ingressi; l'ultimo processo provvede all'assegnazione dei risultati della misura alle uscite del blocco al momento opportuno, cioè in corrispondenza dell'attivazione del segnale di stop_CC fornito dalla macchina a stati finiti dopo che il contatore di timeout ha segnalato la chiusura della finestra di osservazione.

5.2.2 Fine counter

Il modulo più sensibile e nel quale si sfruttano in modo spinto le performance dell'hardware messo a disposizione dalla board ML605 è sicuramente quello adibito al conteggio fine. Il compito degli interpolatori o *fine counter* è quello di migliorare la risoluzione del TDC misurando il tratto iniziale e finale dell'intervallo di interesse, lasciando la misura grossolana, che richiede un ampio range temporale, al coarse counter.
La struttura di questi contatori fini sfrutta le catene di riporto dell'FPGA per generare una linea di ritardo di cui si conosce, con una certa approssimazione, il tempo di percorrenza. Le caratteristiche che rendono adatta la CARRY4 per la realizzazione del fine counter possono essere fissate nei seguenti punti:

- disposizione lineare e sequenziale dei blocchi logici al suo interno;
- possibilità di disporre facilmente più blocchi in cascata;
- capacità di memorizzare dati grazie alla presenza di un flip-flop all'uscita di ogni multiplexer;
- tempo di attraversamento della carry da parte di un segnale breve e noto a priori (con un certa incertezza).

La caratteristica di fondamentale importanza di questa struttura risiede nel basso ritardo che introduce quando un segnale la attraversa: è infatti da questo ritardo che si ricava la minima variazione temporale che può essere rilevata dal sistema, parametro che concorre sicuramente a definire la bontà del TDC.
La necessità di dover osservare quattro canali dai quali può giungere il fotone (ovvero, nel nostro modo di studiare il problema, il segnale di STOP) comporta l'utilizzo di altrettanti interpolatori, a cui se ne aggiunge un quinto da associare alla misura fine dello START. La propagazione di ciascun segnale di ingresso nel relativo fine counter avviene entrando dall'ingresso CYINIT della prima CARRY4 e attraversa i vari blocchi della catena tramite le uscite COUT e gli ingressi CIN collegati tra loro. Il segnale viaggia nella linea fino all'arrivo del primo fronte attivo di clock: in quel momento i flip-flop D presenti in ognuna delle quattro uscite delle varie CARRY4 provvedono a campionare il segnale corrente, che sarà '1' in corrispondenza

dei blocchi nei quali il segnale di ingresso ha fatto in tempo a propagarsi e '0' nelle restanti uscite.
Il codice VHDL sviluppato per istanziare la catena di dimensione N è qui di seguito presentato:

```
   G1: for i in 1 to N generate
G1: if i = 1  generate
CARRY4_inst1 : CARRY4
port map (
CO => A(i*4-1 downto i*4-4),
O => OPEN,
CI => '0',
CYINIT => CI_in,
DI => "0000",
S => "1111"
);
end generate;
G3: if i = N  generate
CARRY4_instN : CARRY4
port map (
CO => A(i*4-1 downto i*4-4),
O => OPEN,
CI => A(i*4-5),
CYINIT => '0',
DI => "0000",
S => "1111"
);
end generate;
G4: if i > 1 and i < N generate
CARRY4_inst2 : CARRY4
port map (
CO => A(i*4-1 downto i*4-4),
O => OPEN,
CI => A(i*4-5),
CYINIT => '0',
DI => "0000",
S => "1111"
);
end generate;
   end generate;
```

Analizzando il datasheet del chip Virtex-6 si appura che il tempo nominale necessario per l'attraversamento di una CARRY4 è pari a 80 ps e in generale, per la board in uso, compreso tra 60 e 90 ps a seconda della velocità delle performance dell'FPGA (vedi figura 5.3).

Symbol	Description	Speed Grade				Units
		-3	-2	-1	-1L	
Combinatorial Delays						
T_{ILO}	An – Dn LUT address to A	0.06	0.07	0.07	0.09	ns, Max
	An – Dn LUT address to AMUX/CMUX	0.18	0.20	0.22	0.25	ns, Max
	An – Dn LUT address to BMUX_A	0.28	0.31	0.36	0.40	ns, Max
T_{ITO}	An – Dn inputs to A – D Q outputs	0.59	0.67	0.79	0.85	ns, Max
T_{AXA}	AX inputs to AMUX output	0.31	0.35	0.42	0.44	ns, Max
T_{AXB}	AX inputs to BMUX output	0.35	0.39	0.47	0.50	ns, Max
T_{AXC}	AX inputs to CMUX output	0.39	0.44	0.52	0.56	ns, Max
T_{AXD}	AX inputs to DMUX output	0.42	0.47	0.55	0.60	ns, Max
T_{BXB}	BX inputs to BMUX output	0.30	0.34	0.39	0.44	ns, Max
T_{BXD}	BX inputs to DMUX output	0.38	0.43	0.50	0.55	ns, Max
T_{CXB}	CX inputs to CMUX output	0.26	0.29	0.34	0.37	ns, Max
T_{CXD}	CX inputs to DMUX output	0.30	0.34	0.40	0.44	ns, Max
T_{DXD}	DX inputs to DMUX output	0.30	0.33	0.38	0.43	ns, Max
T_{CPCYA}	An input to COUT output	0.32	0.36	0.41	0.47	ns, Max
T_{CPCYB}	Bn input to COUT output	0.32	0.36	0.41	0.47	ns, Max
T_{CPCYC}	Cn input to COUT output	0.27	0.30	0.34	0.40	ns, Max
T_{CPCYD}	Dn input to COUT output	0.25	0.28	0.32	0.37	ns, Max
T_{AXCY}	AX input to COUT output	0.25	0.28	0.33	0.36	ns, Max
T_{BXCY}	BX input to COUT output	0.22	0.24	0.28	0.31	ns, Max
T_{CXCY}	CX input to COUT output	0.15	0.17	0.20	0.22	ns, Max
T_{DXCY}	DX input to COUT output	0.14	0.16	0.19	0.21	ns, Max
T_{BYP}	CIN input to COUT output	0.06	0.07	0.08	0.09	ns, Max
T_{CINA}	CIN input to AMUX output	0.21	0.24	0.28	0.30	ns, Max
T_{CINB}	CIN input to BMUX output	0.23	0.25	0.29	0.31	ns, Max
T_{CINC}	CIN input to CMUX output	0.23	0.26	0.30	0.33	ns, Max
T_{CIND}	CIN input to DMUX output	0.25	0.29	0.33	0.36	ns, Max
Sequential Delays						
T_{CKO}	Clock to AQ – DQ outputs	0.29	0.33	0.39	0.44	ns, Max
T_{SHCKO}	Clock to AMUX – DMUX outputs	0.36	0.40	0.47	0.53	ns, Max
Setup and Hold Times of CLB Flip-Flops Before/After Clock CLK						
T_{DICK}/T_{CKDI}	A – D input to CLK on A – D Flip Flops	0.30/ 0.17	0.36/ 0.18	0.43/ 0.20	0.44/ 0.25	ns, Min
$T_{CECK\_CLB}/$ T_{CKCE_CLB}	CE input to CLK on A – D Flip Flops	0.20/ 0.00	0.25/ 0.00	0.32/ 0.00	0.32/ 0.01	ns, Min
T_{SRCK}/T_{CKSR}	SR input to CLK on A – D Flip Flops	0.39/ −0.07	0.44/ −0.07	0.52/ −0.07	0.58/ −0.08	ns, Min
T_{CINCK}/T_{CKCIN}	CIN input to CLK on A – D Flip Flops	0.16/ 0.12	0.19/ 0.14	0.24/ 0.16	0.23/ 0.22	ns, Min

Figura 5.3: Switching characteristics dei CLB della Virtex-6

Ne risulta perciò che la risoluzione incrementale (LSB) è di circa 20 ps. Va sottolineato che il valore fornito dalla ditta produttrice nella documentazione si riferisce al *worst case*, ovvero rappresenta il massimo tempo di attraversamento della carry nella condizione peggiore. Nel caso reale tale valore potrebbe essere minore, e di questo bisogna tenere conto quando si dimensiona la linea di ritardo in fase di progettazione. Considerando che il segnale di clock al quale si esegue il campionamento delle uscite

della carry chain lavora alla frequenza di 400 MHz, è necessario coprire con la cascata di CARRY4 una distanza temporale di $T_0 = 2.5$ ns. Il numero minimo di blocchi che, collegati tra loro a formare la linea di riporto, potrebbero assolvere questo compito si può facilmente calcolare come $\frac{2.5*10^{-9}}{0.8*10^{-9}} = 31.25 \rightarrow 32$ CARRY4. Tenendo conto dello skew del clock e del fatto che i tempi di propagazione nella carry chain potrebbero essere più bassi, si è preferito mantenere un certo margine creando catene di 42 blocchi, per un totale di 168 bit di misura fine, per ogni canale. In linea teorica, i fine counter così progettati dovrebbero poter supportare CARRY4 con tempo di attraversamento pari a 60 ns (che corrisponde alle prestazioni dell'FPGA Virtex-6 a velocità massima -3). La scelta di sovradimensionare le carry chain deriva dall'esperienza di progettazione di questo sistema; nelle prime versioni del TDC, infatti, la misura fine tendeva spesso a saturare proprio a causa di un insufficiente numero di blocchi che non era in grado di garantire la copertura dell'intero periodo di clock.
Vale la pena parlare ora di una questione che si è presentata nello sviluppo delle prime versioni del design e che ha a che fare con un componente fondamentale nel funzionamento dell'intero sistema: il clock. La frequenza di campionamento dei fine counter, coincidente con la frequenza di lavoro delle macchina a stati che gestisce le operazioni sul sistema, è rimasta fedele alla scelta progettuale di partenza: 400 MHz. Si tratta di un valore che il TDC nella sua fase iniziale, sviluppato per board antecedenti alla Virtex-6, era in grado di riprodurre con una certa facilità tramite l'uso dei cosiddetti DCM (*Digital Clock Manager*). I DCM (vedi figura 5.4) sono moduli IP precompilati che permettono di generare clock alla frequenza desiderata a partire da un clock di frequenza nota fornito dall'esterno o estraibile dalla circuiteria interna della board.

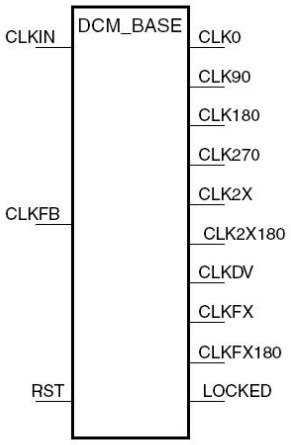


Figura 5.4: Modulo DCM_BASE per la generazione di segnali di clock

Le unità funzionali presenti in un DCM sono tre:

- *DLL* (*Delay Locked Loop*): il suo compito è compensare il ritardo (*skew*) tra il clock in ingresso e ogni circuito da esso pilotato. Per ottenere l'effetto compensativo viene introdotto un ritardo alla sorgente tale da annullare la differenza di fase tra il clock e ogni sua versione presa in un punto qualsiasi del circuito e riportata in ingresso tramite feedback;

- *Sintetizzatore digitale di frequenza*: questo modulo permette di generare (anche dinamicamente, usando il modulo DCM_ADV) la frequenza desiderata come frazione $\frac{M}{D}$ di quella del clock in ingresso, con M e D interi all'interno di un range di valori ammissibili;

- *Unità di sfasamento*: permette di sfasare di una frazione di periodo i clock in uscita dal DCM rispetto al clock di ingresso; è possibile, inoltre, variare in maniera dinamica il valore di sfasamento applicato alle uscite (in questo caso è necessario implementare una macro particolare: DCM_PS).

L'implementazione iniziale del generatore di clock si basava sull'impiego di due DCM posti in cascata collegando l'uscita del primo, alla frequenza di 200 MHz, in ingresso al secondo e sfruttando l'output già predisposto che raddoppia il segnale introdotto nel DCM. Un problema principalmente affliggeva il modulo così composto: la stabilità. Infatti, il segnale *locked*, che rende conto proprio del fatto che il segnale abbia o meno raggiunto il suo andamento di regime, non passava mai allo stato logico HIGH. Dopo aver analizzato il problema anche dal punto di vista simulativo, ci si era resi conto che singolarmente i due blocchi funzionavano, ma ponendoli in cascata necessitavano di essere attivati in tempi diversi in modo da fornire un segnale finale che tenesse conto del tempo di transitorio che entrambi i blocchi necessitavano di fare trascorrere prima di trasferire i dati in uscita. Questa idea era stata tradotta allora assegnando al segnale *resetclk* del secondo DCM la versione invertita dell'uscita locked del primo: in tal modo il secondo generatore avrebbe iniziato a processare il clock solo dopo che questo fosse risultato stabile.
La versione attuale del progetto si avvale nei nuovi core messi a disposizione dalla Xilinx, tra i quali la macro $MMCM\_BASE$ che sostituisce di fatto i DCM per la generazione dei segnali di clock. Oltre alle proprietà già descritte nei DCM, queste strutture consentono la sintetizzazione di segnali in un range più ampio (grazie all'aumento delle performance delle schede che le supportano) e permettono la scelta di parametri per la riduzione del jitter. In figura 5.5 è mostrata la definizione del blocco MMCM che garantisce le funzioni base per la generazione dei riferimenti temporali nelle board elettroniche di ultima generazione come la Virtex-6.

Un altro aspetto importante nella gestione della misura fine è il mantenimento dei dati finchè questi vengono trasmessi all'esterno. Affinchè il codice termometrico converta le misure corrette è necessario che i suoi ingressi si mantengano stabili; per adempiere a questo compito inizialmente si era definito un singolo segnale, chiamato *propagation*, che valeva '1' quando il primo bit del codice termometrico valeva '0' e '0' in caso contrario, disabilitando la scrittura dei registri della carry chain fino all'arrivo del segnale di reset. Questo modo di operare, tuttavia, ha iniziato a entrare

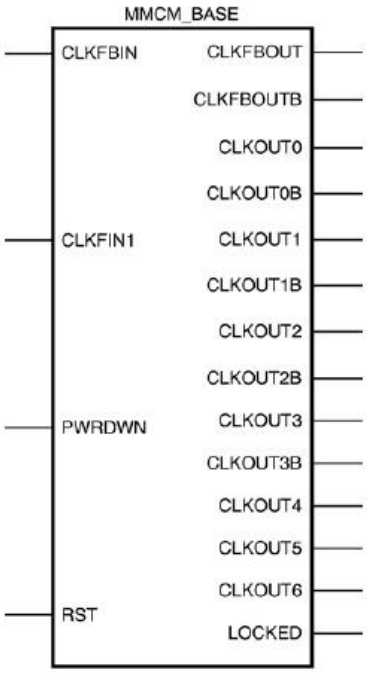


Figura 5.5: Modulo MMCM_BASE per la generazione di segnali di clock

in crisi quando la catena ha raggiunto, per necessità, dimensioni più elevate. Ciò che si verificava era la violazione dei vincoli temporali legati al clock di sistema. Continuando ad aumentare la lunghezza della catena di riporto senza modificare l'architettura di funzionamento avrebbe portato al mancato blocco dei flip-flop "più alti"da parte del segnale propagation, e quindi alla possibilità di sovrascrittura di un dato legato al campionamento con un fronte di clock successivo rispetto a quello dei primi flip-flop.
La soluzione proposta per aggirare questo problema consiste nella "preparazione"del segnale propagation al momento del campionamento con i flip-flop interni alle slice, per poi inviarlo ai registri posti in seguito che provvederanno a mantenere il dato. Con questa struttura si va quindi a eliminare il tempo morto legato allo switch del segnale tramite un inverter.
Il codice seguente descrive l'inizializzazione dei flip-flop presenti nelle slice a ogni uscita CO e la nuova architettura introdotta per modificare il segnale propagation:

```
process(clk,A,reset)
begin
if(reset = '1') then
C <= (others => '0');
elsif(rising_edge(clk))then
C <= A;
```

```
end if;
end process;

propagation <= '1' when D(0) = '0' else '0';

process(clk,C(0),reset,reset_tot,propagation)
begin
if(reset='1' or reset_tot='1') then
propagation_C <= '1';
elsif(rising_edge(clk))then
if(propagation = '1')then
propagation_C <= not(C(0));
end if;
end if;
end process;

process(clk,reset,reset_tot,propagation_C)
begin
if(reset = '1' or reset_tot = '1')then
D <= (others => '0');
elsif(rising_edge(clk)) then
if(propagation_C = '1')then
D <= C;
end if;
end if;
end process;

output <= D;
```

5.2.3 Macchina a stati finiti

I segnali di ingresso al TDC (START, STOP1, STOP2, STOP3, STOP4) sono, in generale, asincroni rispetto al clock di sistema dal momento che possono verificarsi in istanti temporali casuali non periodici. Questa situazione pone in essere alcune problematiche relative alla gestione di operazioni che non sono semplicemente scandite dalla stessa frequenza, bensì dipendono dagli eventi in gioco. Per risolverle si è puntato a realizzare una macchina a stati finiti (abbreviato FSM, dall'inglese *Finite State Machine*) che consente il completamento delle varie fasi del processo, dall'acquisizione degli ingressi all'esecuzione delle misure, dal trasferimento dei dati al picoblaze alla trasmissione seriale tramite prontocollo RS-232.

La macchina a stati pone il sistema in quattro possibili status operandi: INIT, START_ARRIVED, TAKE_MEASURE, WAITING. INIT rappresenta lo stato di inizializzazione in cui il sistema è in attesa dell'arrivo del segnale di inizio conteggio;

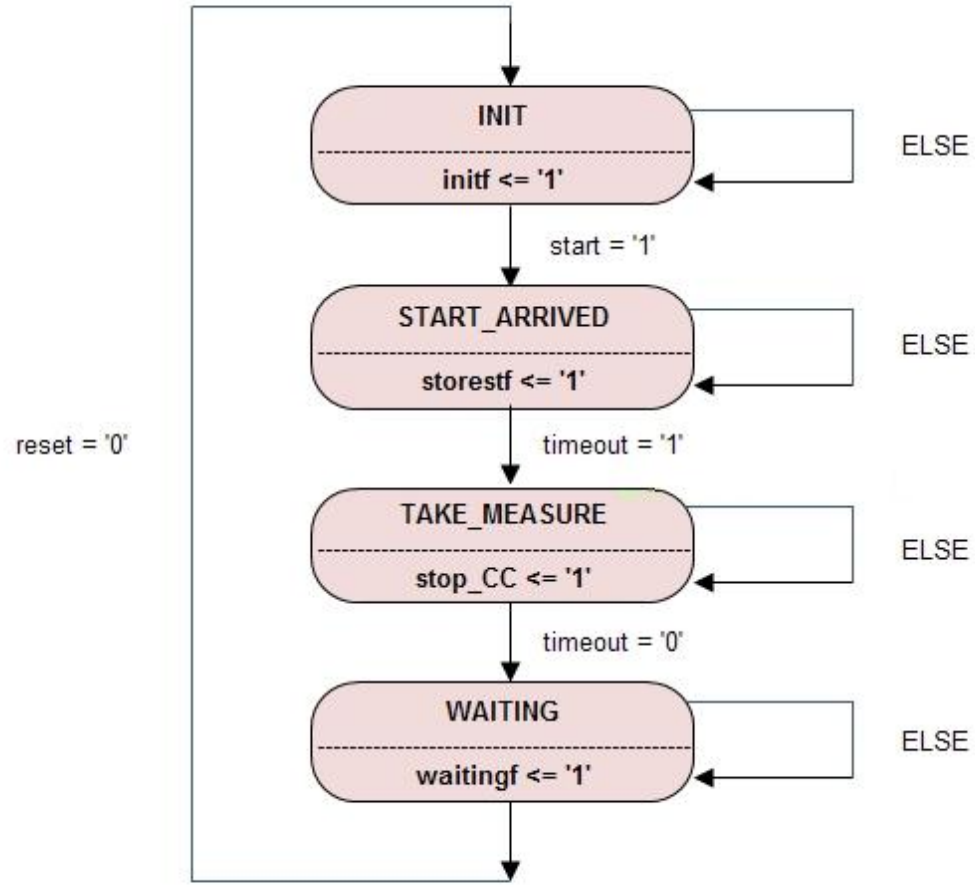


Figura 5.6: Macchina a stati finiti del TDC

il TDC, dunque, terminata la trasmissione della misura precedente, viene resettato e permane in questo stato finchè si verifica un evento di START. Quando ciò accade, ovvero START $\rightarrow$ HIGH, la FSM passa allo stato START_ARRIVED dove vengono avviati i conteggi grezzi per i vari canali: il calcolo della misura fine dello START si esaurisce al primo fronte di clock successivo mentre per tutti i restanti dati si attendono gli eventuali STOP associati. All'arrivo del segnale di *timeout*, indipendentemente da quali e quanti segnali di STOP sono arrivati, il sistema passa allo stato successivo: TAKE_MEASURE. Questo stato è necessario per permettere ai counter coarse e ai fine counter di trasferire i dati delle misure a una memoria FIFO, dalla quale verranno poi prelevati durante la fase di trasmissione. Un solo ciclo di clock è richiesto per queste operazioni, per cui dopo un periodo T_0 il sistema passa automaticamente allo stato WAITING, durante il quale il picoBlaze gestisce

la trasmissione dei vari conteggi e invia il reset globale una volta che questa è terminata.
Per tenere traccia della situazione del processo in corso, vengono definiti quattro segnali di controllo che identificano i rispettivi stati: initf, storestf, stop_CC e waitingf. Quando il sistema si trova nello stato associato il relativo flag passa al livello logico alto, permettendo così di monitorare facilmente le varie fasi del TDC.
Oltre al flusso di operazioni principale, descritto dalla precedente FSM, è stato necessario definire due moduli VHDL per la gestione delle fasi di scrittura e lettura della memoria FIFO, sulla quale vengono convogliati i risultati delle misure e dalla quale questi vengono prelevati per essere trasmessi tramite comunicazione seriale al PC. Ciascuno di essi descrive un'ulteriore macchina a stati che andiamo ora a illustrare, partendo da quella che si occupa della scrittura dei dati (vedi figura 5.7).
Lo stato iniziale della macchina a stati finiti dedicato alla scrittura sulla FIFO è FIFO_INIT; la FSM permane in questo stato fino a che non si verifica la condizione waitingf = HIGH, ovvero l'elaborazione dei dati provenienti dai fine counter e dai coarse counter è conclusa. Al verificarsi di questo evento, la macchina a stati passa a FIFO_TEST, nel quale sostanzialmente viene fatto il controllo della situazione della FIFO interrogando il flag FIFO_FULL: se la FIFO non è piena si può subito procedere alla scrittura del dato a 64 bit (la lunghezza complessiva del dato memorizzato si ottiene tenendo presente che vengono eseguite 5 misure fini a 8 bit e 4 misure grezze a 6 bit), in caso contrario si rimane in attesa del suo svuotamento. Terminata il controllo sulla FIFO, la macchina a stati passa WR_ON_FIFO_1. Questo stato si è reso necessario per abilitare la scrittura vera e propria dei dati nel buffer. Al successivo fronte attivo di clock di sistema si passa, quindi, allo stato WR_OFF_FIFO durante il quale viene disabilitata la scrittura e inviato un segnale di reset al sistema. Il ciclo di scrittura è stato completato, e per riportare la FSM alle condizioni iniziali si attende che i segnali di done_start e done_stop tornino a livello logico basso, così da assicurarsi l'effettivo ripristino dello stato di partenza del TDC.
Passiamo ora a descrivere il comportamento del modulo di lettura dei dati in uscita dalla FIFO in vista della loro trasmissione attraverso il protocollo UART al PC.
Come è possibile vedere in figura 5.8, il primo stato di questa nuova FSM è FIFO_READ_TEST. Anche in questo caso, analogamente a quanto visto per la scrittura dei dati, bisogna eseguire un controllo preliminare sulla FIFO, ora però per accertarsi che non sia vuota. Se sono presenti dati - e questo si può appurare osservando il segnale di controllo FIFO_EMPTY fornito dall'IP che definisce la memoria - la FSM fa avanzare il sistema allo stato READ_FIFO_INIT. Qui attraverso la commutazione a '1' dei segnali pb_run e read_en_fifo si procede rispettivamente ad accendere il picoBlaze e abilitare la lettura del dato in uscita. Il segnale valid_read, definito dall'IP che genera la FIFO, fornisce l'indicazione sulla tempistica relativa alla presenza di un dato valido pronto a essere prelevato. Quando tale segnale passa ad HIGH si compie un nuovo step in avanti portando il sistema allo stato RE_OFF_FIFO; il segnale di lettura viene disabilitato mentre il segnale di atti-

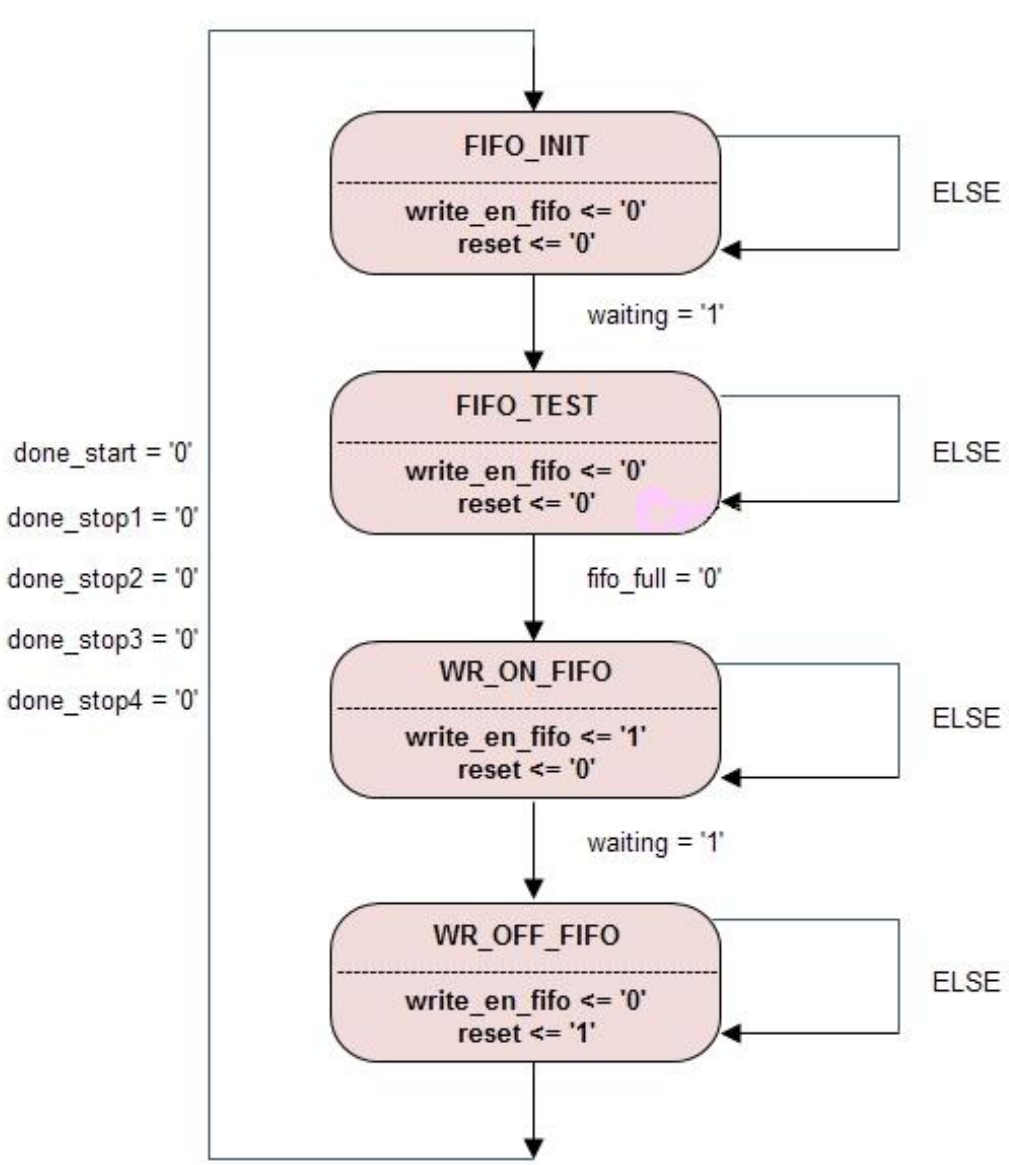


Figura 5.7: Macchina a stati finiti per la scrittura della FIFO

vazione del picoBlaze rimane ancora HIGH. Una volta scaricati tutti i dati, il picoBlaze provede a impostare i registri in modo da segnalare l'avvenuta trasmissione, il segnale pB_OK viene settato a '1' e la macchina stati torna alla condizione iniziale FIFO_READ_TEST.

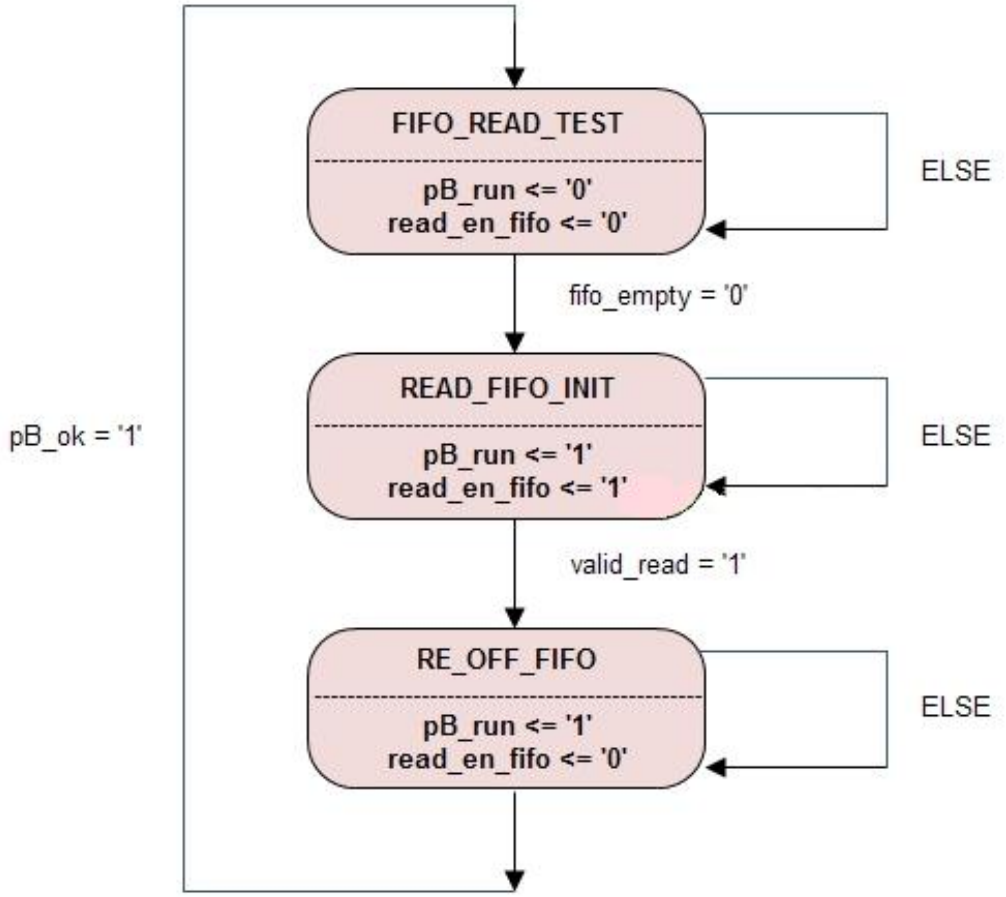


Figura 5.8: Macchina a stati finiti per la lettura della FIFO

5.2.4 Gestione dell'I/O

Dopo aver descritto come il sistema opera in relazione all'arrivo dei segnali di ingresso e come coordina la fase preliminare della trasmissione in base allo stato della memoria, pare opportuno introdurre la discussione su come i dati vengono effettivamente trasferiti dalla board al PC.
Come detto, per curare la fase di elaborazione e visualizzazione dei dati si è scelto di sfruttare la trasmissione seriale in modo da poter analizzare e salvare le misure sul calcolatore, tramite appropriati tool di gestione delle periferiche di comunicazione. Fisicamente la board ML605 mette a disposizione un bridge UART-USB che permette di trasferire i dati in modalità seriale a un dispositivo esterno attraverso un cavo mini USB-USB.

Il chip che fa da ponte tra i due protocolli di trasmissione è il CP2103 realizzato dalla Silicon Labs. Si tratta di un componente integrato che permette un facile updating dal design RS-232/RS-485[2] all'USB impiegando una porzione minima di componenti e di spazio nella scheda su cui risiede. All'interno di un package compatto e senza bisogno di ulteriori elementi esterni, nel chip sono presenti:

- USB 2.0 full-speed function controller;
- USB transceiver;
- oscillatore locale;
- memoria EEPROM;
- bus dati seriale asincrono UART.

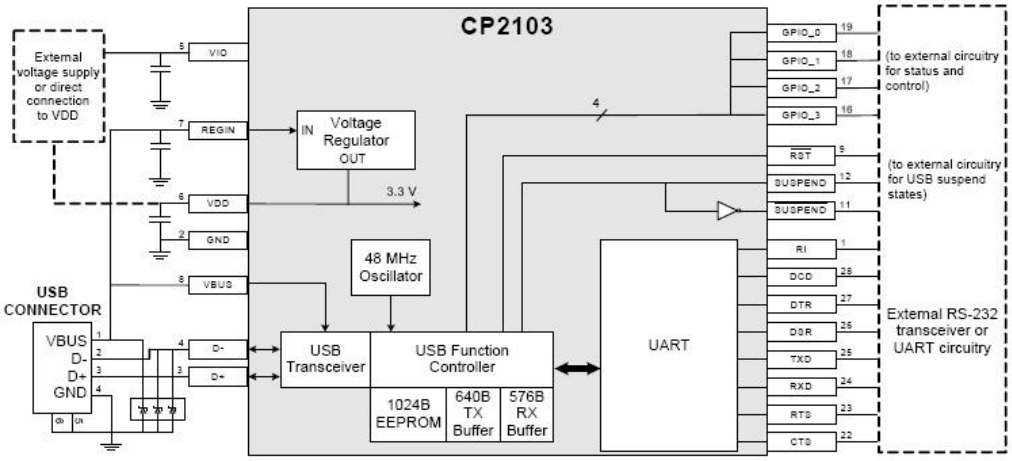


Figura 5.9: Chip CP2103 UART to USB Bridge

L'interfaccia UART del chip CP2103 implementa tutti i segnali dell'RS-232/RS-485, inclusi i segnali di controllo e di handshaking, per cui il firmware esistente non richiede di essere modificato. Sono inoltre previsti fino a 4 segnali GPIO definiti dall'utente per monitorare le informazioni di stato e di controllo del dispositivo. L'UART può essere programmata in modo da implementare un'ampia varietà di formati di trasmissione con diversi baud rate, come mostrato in figura 5.10.

All'interno del design del TDC l'UART è stata implementata attraverso il modulo *uart_tx.vhd* (di cui è illustrata la entity VHDL) che a sua volta definisce due sottoblocchi: *kcuart.vhd* (dalle iniziali del programmatore che l'ha sviluppato,

[2]EIA RS-232 (*Recommended Standard* 232) è uno standard EIA equivalente allo standard europeo CCITT V21/V24, che definisce un'interfaccia seriale a bassa velocità per lo scambio di dati tra dispositivi digitali. EIA RS-485, equivalente allo standard Europeo CCITT V11, è una specifica Modello OSI a livello fisico di una connessione seriale a due fili, half-duplex e multipoint.

Data Bits	5, 6, 7, and 8
Stop Bits	1, 1.5[1], and 2
Parity Type	None, Even, Odd, Mark, Space
Baud Rates[2]	300, 600, 1200, 1800, 2400, 4000, 4800, 7200, 9600, 14400, 16000, 19200, 28800, 38400, 51200, 56000, 57600, 64000, 76800, 115200, 128000, 153600, 230400, 250000, 256000, 460800, 500000, 576000, 921600[3]

Notes:
1. 5-bit only.
2. Additional baud rates are supported. See "AN205: CP210x Baud Rate Support".
3. 7 or 8 data bits only.

Figura 5.10: Formato dei dati e baud rate del protocollo UART

Kevin Chapman di Xilinx) e *bbfifo16x8.vhd*. Il primo rappresenta il vero e proprio trasmettitore seriale mentre il secondo descrive una FIFO di parole a 16 byte.

```
entity uart_tx is
    Port (            data_in : in std_logic_vector(7 downto 0);
                 write_buffer : in std_logic;
                 reset_buffer : in std_logic;
                 en_16_x_baud : in std_logic;
                   serial_out : out std_logic;
                  buffer_full : out std_logic;
              buffer_half_full : out std_logic;
                          clk : in std_logic);
    end uart_tx;
```

Come già accennato, la comunicazione seriale non è sincronizzata con un segnale di clock bensì opera a un certo baud rate. Per questo motivo è richiesto che i parametri indicati in figura 5.10 siano gli stessi per trasmettitore e ricevitore.
Il trasmettitore è essenzialmente un registro a scorrimento contenente i dati che devono essere inviati. Al ricevitore di solito si adotta uno schema di sovracampionamento per individuare la posizione centrale dei bit trasmessi. Il rate di sovracampionamento è 16 volte il baud rate, perciò il bit seriale viene campionato 16 volte per memorizzarne uno soltanto (quello centrale). Fatta questa premessa, la frequenza dei bit in uscita deve risultare banalmente 16 volte più rapida della frequenza del ricevitore per rispettare questo schema, per cui se scegliamo (come è stato fatto) un baud rate del ricevitore pari a 115200 baud/s è richiesto un baud clock in trasmissione di 115200*16 = 1.8432 MHz. Per rispettare questa cadenza temporale è stato quindi generato un segnale di clock a tale frequenza a partire da un riferimento temporale a 100 MHz, come descritto nella seguente porzione di codice VHDL:

```
process(clk100temp,resetclk)
begin
if resetclk = '1' then
```

```
cnt <= 0;
elsif rising_edge(clk100temp) then
if (cnt = divideby - 1)  then
cnt <= 0;
else
cnt <= cnt + 1;
end if;
end if;
end process;

baudclk <= '1' when cnt = divideby - 1 else '0';
```

Se il mezzo fisico che supporta la trasmissione e l'apparato software dedicato sono stati descritti, passiamo ora all'elemento adibito alle operazioni di controllo sulla macchina a stati del sistema e gestione del flusso di dati in uscita: il picoBlaze. Il picoBlaze (vedi figura 5.11) è un microcontrollore fully-embedded 8-bit RISC[3] compatto, performante e flessibile utilizzabile in numerosi modelli delle famiglie di FPGA Spartan e Virtex. Se originariamente questa risorsa era stata progettata e ottimizzata per le prime schede commercializzate dalla Xilinx (Spartan-3, Virtex-2, Virtex-2 Pro), trattandosi di un modulo rilasciato come codice VHDL sintetizzabile, può essere facilmente migrato su architetture più recenti; in particolare, nel Settembre 2010 ne è stata rilasciata un nuova versione per schede Virtex-6 che fa proprio al caso del progetto in questione.

Le risorse messe a disposizione dal PicoBlaze sono:

- 2 banchi di registri general-purpose a 16 bit;
- memoria di programma fino a 4K con memorizzazione e caricamento automatico delle istruzioni durante la configurazione dell'FPGA;
- ALU (*Arithmetic Logic Unit* con flag di CARRY e di ZERO;
- scatchpad RAM a 64, 128, 256 byte;
- porte di ingresso/uscita a 8 bit;
- CALL/RETURN stack automatico a 31 locazioni;
- completa prevedibilità delle operazioni grazie all'esecuzione delle istruzioni sempre in due cicli di clock;

[3]Un dispositivo RISC (*Reduced Instruction Set Computer*) definisce tipicamente un microprocessore caratterizzato da un set di istruzioni ridotto, di formato fisso e tempo di esecuzione costante (un singolo ciclo di clock). Le architetture RISC vengono anche dette *load-store*, dal momento che permettono di accedere alla memoria unicamente tramite istruzioni specifiche (appunto, *load* e *store*) e si differenziano dai dispositivi CISC (*Complex Instruction Set Computer*), che di fatto implementano il paradigma opposto.

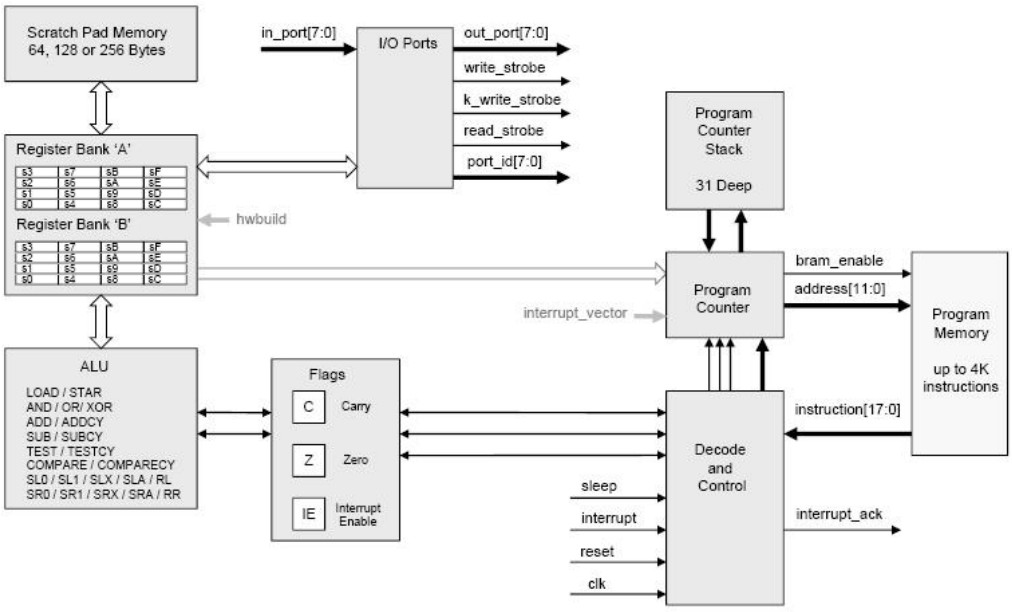


Figura 5.11: Architettura interna del picoBlaze

- massima frequenza di lavoro per un'FPGA Virtex-6 pari a 240 MHz (equivalenti a 120 MIPS);
- one mask-able interrupt response con riconoscimento;
- modalità sleep per il risparmio energetico.

Entrando più nel dettaglio dei compiti a cui questo microcontrollore è adibito, rientra nei suoi task la sincronizzazione con la macchina a stati del TDC attraverso il controllo del segnale waitingf, che, se attivo, è il preludio della fase di trasmissione. Lo stato complessivo del sistema è stato per comodità riassunto in un unico registro, *status_ register*, ed è in realtà esso a essere interrogato dal PicoBlaze. Qualora il TDC sia in attesa di trasferire i dati, le misure dei vari contatori vengono prelevate e memorizzate nell'apposita FIFO; a questo punto viene verificato se la coda della trasmissione UART è piena, e se non lo è la trasmissione può avere inizio. Una volta che tutti i dati sono stati trasmessi, è ancora il picoBlaze a inviare al sistema il reset globale per pulire i registri e riportare la macchina a stati nelle posizione di partenza. Per inserire il picoBlaze nel design del progetto è necessario definire soltanto due componenti: *kcpsm6*, che definisce il processore e le sue porte di interfaccia, e *jtag_ loader*, che implementa la memoria contenente il programma assemblato nel quale sono descritti i task da svolgere. La memoria di programma è allocata in una BRAM di dimesione variabile a seconda della tipologia di chip dell'FPGA e del programma che deve contenere. La stesura di quest'ultimo, fondamentale in quanto permette di impartire al picoBlaze le istruzioni che dovrà eseguire, va elaborata in linguaggio

assembly[4] e successivamente convertita nel corrispettivo modulo VHDL utilizzando l'applicazione *KCPSM6 Assembler*.
Qui di seguito viene presentato il file assembly con le operazioni da assegnare al picoBlaze:

```
; porta adibita alla comunicazione UART: 10,
; porta registro di sistema: 09


ADDRESS 000 ; indirizzo di inizio programma: x"000

begin:
CONSTANT UART_PORT ,10

 loop1: INPUT s0,0F          ; sys reg
TEST s0,01                   ; testo l'ultimo bit
;che rappresenta il segnale waiting
JUMP Z,loop1                 ; if  Z = 1 significa che s0 and 01 = 00
                             ; cioè waiting and 1 = 0
CALL transmit


LOAD s0,23 ;
CALL TandT

LOAD s0,0A                   ; LF ASCII SPECIAL CHARACTER
CALL TandT

LOAD s0,0D                   ; CR ASCII SPECIAL CHARACTER
CALL TandT


JUMP loop1


; Test and Transmit: per testare se la UART è piena o no e trasmettere

TandT:
INPUT sF,0F                  ; sys reg
TEST sF,20                   ; uart_tx_full è il terzo bit MSB
JUMP NZ,TandT                ; se Z = 1 => (uart_tx_full and '1') = 0
```

[4] L'assembly è il linguaggio di programmazione più vicino al linguaggio macchina. Esso sostituisce il formato binario del codice operativo del linguaggio macchina con una sequenza di caratteri che lo rappresentano in forma mnemonica.

```
;cioè coda non piena
;OUTPUT s0,10
OUTPUT s0, UART_PORT
RETURN



convert:
LOAD S2,S0
SR0 s0
SR0 s0
SR0 s0
SR0 s0
COMPARE S0,0A
JUMP C, pio
ADD S0,07
pio: ADD s0,30
call TandT
load s0,s2
AND s0,0F
COMPARE S0,0A
JUMP C, pio1
ADD S0,07
pio1: ADD s0,30
call TandT
return


  transmit:


LOAD s0,23        ; carattere speciale
CALL TandT

LOAD s0,43        ; carattere C
CALL TandT

LOAD s0,46        ; carattere F
CALL TandT

LOAD s0,23        ; carattere speciale
CALL TandT
```

```
INPUT s0,01       ; s0 <- fine counter START
CALL convert

LOAD s0,23        ; carattere speciale
CALL TandT

INPUT s0,02       ; s0 <- coarse counter STOP1
CALL convert

LOAD s0,23        ; carattere speciale
CALL TandT

INPUT s0,03       ; s0 <- fine counter STOP1
CALL convert

LOAD s0,23        ; carattere speciale
CALL TandT

INPUT s0,04       ; s0 <- coarse counter STOP2
CALL convert

LOAD s0,23        ; carattere speciale
CALL TandT

INPUT s0,05       ; s0 <- fine counter STOP2
CALL convert

LOAD s0,23        ; carattere speciale
CALL TandT

INPUT s0,06       ; s0 <- coarse counter STOP3
CALL convert

LOAD s0,23        ; carattere speciale
CALL TandT

INPUT s0,07       ; s0 <- fine counter STOP3
CALL convert

LOAD s0,23        ; carattere speciale
CALL TandT

INPUT s0,08       ; s0 <- coarse counter STOP4
```

```
CALL convert

LOAD s0,23        ; carattere speciale
CALL TandT

INPUT s0,09       ; s0 <- fine counter STOP4
CALL convert

LOAD s0,23        ; carattere speciale
CALL TandT


LOAD s0,FF       ; RESET!!!  check sul valore e sulla porta (FF)
OUTPUT s0,FF

RETURN
```

Per prima cosa, il picoBlaze deve sincronizzarsi con il processo di misura in corso. Per farlo, acquisisce il valore del registro di sistema status_register, nel quale sono contenuti i flag relativi alla macchina a stati finiti e alla gestione dell'UART, e testa l'ultimo bit, che rappresenta il segnale di waitingf. Quando questo è a livello logico alto viene richiamata la funzione transmit, che carica sul registro s0 i risultati delle varie misure, opportunamente convertiti secondo la mappatura della codifica ASCII[5] (attraverso la funzione *convert*) e intervallati da caratteri speciali. Per ogni dato trasferito viene quindi eseguito il controllo sullo stato dell'UART e, se la coda di questa non risulta piena, si procede alla trasmissione. Questo processo è definito all'interno della funzione *TandT* (*Test and Transmit*). Una volta terminata l'intera procedura, la porta di indirizzo x'FF viene settata al valore x'FF informando il TDC che la comunicazione ha avuto successo ed è possibile quindi resettare i vari registri e iniziare una nuova fase di misura.

[5]Per codifica ASCII (*American Standard Code for Information Interchange*) si intende la versione estesa a 8 bit dell'originaria codifica US-ASCII.

Capitolo 6

Test e risultati

Il Time-to-Digital Converter, dettagliatamente analizzato dal punto di vista strutturale nel capitolo precedente, richiede di essere testato per verificarne il corretto funzionamento e poterne estrapolare le reali caratteristiche e performance di misura. In questa sezione verranno trattate tali prove sperimentali, a cominciare dalla verifica sul connettore di I/O impiegato per acquisire i segnali di ingresso provenienti dagli SPAD, e successivamente presentando a livello descrittivo e grafico la misura di intervalli temporali realizzati opportunamente attraverso un setup strumentale e ottico.

6.1 Verifica delle prestazioni di I/O

La board ML605 mette a disposizione un elevato numero di pin di I/O. Rispetto alle versioni precedenti, le nuove schede che montano il chip Virtex-6 fanno uso, in particolare, di due connettori, la cui denominazione completa è VITA 57.1 FMC HPC (*High Pin Count*) e VITA 57.1 FMC LPC (*Low Pin Count*)[1].
Si tratta di due connettori a 400 pin (10x40) completamente sfruttati nel caso dell'HPC, solo parzialmente popolati (160 dei 400 pin disponibili) nel caso dell'LPC. Le varie locazioni sono destinate non soltanto ai segnali di informazione vera e propria, bensì anche ai segnali di clock e ai riferimenti di tensione per l'alimentazione e la massa, come illustrato nella mappatura di figura 6.1 e 6.2.
Il connettore FMC HPC supporta la seguente connettività:

- 160 segnali single-ended o 80 segnali differenziali;
- 10 MTG[2];
- 2 clock MTG;

[1]Questi due connettori fanno parte della serie di componenti SEARAY della Samtec (codice seriale: ASP-134486-01).
[2]Multi-Gigabit Transceiver

- 4 clock differenziali;
- 159 segnali di massa;
- 15 segnali di alimentazione.

Entrando più nel dettaglio per i segnali di informazione e di clock, vengono implementati:

- 78 coppie differenziali definite dall'utente, di cui:
  - 34 coppie LA,
  - 24 coppie HA,
  - 20 coppie HB;
- 8 MGT;
- 2 clock MGT;
- 4 clock differenziali.

Di contro, il connettore FMC LPC fornisce connettività analoga, ma ridotta in numero:

- 68 segnali single-ended o 34 segnali differenziali;
- 1 MTG;
- 1 clock MTG;
- 2 clock differenziali;
- 61 segnali di massa;
- 10 segnali di alimentazione.

Anche qui, volendo approfondire le capacità del connettore, vengono supportati:

- 34 coppie differenziali (di tipo LA) definite dall'utente;
- 1 MGT;
- 1 clock MGT;
- 2 clock differenziali.

Le ottime performance di I/O di questi componenti si possono valutare soprattutto in termini di velocità di comunicazione: le specifiche garantiscono speed rating di segnale fino a 9 GHz / 18 Gb/s per terminazioni single-ended e di massimo compreso tra 7 GHz / 15 Gb/s e 16 GHz / 32 Gb/s per pin differenziali. La tensione di riferimento è fissa e pari a 2.5 V.

Al fine di garantire la corretta acquisizione degli ingressi STOP1, STOP2, STOP3, STOP4 provenienti dagli SPAD, ciascuno associato a un canale distinto, si è voluto testare l'effettiva compatibilità di questi segnali con le caratteristiche di I/O dei connettori. In particolare, la nostra verifica si è concentrata sulle terminazioni dell'HPC. I test sono stati eseguiti implementando sul chip prima un semplice modulo equivalente a un *buffer* e poi un *linear feedback shift register* (LFSR), e sono stati volti, oltre a valutare sperimentalmente l'effettiva capacità di supportare segnali dell'ordine di centinaia di MHz, a identificare lo standard di I/O tra i seguenti:

- standard single-ended: LVCMOS, HSTL, SSTL;
- standard differenziali: LVDS, HT, LVPECL, BLVDS, HSTL differenziale, SSTL differenziale;
- standard differenziale con ingressi dipendenti dalla tensione di riferimento.

In modalità di funzionamento "buffer", al chip è stato richiesto di acquisire un segnale periodico in ingresso (un'onda quadra di frequenza variabile e livello di tensione compatibile con gli standard via via testati) e riprodurlo in uscita. Oltre ai pin del connettore HPC si è alternato l'uso di un cavo SMA, il cui collegamento è anch'esso supportato dalla scheda. Si è provato dunque a vedere tramite oscilloscopio, noto il segnale di ingresso fornito da un generatore di funzione, quale segnale venisse posto in uscita, in modo da valutare gli effetti di distorsione introdotti in output e le eventuali difficoltà in fase di ricezione con il connettore sotto esame (vedi l'esempio di figura 6.3).

La seconda parte dei test ha previsto l'implementazione di un registro a scorrimento a retroazione lineare (LFSR) per valutare le performance del connettore, questa volta soltanto in uscita. L'LFSR è un registro a traslazione di lunghezza arbitraria i cui dati di ingresso sono ottenuti dalla combinazione lineare (xor) di alcuni dei bit memorizzati. La peculiarità di questo sistema sta nel fatto che, una volta fissati i tempi che scandiscono lo scorrimento dei dati (a livello pratico, il clock che governa il processo), il vettore di bit memorizzati nel registro può non annullarsi mai, e la sequenza di valori '0' e '1' in esso contenuta risulta pseudo-casuale. Sebbene la realizzazione di questo modulo sia funzionale solamente al test sulla risposta di I/O della board, ci pare opportuno spiegare il funzionamento dell'LFSR. Lo faremo brevemente.
L'LFSR è descritto dalla lista delle posizioni dei bit che influenzano lo stato successivo dei suoi elementi memorizzati. Tale lista viene chiamata *sequenza di tap*. Se le posizioni della sequenza di tap sono scelte in maniera opportuna, l'LFSR si comporta come detto poc'anzi e viene detto LFSR *massimale*. Questa sequenza di bit può essere rappresentata nei termini del suo *polinomio caratteristico* (detto anche *polinomio di retroazione*), dove ogni posizione significativa è associata a una potenza in modulo 2 avente come esponente l'indice della stessa nella sequenza. Si può dimostrare che se il polinomio caratteristico associato al registro a scorrimento

è primitivo[3] allora quest'ultimo risulta massimale.
A titolo esemplicativo, si osservi la figura 6.4. L'LFSR illustrato ha polinomio caratteristico:

$$x^{11} + x^{13} + x^{14} + x^{16} + 1. \tag{6.1}$$

Ebbene, se si calcola la xor tra i bit evidenziati nelle posizioni corrispondenti alla sequenza di tap, e si introduce il risultato di tale operazione nel registro, shiftando tutti i dati verso destra, si ottiene una nuova sequenza che, come detto, fornisce a sua volta un bit pseudo-causale.

Tornando ora alla discussione sui test di I/O della board, l'implementazione dell'LFSR sull'FPGA e la successiva analisi dei segnali posti in uscita dal connettore ha permesso di capire quanto segue:

- i segnali di nostro interesse sono compatibili con le capacità di I/O della board;
- lo standard più adeguato per l'acquisizione dei segnali di ingresso da parte del connettore HPC, in base alle misure fatte, è l'HSTL[4].

6.2 Comportamento della carry chain

Nella valutazione delle capacità di misura del TDC realizzato, l'analisi del comportamento della carry chain riveste sicuramente un ruolo molto importante.
Si è visto nel capitolo precedente che per il corretto funzionamento della logica di riporto è necessario definire un'architettura in grado di mantenere stabili i dati relativi alla misura fine. Affinchè ciò avvenga, le uscite delle varie CARRY4 devono essere campionate contemporaneamente in corrispondenza del fronte di salita del clock di sistema, e le misure che ne scaturiscono devono seguire un percorso analogo (che richiede lo stesso tempo di transito) nell'attraversare i banchi di flip-flop che memorizzano i segnali A, C e D, da cui verrà estratto in uscita il risultato da elaborare. Il test sulla carry chain si è articolato in due fasi: una verifica in simulazione del comportamento della linea di ritardo e una serie di misure di intervalli temporali dalla cui analisi è stata ricavata la risoluzione incrementale del sistema.

La verifica a livello software ha dato un responso da un certo punto di vista sorprendente: le uscite della CARRY4 non si attivano secondo l'ordine che la logica vorrebbe, ovvero dalla prima all'ultima. Ripetendo più volte la simulazione si è notato come la prima uscita nello schema della carry sia in realtà la terza a essere

[3] Sia D un dominio. Un polinomio non nullo $f(X) \in D[X]$ si dice primitivo se soltanto gli elementi invertibili di D dividono ciascun coefficiente di f(X).

[4] HSTL (*High-Speed Transceiver Logic*) è un'interfaccia di I/O single-ended ideale per l'acquisizione dati in memoria ad alta velocità e il pilotaggio di address bus su banchi di memoria. Si distinguono, in base al voltaggio di uscita richiesto, quattro classi dello standard (I, II, III, IV), tre delle quali supportate dai dispositivi Virtex (I, III, IV). Nel nostro caso faremo riferimento alla classe HSTL_I.

campionata e la sequenza di attivazione sia 2-1-0-3 (vedi figura 6.5); questo fatto non deve, tuttavia, sorprendere più di tanto, dal momento che è semplicemente frutto di una diversa collocazione della logica (multiplexer, flip-flop, porte) interna al chip: il principio di concatenazione delle varie slice a formare una lunga linea di ritardo è comunque rispettato. Tale comportamento pare comunque dipendere dal design sviluppato, perciò è molto probabile che la definizione di una carry chain all'interno di un altro sistema sia caratterizzata da un ordine di campionamento dei flip-flop differente.
Si è visto, inoltre, che per il medesimo motivo il tempo di attraversamento del segnale di START non è lo stesso per ciascuna delle uscite della CARRY4, per cui non è possibile considerare la risoluzione fine del sistema globale pari a $\frac{1}{4}$ del ritardo introdotto dal singolo blocco della catena.

Se, dal punto di vista della struttura interna della logica di riporto, il comportamento della linea di ritardo è stato delineato, il passo successivo da verificare sarebbe stato la corretta misura di istanti temporali molto brevi - al di sotto di 2.5 ns - al fine di considerare soltanto le prestazioni dei fine counter e non avere a che fare, invece, con il risultato della composizione di componente di misura fine e grezza. Ciò non è stato possibile a causa dell'indisponibilità di cavi in grado introdurre un ritardo con queste caratteristiche.
Essendo la risoluzione fine un'informazione imprescindibile, è stata trovata una soluzione alternativa, anche se meno accurata del test sperimentale. Rimandiamo la sua trattazione al paragrafo seguente.

6.3 Taratura del sistema

Il setup predisposto per tarare il sistema ha previsto l'utilizzo di un generatore di funzioni Agilent 33220A, la cui uscita è stata sdoppiata attraverso un connettore a "T"e quindi opportunamente "differenziata"aggiungendo su uno dei due cavi coassiali BNC degli adattatori con funzione ritardatrice. In questo modo è stato possibile generare due segnali molto ravvicinati tra loro (l'entità della distanza temporale è stata via via variata per analizzare intervalli diversi) il più possibile somiglianti a quelli generati dagli SPAD durante la rivelazione di fotoni. A tal scopo, e anche per ragioni di limiti strumentali, sono stati creati impulsi di 20 ns a 5V in formato TTL[5].
Presentiamo di seguito le rilevazioni fatte a varie distanze temporali, precisando che gli arrivi sono stati analizzati in un solo canale, l'unico a essere collegato con il generatore di funzioni.
Come si può notare osservando le figure 6.6, 6.7, 6.8, 6.9 e 6.10, l'andamento delle misure è di tipo gaussiano con picco attorno al valore che ci si potrebbe attendere, ma

[5]La famiglia Transistor-Transistor Logic (TTL) è una tecnologia di circuiti integrati caratterizzata dall'adozione come elemento base del transistor, presente sia nello stadio di ingresso sia nello stadio di uscita del dispositivo logico. Tutti i circuiti TTL sono alimentati con V_{CC}=+5V e presentano in ingresso tensione di livello logico basso (V_{IL}) compresa tra 0 e +0.8V e tensione di livello logico alto (V_{IH}) compresa tra 2 e 5V.

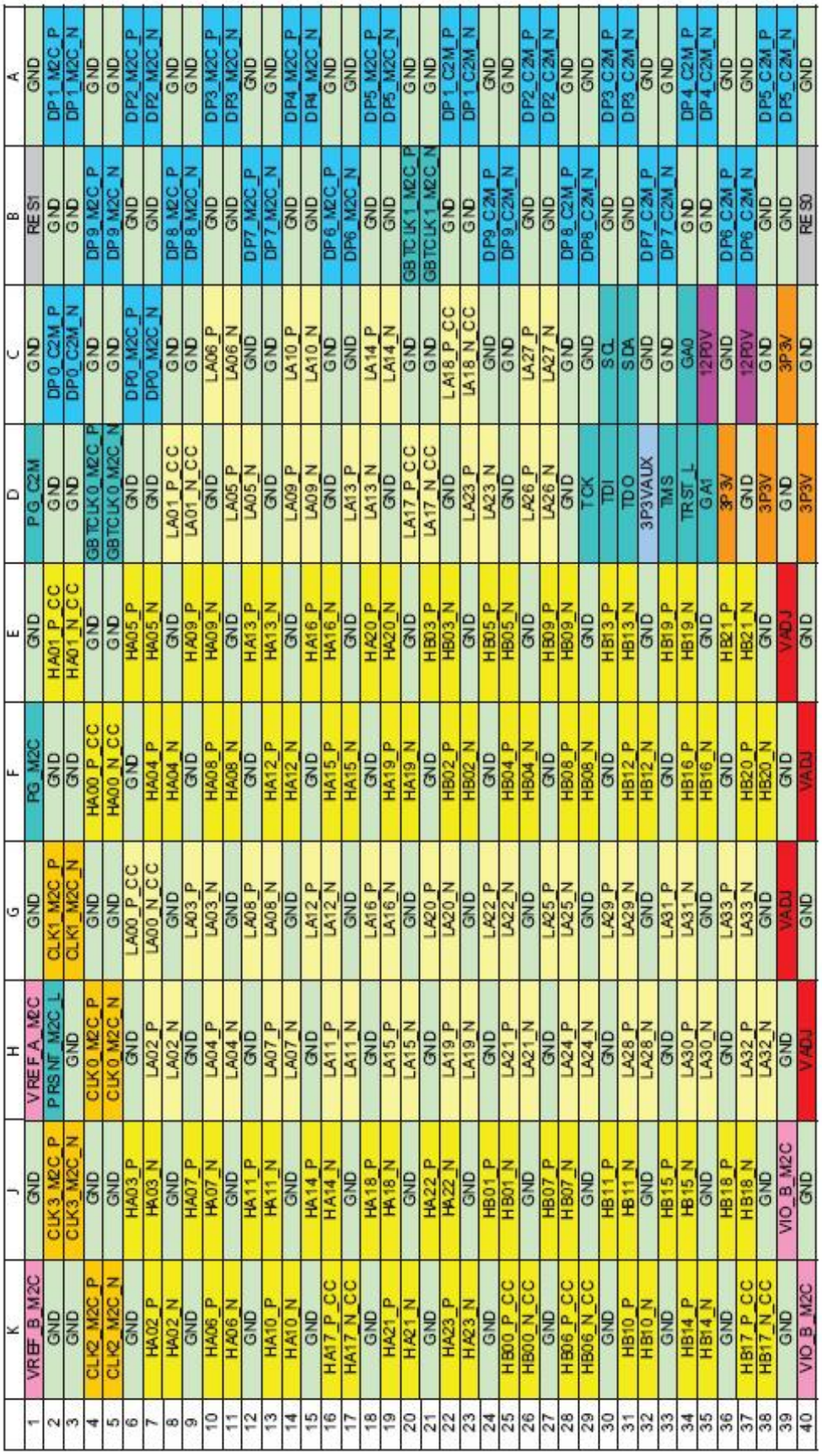
presenta alcuni scostamenti significativi, anche se sporadici, dal valore medio (calcolato su una base variabile di almeno 1500 campioni) che potrebbero essere frutto dell'errore del coarse counter. Risulta infatti che buona parte delle (per la verità poche) misure che si discostano sensibilmente dalla media statistica cadono nelle vicinanze del periodo di clock precedente e successivo a quello indicato dal coarse counter.
L'elevata varianza riscontrata, tuttavia, sembra derivare da altri fattori e potrebbe essere attribuibile al sistema di generazione del segnale. Gli impulsi che il sistema dovrebbe rilevare, infatti, a causa di limiti strumentali (in particolare la banda di funzionamento, parametro critico in questo tipo di applicazioni), presentano un tempo di salita molto alto in relazione alla loro durata; inoltre, i connettori a "T"impiegati come raccordo tra i cavi BNC che devono introdurre il ritardo di interesse danno luogo a fenomeni di afterpulsing: dunque, l'effettivo segnale generato è tutt'altro che pulito e stabile. Tenendo presente queste considerazioni non di poco conto, i risultati ottenuti appaiono compatibili e plausibili in riferimento alle condizioni in cui la misura è stata eseguita; tuttavia, se il nostro scopo era quello di effettuare una calibrazione dello strumento che tenesse conto degli scostamenti verificati rispetto ai valori temporali di riferimento, questa operazione è apparsa poco significativa, e quindi non è stata apportata alcuna correzione di default alle rilevazioni fatte.
Analizzando le misure con un post-processing via Matlab dei dati raccolti (vedi tabella 6.1), va messo in evidenza il fatto che, nel caso peggiore, in un intervallo di 440 ps si concentra il 90% degli esiti della quantificazione dell'intervallo: possiamo considerare questa durata temporale come la deviazione standard a cui è soggetta la misura nel nostro sistema.
Sempre dall'elaborazione delle misure eseguite è stato ricavato il tempo di attraversamento del singolo blocco di riporto. Per determinarlo sono state incrociate le informazioni relative alle misure dei differenti intervalli temporali e sono stati *plottati* i vari andamenti fino a che si è ottenuta una distribuzione dei dati approssimabile a una gaussiana. La risoluzione calcolata con questo metodo è stata quantificata in circa 40 ps.
Se dal punto di vista pratico la carry chain permette di rilevare, dunque, variazioni temporali di circa 40 ps (anche meno in realtà, ma come detto il tempo di attraversamento dei singoli blocchi relativi alle 4 uscite non è omogeneo e quindi non può essere tenuto presente nel computo della risoluzione), sperimentalmente le oscillazioni rispetto al valore medio risultano ben più ampie e sono descritte proprio dal valore della deviazione standard (o, in alternativa, della varianza). E' per questo motivo che nel tirare le fila sulle prestazioni del nostro TDC nel capitolo conclusivo parleremo distintamente di *risoluzione incrementale teorica* e *incertezza* dovuta alle condizioni di misura.

lunghezza [m]	0.25	1	1.5	3	4
misura oscilloscopio [ps]	2600	6000	10200	21500	29000
misura TDC [ps]	3024	5495	10779	22434	29174
incertezza [ps]	420	440	440	220	440

Tabella 6.1: Analisi delle misure: nella prima riga lo sfasamento in termini di lunghezza di cavo aggiunto per creare il ritardo temporale, nella seconda la durata di riferimento di tale intervallo misurata tramite oscilloscopio, nella terza l'esito della misura da parte del TDC e nell'ultima la risoluzione complessiva del sistema, all'interno della quale è contenuto almeno il 90% delle misure eseguite.

6.4 Setup ottico per la rivelazione di fotoni

Per avere un riscontro più vicino alla realtà del fenomeno che vogliamo osservare, la misura di intervalli temporali ottenuti ritardando, mediante cavi di lunghezza diversa, uno stesso segnale proveniente da un generatore di funzioni non ci è parso sufficiente. Non avendo a disposizione una vera sorgente di fotoni entangled sulla falsa riga di quella descritta nel Capitolo 2, è stato messo a punto un setup ottico in grado di generare e, successivamente, rilevare la presenza di fotoni in intervalli temporali molto ridotti, compatibili con le specifiche del TDC sviluppato. Osservando la figura esplicativa 6.11, la sorgente è un laser a diodo World Star Tech a 850 nm e 3.5 mW di potenza che genera un fascio nell'infrarosso a frequenza impostabile in maniera arbitraria. Per ottenere un flusso di fotoni singoli è stato frapposto nella direzione di emissione un filtro neutro di attenuazione 4 (in scala logaritmica, a cui corrisponde quindi un riduzione di intensità di 10^4). Il fascio viene fatto propagare in spazio libero per un metro circa e quindi fatto incidere su un beam splitter non polarizzante con *split ratio* pari al 50%. A questo punto ciascun fotone ha la stessa probabilità di essere deviato di 90° lateralmente verso destra oppure proseguire lungo la direzione di emissione del laser: in entrambi i casi è presente uno SPAD MPD con efficienza del 10% alla lunghezza d'onda di interesse (vedi la caratteristica di rivelazione in figura 6.11) e dead time pari a 70 ns che lo raccoglie e pone il segnale elettrico frutto della conversione in ingresso al connettore HPC dell'FPGA Virtex-6 in nostro possesso.
Con questa configurazione vengono trasmesse coppie di fotoni alla frequenza di 1 MHz. Tenendo presente che l'efficienza di rivelazione degli SPAD nel caso specifico è del 10% e che la probabilità di individuare una coincidenza comporta di fatto un'ulteriore riduzione di un fattore 10 (perchè la probabilità che il clic degli SPAD sia relativo alla coppia di fotoni sparati assieme è $\frac{1}{10}\cdot\frac{1}{10}$), il rate finale di osservazione di coppie di fotoni nello stesso intervallo analizzato è all'incirca pari a 10 KHz.
Questo rapporto è stato effettivamente verificato, fornendo un'ulteriore riscontro sul funzionamento del TDC.



	K	J	H	G	F	E	D	C	B	A
1	VREF_B_M2C	GND	VREF_A_M2C	GND	PG_M2C	GND	PG_C2M	GND	RESt	GND
2	GND	CLK3_M2C_P	PRSNT_M2C_L	CLK1_M2C_P	GND	HA01_P_CC	GND	DP0_C2M_P	GND	DP1_M2C_P
3	GND	CLK3_M2C_N	GND	CLK1_M2C_N	GND	HA01_N_CC	GND	DP0_C2M_N	GND	DP1_M2C_N
4	CLK2_M2C_P	GND	CLK0_M2C_P	GND	HA00_P_CC	GND	GBTCLK0_M2C_P	GND	DP9_M2C_P	GND
5	CLK2_M2C_N	GND	CLK0_M2C_N	GND	HA00_N_CC	GND	GBTCLK0_M2C_N	GND	DP9_M2C_N	GND
6	GND	HA03_P	GND	LA00_P_CC	GND	HA05_P	GND	DP0_M2C_P	GND	DP2_M2C_P
7	HA02_P	HA03_N	LA02_P	LA00_N_CC	HA04_P	HA05_N	GND	DP0_M2C_N	GND	DP2_M2C_N
8	HA02_N	GND	LA02_N	GND	HA04_N	GND	LA01_P_CC	GND	DP8_M2C_P	GND
9	GND	HA07_P	GND	LA03_P	GND	HA09_P	LA01_N_CC	GND	DP8_M2C_N	GND
10	HA06_P	HA07_N	LA04_P	LA03_N	HA08_P	HA09_N	GND	LA06_P	GND	DP3_M2C_P
11	HA06_N	GND	LA04_N	GND	HA08_N	GND	LA05_P	LA06_N	GND	DP3_M2C_N
12	GND	HA11_P	GND	LA08_P	GND	HA13_P	LA05_N	GND	DP7_M2C_P	GND
13	HA10_P	HA11_N	LA07_P	LA08_N	HA12_P	HA13_N	GND	GND	DP7_M2C_N	GND
14	HA10_N	GND	LA07_N	GND	HA12_N	GND	LA09_P	LA10_P	GND	DP4_M2C_P
15	GND	HA14_P	GND	LA12_P	GND	HA16_P	LA09_N	LA10_N	GND	DP4_M2C_N
16	HA17_P_CC	HA14_N	LA11_P	LA12_N	HA15_P	HA16_N	GND	GND	DP6_M2C_P	GND
17	HA17_N_CC	GND	LA11_N	GND	HA15_N	GND	LA13_P	GND	DP6_M2C_N	GND
18	GND	HA18_P	GND	LA16_P	GND	HA20_P	LA13_N	LA14_P	GND	DP5_M2C_P
19	HA21_P	HA18_N	LA15_P	LA16_N	HA19_P	HA20_N	GND	LA14_N	GND	DP5_M2C_N
20	HA21_N	GND	LA15_N	GND	HA19_N	GND	LA17_P_CC	GND	GBTCLK1_M2C_P	GND
21	GND	HA22_P	GND	LA20_P	GND	HB03_P	LA17_N_CC	GND	GBTCLK1_M2C_N	GND
22	HA23_P	HA22_N	LA19_P	LA20_N	HB02_P	HB03_N	GND	LA18_P_CC	GND	DP1_C2M_P
23	HA23_N	GND	LA19_N	GND	HB02_N	GND	LA23_P	LA18_N_CC	GND	DP1_C2M_N
24	GND	HB01_P	GND	LA22_P	GND	HB05_P	LA23_N	GND	DP9_C2M_P	GND
25	HB00_P_CC	HB01_N	LA21_P	LA22_N	HB04_P	HB05_N	GND	GND	DP9_C2M_N	GND
26	HB00_N_CC	GND	LA21_N	GND	HB04_N	GND	LA26_P	LA27_P	GND	DP2_C2M_P
27	GND	HB07_P	GND	LA25_P	GND	HB09_P	LA26_N	LA27_N	GND	DP2_C2M_N
28	HB06_P_CC	HB07_N	LA24_P	LA25_N	HB08_P	HB09_N	GND	GND	DP8_C2M_P	GND
29	HB06_N_CC	GND	LA24_N	GND	HB08_N	GND	TCK	GND	DP8_C2M_N	GND
30	GND	HB11_P	GND	LA29_P	GND	HB13_P	TDI	SCL	GND	DP3_C2M_P
31	HB10_P	HB11_N	LA28_P	LA29_N	HB12_P	HB13_N	TDO	SDA	GND	DP3_C2M_N
32	HB10_N	GND	LA28_N	GND	HB12_N	GND	3P3VAUX	GND	DP7_C2M_P	GND
33	GND	HB15_P	GND	LA31_P	GND	HB19_P	TMS	GND	DP7_C2M_N	GND
34	HB14_P	HB15_N	LA30_P	LA31_N	HB16_P	HB19_N	TRST_L	GA0	GND	DP4_C2M_P
35	HB14_N	GND	LA30_N	GND	HB16_N	GND	GA1	12P0V	GND	DP4_C2M_N
36	GND	HB18_P	GND	LA33_P	GND	HB21_P	3P3V	GND	DP6_C2M_P	GND
37	HB17_P_CC	HB18_N	LA32_P	LA33_N	HB20_P	HB21_N	GND	12P0V	DP6_C2M_N	GND
38	HB17_N_CC	GND	LA32_N	GND	HB20_N	GND	3P3V	GND	GND	DP5_C2M_P
39	GND	VIO_B_M2C	GND	VADJ	GND	VADJ	GND	3P3V	GND	DP5_C2M_N
40	VIO_B_M2C	GND	VADJ	GND	VADJ	GND	3P3V	GND	RESO	GND

Figura 6.1: Pinout del connettore FMC HPC.

	K	J	H	G	F	E	D	C	B	A
1	NC	NC	VREF_A_M2C	GND	NC	NC	PG_C2M	GND	NC	NC
2	NC	NC	PRSNT_M2C_L	CLK1_M2C_P	NC	NC	GND	DP0_C2M_P	NC	NC
3	NC	NC	GND	CLK1_M2C_N	NC	NC	GND	DP0_C2M_N	NC	NC
4	NC	NC	CLK0_M2C_P	GND	NC	NC	GBTCLK0_M2C_P	GND	NC	NC
5	NC	NC	CLK0_M2C_N	GND	NC	NC	GBTCLK0_M2C_N	GND	NC	NC
6	NC	NC	GND	LA00_P_CC	NC	NC	GND	DP0_M2C_P	NC	NC
7	NC	NC	LA02_P	LA00_N_CC	NC	NC	GND	DP0_M2C_N	NC	NC
8	NC	NC	LA02_N	GND	NC	NC	LA01_P_CC	GND	NC	NC
9	NC	NC	GND	LA03_P	NC	NC	LA01_N_CC	GND	NC	NC
10	NC	NC	LA04_P	LA03_N	NC	NC	GND	LA06_P	NC	NC
11	NC	NC	LA04_N	GND	NC	NC	LA05_P	LA06_N	NC	NC
12	NC	NC	GND	LA08_P	NC	NC	LA05_N	GND	NC	NC
13	NC	NC	LA07_P	LA08_N	NC	NC	GND	GND	NC	NC
14	NC	NC	LA07_N	GND	NC	NC	LA09_P	LA10_P	NC	NC
15	NC	NC	GND	LA12_P	NC	NC	LA09_N	LA10_N	NC	NC
16	NC	NC	LA11_P	LA12_N	NC	NC	GND	GND	NC	NC
17	NC	NC	LA11_N	GND	NC	NC	LA13_P	GND	NC	NC
18	NC	NC	GND	LA16_P	NC	NC	LA13_N	LA14_P	NC	NC
19	NC	NC	LA15_P	LA16_N	NC	NC	GND	LA14_N	NC	NC
20	NC	NC	LA15_N	GND	NC	NC	LA17_P_CC	GND	NC	NC
21	NC	NC	GND	LA20_P	NC	NC	LA17_N_CC	GND	NC	NC
22	NC	NC	LA19_P	LA20_N	NC	NC	GND	LA18_P_CC	NC	NC
23	NC	NC	LA19_N	GND	NC	NC	LA23_P	LA18_N_CC	NC	NC
24	NC	NC	GND	LA22_P	NC	NC	LA23_N	GND	NC	NC
25	NC	NC	LA21_P	LA22_N	NC	NC	GND	GND	NC	NC
26	NC	NC	LA21_N	GND	NC	NC	LA26_P	LA27_P	NC	NC
27	NC	NC	GND	LA25_P	NC	NC	LA26_N	LA27_N	NC	NC
28	NC	NC	LA24_P	LA25_N	NC	NC	GND	GND	NC	NC
29	NC	NC	LA24_N	GND	NC	NC	TCK	GND	NC	NC
30	NC	NC	GND	LA29_P	NC	NC	TDI	SCL	NC	NC
31	NC	NC	LA28_P	LA29_N	NC	NC	TDO	SDA	NC	NC
32	NC	NC	LA28_N	GND	NC	NC	3P3VAUX	GND	NC	NC
33	NC	NC	GND	LA31_P	NC	NC	TMS	GND	NC	NC
34	NC	NC	LA30_P	LA31_N	NC	NC	TRST_L	GA0	NC	NC
35	NC	NC	LA30_N	GND	NC	NC	GA1	12P0V	NC	NC
36	NC	NC	GND	LA33_P	NC	NC	3P3V	GND	NC	NC
37	NC	NC	LA32_P	LA33_N	NC	NC	GND	12P0V	NC	NC
38	NC	NC	LA32_N	GND	NC	NC	3P3V	GND	NC	NC
39	NC	NC	GND	VADJ	NC	NC	GND	3P3V	NC	NC
40	NC	NC	VADJ	GND	NC	NC	3P3V	GND	NC	NC

Figura 6.2: Pinout del connettore FMC LPC.

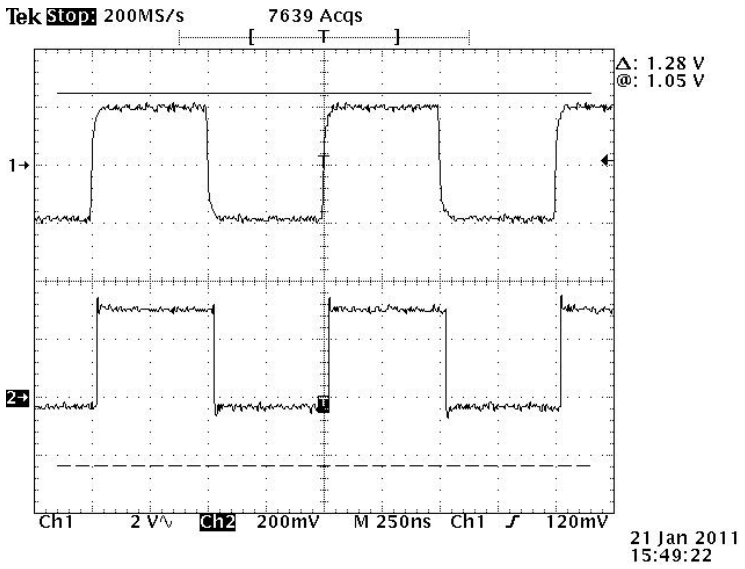


Figura 6.3: Test sulle prestazioni di I/O del connettore FMC HPC. L'oscilloscopio presenta il segnale di ingresso sul canale 1 e il segnale di uscita sul canale 2 (x10).

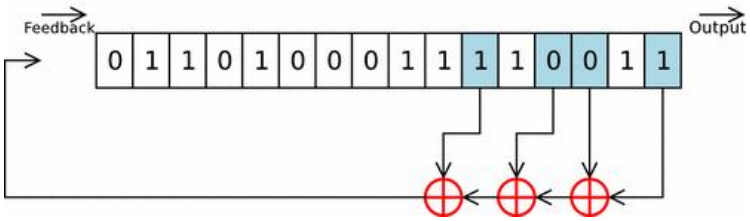


Figura 6.4: Rappresentazione di un LFSR con polinomio caratteristico $x^{11} + x^{13} + x^{14} + x^{16} + 1$.

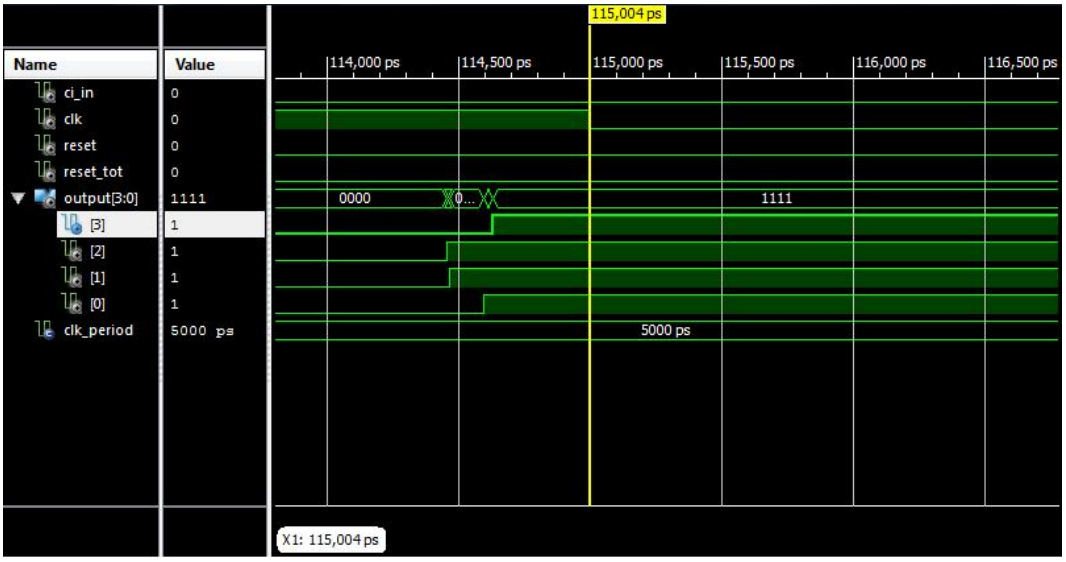
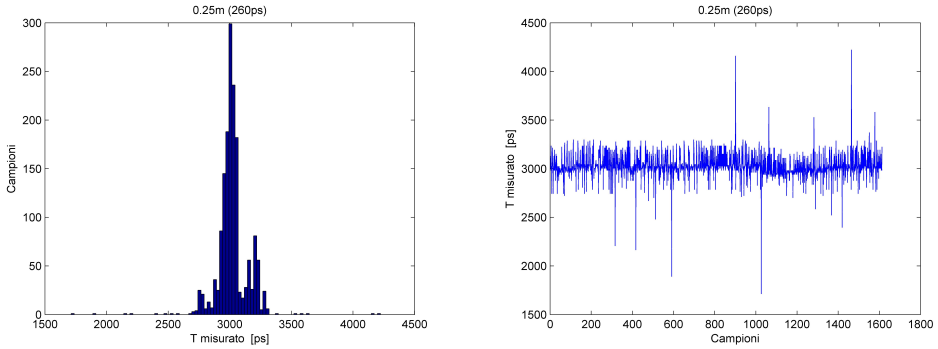
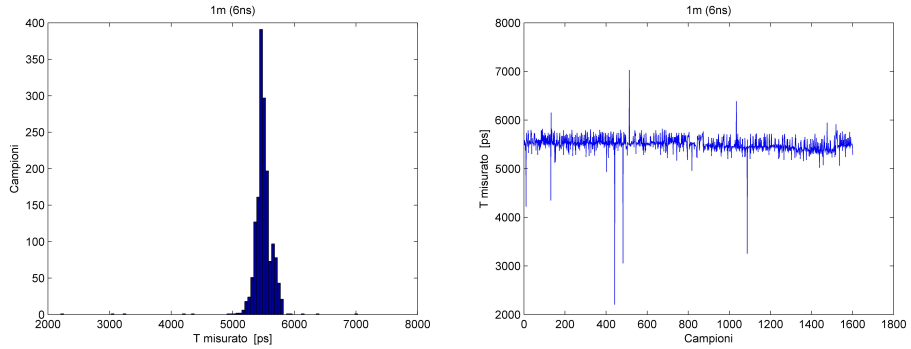


Figura 6.5: Simulazione post-route di una CARRY4.



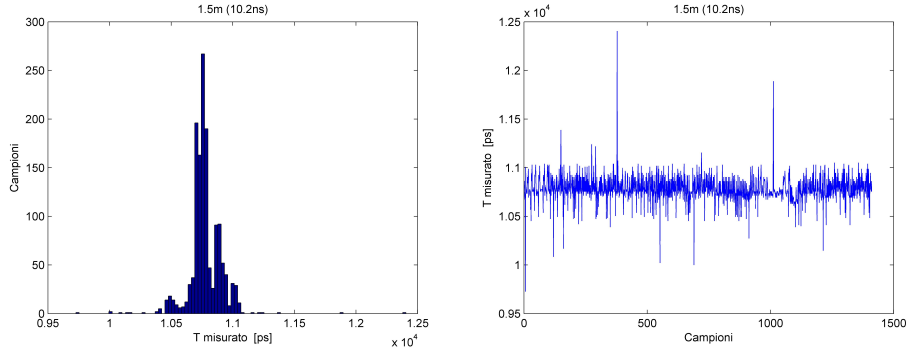
(a) *Istogramma.* (b) *Plot.*

Figura 6.6: Misure relative a uno sfasamento di 25 cm, corrispondenti a un intervallo temporale di circa 260 ps.



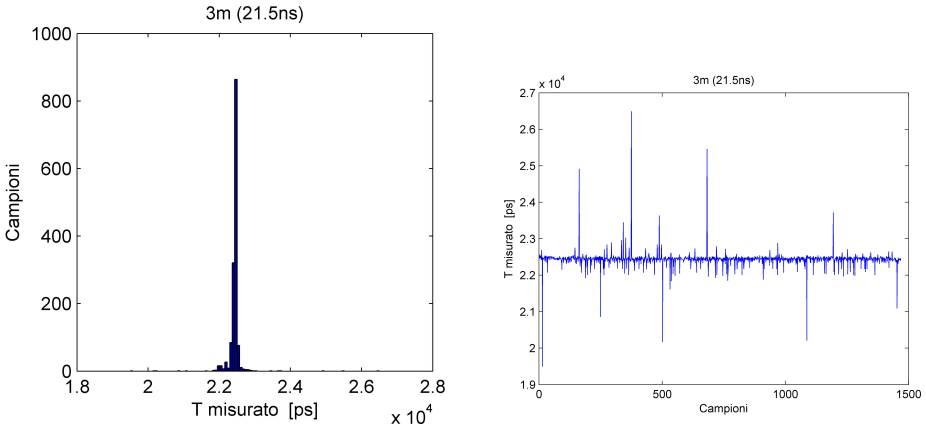
(a) *Istogramma.* (b) *Plot.*

Figura 6.7: Misure relative a uno sfasamento di 1 m, corrispondenti a un intervallo temporale di circa 6 ns.



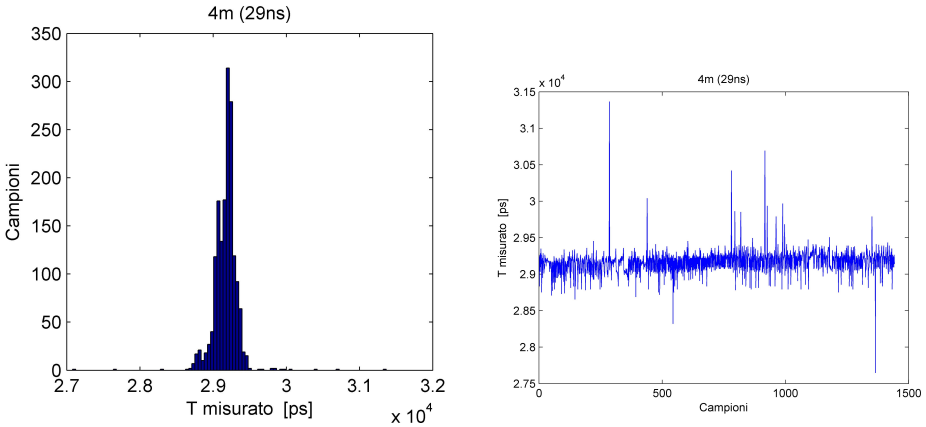
(a) *Istogramma.* (b) *Plot.*

Figura 6.8: Misure relative a uno sfasamento di 1.5 m, corrispondenti a un intervallo temporale di circa 10.2 ns.



(a) *Istogramma.* (b) *Plot.*

Figura 6.9: Misure relative a uno sfasamento di 3 m, corrispondenti a un intervallo temporale di circa 21.5 ns.



(a) *Istogramma.* (b) *Plot.*

Figura 6.10: Misure relative a uno sfasamento di 4 m, corrispondenti a un intervallo temporale di circa 29 ns.

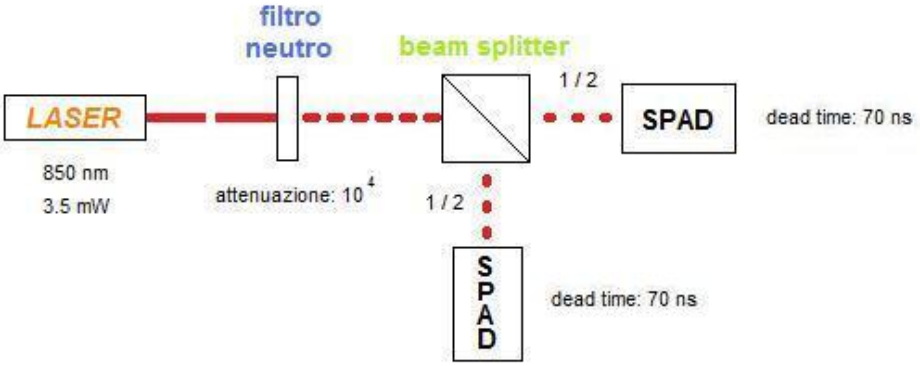


Figura 6.11: Schema del setup ottico per la generazione e rivelazione di fotoni in istanti ravvicinati.

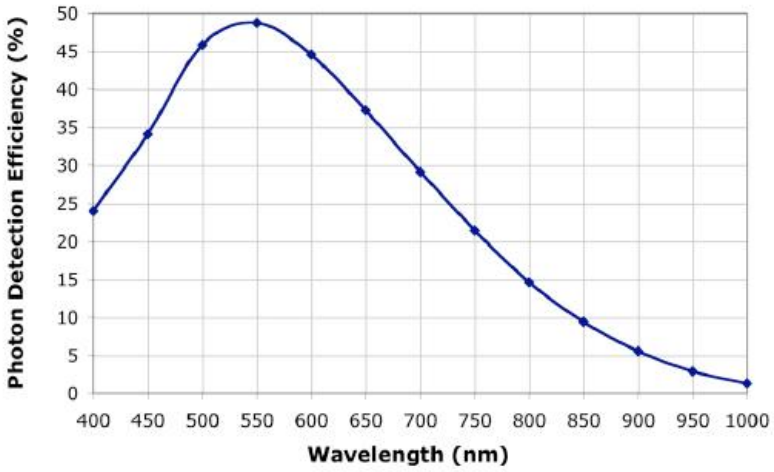


Figura 6.12: Efficienza di rivelazione dello SPAD MPD utilizzato in funzione della lunghezza d'onda.

Capitolo 7

Conclusioni

I TEST e l'analisi delle misure eseguite dal Time-to-Digital Converter realizzato danno sicuramente adito ad alcune considerazioni sulle capacità reali di questo sistema, sui punti dolenti che il design attuale presenta e sui prossimi step che avranno questo progetto come punto di partenza.
Va premesso che l'obiettivo perseguito in questo lavoro è stato quello di raggiungere le massime performance compatibili con le caratteristiche dell'FPGA a nostra disposizione. Tuttavia, partendo da questo presupposto, è stato necessario ricorrere a qualche compromesso progettuale mano a mano che il TDC ha preso forma.

Per iniziare, sembra giusto sottolineare il fatto che un dispositivo con un simile potenziale richiede una strumentazione di contorno adeguata. Dalle informazioni raccolte durante il collaudo del sistema, infatti, si sono verificate alcune situazioni controverse a cui non sempre si è potuta dare una spiegazione assolutamente chiarificatrice.
Prendiamo in considerazione la risoluzione globale del nostro TDC: assunto con buona approssimazione che il tempo di percorrenza della singola CARRY4 sia costante, teoricamente è possibile rilevare variazioni di 40 ps dell'intervallo temporale in esame; tuttavia, è stata accertata un'incertezza nelle misure di un ordine di grandezza più alto. Sebbene questo fatto sia stato spiegato e per forza di cose accettato stante la strumentazione a disposizione, si tratta di uno status operandi che non può essere tollerato se si vuole raggiungere una caratterizzazione completa e accurata del dispositivo. Una volta discriminata questa "tara", il risultato dei conteggi potrà essere veramente inputato quasi esclusivamente all'apparato di misura.
Anche la supposizione stessa di tempo di attraversamento fisso del blocco di carry chain non è stata verificata con i fatti: dalla documentazione fornita è noto solamente il massimo ritardo che può essere introdotto dall'elemento, mentre il suo reale valore è stato quantificato in post-processing sui dati raccolti, con un margine d'errore al momento non trascurabile. A tal proposito, va altresì detto che la varianza sull'esito della misura, con le condizioni attuali per la stessa, avrebbe reso questo problema probabilmente secondario.
Queste limitazioni in fase sperimentale, comunque, potranno essere superate senza

problemi affidando la fase di generazione e detection via cavo a un'apparecchiatura adeguata, oppure più semplicemente concentrandosi sul setup ottico, che dovrebbe garantire condizioni più favorevoli, oltre a essere l'ambiente naturale in cui il sistema va messo alla prova. Per la caratterizzazione della linea di ritardo si è pensato di mantenere lo stadio di emissione di coppie di fotoni già descritto in precedenza, e creare arbitrariamente il ritardo temporale da rilevare allungando il percorso del fascio in una delle due direzioni dopo il beam splitter, attraverso un'opportuna serie di specchi.

Anche a livello progettuale si possono identificare lacune e punti su cui lavorare. Di sicuro tra questi va annoverata la presente modalità di trasmissione dei dati: la camunicazione UART, pur tenendo conto del dead time dei fotorivelatori che di fatto ne limita la frequenza di misura, risulta inefficiente e rappresenta al momento il collo di bottiglia del sistema. La scelta più consona è il passaggio alla trasmissione via Ethernet che permette di raggiungere il Gbit/s a fronte dell'attuale baud rate di 115200 bit/s del trasferimento dati seriale. Questa opzione progettuale porterebbe a una maggiore complessità nella gestione del payload (richiedendo tra l'altro la definizione di protocolli di livello più alto), ma sicuramente l'efficienza del TDC ne gioverebbe, permettendo uno smaltimento più rapido dei dati immagazzinati in memoria. Proprio questo punto rappresenta un'altro nodo cruciale, dal momento che, allo stato attuale del design, una volta che la FIFO viene completamente riempita, le successive misure vengo perse se nel frattempo la trasmissione (molto più lenta della fase di acquisizione ed elaborazione delle misure) non ha provveduto a creare spazio svuotandola almeno in parte. Anche senza scomodare un diverso protocollo di comunicazione, tuttavia, è possibile ovviare a tale questione definendo per l'UART degli appositi segnali di controllo della memoria (RTS e CTS), peraltro suggeriti dalla stessa casa produttrice.
Un metodo alternativo allo schema di Nutt implementato in questo progetto potrebbe essere preso in considerazione nel caso (già prospettato) in cui si volesse ridurre la finestra di osservazione degli arrivi: l'impiego di lunghe carry chain in grado di coprire l'intero range di misura sarebbe una soluzione interessante per aggirare il problema dell'uniformità dei dati provenienti da fine e coarse counter, a patto di caratterizzare adeguatamente il comportamento della logica di riporto interna al chip, in accordo con le considerazioni appena fatte.

Lo sviluppo futuro del TDC passerà senza ombra di dubbio attraverso il processing on-board dei risultati della misura, dal momento che l'interesse applicativo si sta spostando verso il conteggio del numero di occorrenze osservate in un dato periodo lungo (che potrebbe essere 1 s e quindi significare frequenza di coincidenza), piuttosto che fossilizzarsi sul miglioramento dell'accuratezza di rivelazione.

Bibliografia

[1] Bouwmeester D. Ekert A. Zeilinger A. The Physics of Quantum Information. *Università di Vienna*, 5 2001.

[2] Walls D. F. Milburn J. G. Quantum Optics. *Physics Department - University of Queensland*, 2, 2008.

[3] Foster A. S. Allen M. S. Chu D.C. Universal counter resolves picoseconds in time interval measurements. *Hewlett-Packard J.*, 29 210, 1980.

[4] Cicalese R. Giordano R. Izzo V. Loffredo S. Aloisio A. Branchini P. Fpga implementation of a high-resolution timte-to-digital converter. *IEEE Nuclear Science Symposium Conference Record*, 2007.

[5] Maatta K. Kostamovaara J. Accurate time interval measurement electronics for pulsed time of flight laser radar. *Oulu University.*

[6] Ikeno M. Arai Y. A time digitizer cmos gate-array with 250 ps time risolution. *IEEE J. Solid State Circuits*, 31 21220, 1996.

[7] Brian F. Aull Andrew H. Loomis Douglas J. Young Richard M. Heinrichs Bradley J. Felton Peter J. Daniels Deborah J. Landers Geiger-Mode Avalanche Photodiodes for Three-Dimensional Imaging. *Lincoln Laboratory Journal*, 13 2, 2002.

[8] Alexander V. Sergienko Quantum Communications and Cryptography. *Taylor & Francis Group*, 10, 2006.

[9] Kalisz J. Szplet R. Pasierbinski J. Poniecki A. Field-Programmable-Gate-Array-Based Time-to-Digital Converter with 200-ps Resolution. *IEEE Transaction of Instrumentation and Measurement*, 46 1, 1997.

[10] Dudek P. Szczezepanski S. Hartfield J. V. A High-Resolution CMOS Time-to-Digital Converter Utilizing a Vernier Delay Line. *IEEE J. Solid State Circuits*, 35 2, 2000.

[11] Iafolla L. Sistema di acquisizione elettronico per lo studio della Fisica gamma-gamma con KLOE2 presso DAϕNE. *Tesi laurea Specialistica in Fisica - Università di Roma*, 2009.

[12] Pezzini M. A High-Resolution FPGA Time-tagging System for Quantum Optics Applications. *Tesi di dottorato, Laurea Specialistica in ing.Informatica - Università di Padova*, 2009.

[13] Merlin A. Sviluppo di un sistema di time-tagging implementato su FPGA Virtex 5 per l'analisi di correlazioni quantistiche. *Tesi laurea Specialistica in ing. Elettronica*, 2009.

[14] Orazi M. Studio e progettazione dell'elettronica di lettura dei contatori di tempo di volo dell'esperimento PAMELA per la ricerca di antimateria nello spazio. *Tesi laurea in Fisica - Università di Napoli*, 2002.

[15] Gugole L. Time-to-Digital Converter su FPGA: studio e confronto di possibili architetture. *Tesi laurea in ing. Elettronica - Università di Padova*, 2009.

[16] Hewlett-Packard Company. The fundamentals of time interval measurement. *Hewlett-Packard J.*, 29 210, 1997.

[17] Zhou W. Xuan Z. Yu J. Some new methods for precision time interval measurement. *IEEE International Frequency Control Symposium*, 1997.

[18] Genat J. F. Rossel F. Ultra high-speed time-to-digital converter. *French Patent*, 84 07344, 1984.

[19] Baron R. G. The Vernier time-measuring technique. *Proc. IRE*, pp2130, 1957.

bibitem Eichel H. W. Horstmann J. U. Coates R. L. Metastability behaviour of CMOS ASIC ip-ops in theory and test. *IEEE J. Solid-State Circuits*, 24 14657, 1989.

[20] Kostamovaara J. Myllyla R. Time-to-digital with an analog interpolation circuit. *Rev. Sci Instrum.*, 57 28805, 1986.

[21] Pelka R. Kalisz J. Szplet R. Poniecki A. Single-chip interpolating time counter with 200-ps resolution and 43-s range. *IEEE transaction of Instrumentation and Measuremnt*, 46 4 , 1997.

[22] Kalisz J. Review of methods for time interval measurement with picosecond resolution. *Metrologia*, 41 17-32, 2004.

[23] Pawowski M. Kalisz J. Peka R. A multiple-interpolation method for fast and precise time digitizing. *IEEE Trans. Instrum. Meas.*, 35 1639, 1986.

[24] Pawowski M. Kalisz J. Peka R. Error analysis and design of the Nutt time-interval digitiser with picosecond resolution. *Phys. E. Sci. Instrum.*, 20 133041, 1987.

[25] Mota M Christiansen J. A high-resolution time interpolator based on a delay locked loop and RC delay line. *IEEE J. Solid State Circuits*, 34 13606, 1999.

[26] Mota M Christiansen J. A four channel, self calibrating, high resolution, time-to-digital converter. *Proc. 1998 Int. Conf. Electronics, Circuits and Systems*, Vol.1, pp 40912, 1998.

[27] Antti Mantyniemi. An integrated CMOS high precision time-to-digital converter based on stabilized three stagedelay line interpolation. *Oulu University*, 2004.

[28] Hoppe D. R. Differential time interpolator. *US Patent*, 4.433,919.

[29] Nutt R. Digital time intervalometer. *Rev. Sci. Instrum.*, 39 12425, 1968.

[30] Nutt R. Digital time intervalometer with analogue Vernier timing. *US Patent*, 3.541,448.

[31] Xilinx. ML605 Hardware User Guide. Version 1.4, 2010.

[32] Xilinx. Virtex-6 Family Overview. Version 1.0, 2009.

[33] Xilinx. PlanAhead User Guide. Version 11.4, 2009.

[34] Xilinx. PicoBlaze 8-bit Embedded Microcontroller User Guide. Version 2.0, 2010.

[35] Silicon Labs. Single-Chip USB to UART Bridge. Rev.1.0.

[36] Xilinx. Virtex-6 FPGA Configurable Logic Block. Version 1.1, 2009.

[37] Xilinx. Virtex-6 FPGA Packaging and Pinout Specifications. Version 2.3, 2010.

[38] Xilinx. Virtex-6 FPGA SelectIO Resources. Version 1.3, 2010.

[39] Xilinx. Virtex-6 FPGA Data Sheet: DC and Switching Characteristics. Version2.10, 2010.

[40] Lacaita A. Samori C. Zappa F. Cova S. Ghioni M. Avalanche photodioded and quenching ircuits for single-photon detection. *Appl. Opt.*, 35 no12:1956-1976, 1996.

Indice analitico