

**POLITECNICO DI MILANO**  
**Scuola di Ingegneria Industriale e dell'Informazione**  
**Corso di Laurea MAGISTRALE in Ingegneria Informatica**



**Sviluppo firmware per misuratore laser di  
distanza basato su FPGA**

**Relatore: Prof. Michele Norgia**

**Tesi di Laurea di:  
Leonardo Cavagnis, matricola 816646  
Diego Rondelli, matricola 817108**

**Anno Accademico 2014-2015**



*Alle nostre famiglie*



# Sommario

Lo scopo di questo lavoro di tesi è sviluppare il firmware di un misuratore laser basato sulla tecnica di interferometria a retroiniezione per misurare la distanza assoluta di un bersaglio.

La prima parte del lavoro consiste nell'implementazione del firmware, sviluppato usando NI LabVIEW FPGA e Real-Time, e degli algoritmi necessari per la misura della distanza assoluta.

Nella seconda parte ci si è concentrati sull'ottimizzazione degli algoritmi implementati e sulla calibrazione dei parametri di funzionamento del sistema, al fine di migliorare la precisione e l'accuratezza della misura.



# Abstract

The aim of this project is to develop the firmware of a laser sensor based on the *self-mixing* interferometer technique to measure the absolute distance from a target.

The first part of the work consists in the implementation of the firmware, developed using NI LabVIEW FPGA and Real-Time, and the algorithms required for the absolute distance measurement.

The second part is focused on the optimization of the implemented algorithms and the calibration of the working parameters of the system, in order to improve the precision and the accuracy of the measurement.



# Ringraziamenti



# Indice

<b>Introduzione</b>	<b>1</b>
<b>1 Principi di Laser e Telemetria</b>	<b>3</b>
1.1 Principi di funzionamento del laser . . . . .	3
1.1.1 Emissione stimolata di radiazione . . . . .	4
1.1.2 Inversione di popolazione . . . . .	7
1.1.3 Cavitá ottica e materiale attivo . . . . .	8
1.2 Laser a semiconduttore . . . . .	10
1.2.1 Laser Fabry-Perot . . . . .	11
1.2.2 Laser DFB . . . . .	12
1.2.3 Laser VCSEL . . . . .	12
1.3 Classi di sicurezza dei laser . . . . .	13
1.4 Telemetri ottici . . . . .	14
1.4.1 Telemetri a triangolazione . . . . .	15
1.4.2 Telemetri a tempo di volo . . . . .	16
1.4.3 Telemetri a onda continua . . . . .	18
1.5 Una tecnica alternativa per la misura . . . . .	18
<b>2 Interferometria a retroiniezione</b>	<b>21</b>
2.1 Principi di interferometria convenzionale . . . . .	21
2.1.1 Svantaggi dell'interferometria convenzionale . . . . .	23
2.2 Interferometria a retroiniezione . . . . .	24
2.2.1 Regimi di retroiniezione tramite risoluzione qualitativa	26
2.2.2 Vantaggi e svantaggi dell'interferometria a retroiniezione	33
2.3 Principali limitazioni della tecnica interferometrica . . . . .	34
2.4 Applicazioni dell'interferometria a retroiniezione . . . . .	37
2.4.1 Misura di distanza assoluta . . . . .	38
<b>3 Architettura Hardware dello strumento</b>	<b>43</b>
3.1 Struttura complessiva dello strumento . . . . .	43

3.2	Parte analogica . . . . .	45
3.2.1	Sistema ottico e sorgente laser . . . . .	45
3.2.2	Circuito di interfacciamento . . . . .	47
3.3	Parte di Conversione . . . . .	47
3.3.1	Convertitori . . . . .	49
3.4	Parte digitale . . . . .	60
3.4.1	Scheda di prototipazione utilizzata . . . . .	61
3.4.2	FPGA . . . . .	62
3.4.3	Xilinx Spartan-6 LX45 . . . . .	70
3.4.4	Microcontrollore . . . . .	70
<b>4</b>	<b>Architettura Software dello strumento</b>	<b>75</b>
4.1	Ambiente di sviluppo LabVIEW . . . . .	75
4.2	Sistema Real-Time e FPGA . . . . .	77
4.2.1	Programmazione del microcontrollore . . . . .	77
4.2.2	Progettazione e programmazione dell'FPGA . . . . .	82
4.3	Analisi teorica degli algoritmi implementati . . . . .	88
4.3.1	Fast Fourier Transform (FFT) . . . . .	89
4.3.2	Interpolated Fast Fourier Transform (IFFT) . . . . .	95
4.3.3	Calcolo della distanza assoluta . . . . .	98
4.4	Architettura Software . . . . .	98
4.4.1	FPGA . . . . .	99
4.4.2	Microcontrollore . . . . .	110
4.4.3	Comunicazione tra FPGA e Microcontrollore . . . . .	113
<b>5</b>	<b>Misure effettuate e dati sperimentali</b>	<b>117</b>
	<b>Conclusioni e sviluppi futuri</b>	<b>119</b>
<b>A</b>	<b>Documentazione software</b>	<b>121</b>
	<b>Bibliografia</b>	<b>122</b>

# Elenco delle figure

1.1	Sistema atomico composto da due livelli . . . . .	5
1.2	Schema di funzionamento di un atomo eccitato: assorbimento, emissione spontanea ed emissione stimolata . . . . .	6
1.3	Sistema atomico a tre livelli energetici . . . . .	7
1.4	Sistema atomico a quattro livelli energetici . . . . .	8
1.5	Cavità di Fabry-Perot . . . . .	9
1.6	Schema del Roundtrip ottico nella cavità ottica . . . . .	9
1.7	Propagazione della radiazione all'interno della cavità laser . .	10
1.8	Laser Fabry-Perot . . . . .	11
1.9	Laser DFB . . . . .	12
1.10	Laser VCSEL . . . . .	12
1.11	Classi di sicurezza dei laser . . . . .	14
1.12	Schema di funzionamento di un telemetro a triangolazione . .	15
1.13	Schema di funzionamento di un telemetri a tempo di volo . .	17
1.14	Potenza trasmessa e potenza riflessa in un telemetro a onda continua . . . . .	17
2.1	Interferometro di <i>Michelson</i> . . . . .	22
2.2	Ambiguità sul verso di spostamento . . . . .	24
2.3	Schema di principio di un interferometro a retroiniezione . .	25
2.4	Rappresentazione vettoriale dell'interferenza all'interno della cavità laser . . . . .	25
2.5	Round trip ottico del campo elettrico all'interno della cavità laser . . . . .	26
2.6	Perturbazione della frequenza reale rispetto alla frequenza ideale	29
2.7	Regimi di retroiniezione: funzione $F(\Phi)$ al variare del parametro $C$ . . . . .	32
2.8	Esempi di segnale interferometrico per i differenti valori del parametro $C$ . . . . .	32
2.9	Diagramma di Wegel . . . . .	34
2.10	Esempio di salti di fase dell'onda nel tempo . . . . .	34

2.11	Speckle causato da una superficie diffusiva . . . . .	35
2.12	Rappresentazione esemplificata del fenomeno dello speckle-pattern . . . . .	37
3.1	Risultato finale del misuratore realizzato . . . . .	44
3.2	Schema a blocchi dell'architettura complessiva del misuratore	45
3.3	Struttura del dispositivo <i>WLSD-1550-020m-1-PD</i> . . . . .	46
3.4	Sistema ottico con laser e lente . . . . .	46
3.5	Schema a blocchi del circuito di interfacciamento . . . . .	47
3.6	Foto della scheda di conversione <i>SCO Board</i> e identificazione delle connessioni . . . . .	48
3.7	Schema a blocchi del sistema di conversione A/D - D/A, <i>SCO Board</i> . . . . .	49
3.8	Schema di funzionamento di un convertitore Digitale-Analogico (DAC) . . . . .	50
3.9	Struttura circuitale di un DAC a resistori pesati . . . . .	50
3.10	Struttura circuitale di un DAC a scala R-2R . . . . .	52
3.11	Schema a Blocchi interno del DAC - DAC902 . . . . .	53
3.12	Schema di funzionamento di un convertitore Analogico-Digitale (ADC) . . . . .	54
3.13	Schema di funzionamento di un ADC a conteggio . . . . .	55
3.14	Schema di funzionamento di un ADC ad approssimazioni successive . . . . .	56
3.15	Schema di funzionamento di un ADC a doppia rampa . . . . .	57
3.16	Schema di funzionamento di un ADC Flash . . . . .	58
3.17	Schema di funzionamento di un ADC Pipeline . . . . .	59
3.18	Schema di funzionamento del ADS807 . . . . .	60
3.19	Scheda di elaborazione digitale, National Instruments Single-Board RIO (sbRIO) 9636 . . . . .	61
3.20	Tipologie di PLA . . . . .	63
3.21	Struttura semplificata del chip FPGA . . . . .	65
3.22	Static Memory . . . . .	66
3.23	Implementazione del floating gate . . . . .	67
3.24	Schema a blocchi dell'architettura del microcontrollore MPC5125	71
4.1	Esempio di LabVIEW VI che calcola la somma di due numeri in virgola mobile . . . . .	76
4.2	Esempio di Nodo e di Collegamento . . . . .	76
4.3	Schema di funzionamento dei due approcci: RTOS e Bare-metal	78
4.4	Confronto tra kernel monolitico e micro-kernel . . . . .	80

4.5	Struttura del sistema operativo VxWorks . . . . .	80
4.6	Struttura del modulo LabVIEW Real-Time . . . . .	81
4.7	Architettura software di un applicazione LabVIEW Real-Time	82
4.8	Cicli deterministici e non-deterministici . . . . .	82
4.9	Sintesi hardware . . . . .	85
4.10	High Level Synthesis . . . . .	88
4.11	Processo di compilazione LabVIEW FPGA . . . . .	89
4.12	Segnale campionato con treno di impulsi . . . . .	89
4.13	Spectral leakage . . . . .	92
4.14	Finestra di Hanning . . . . .	94
4.15	Spectral leakage per diversi tipi di finestratura . . . . .	94
4.16	Confronto tra DFT e FT di un segnale a frequenza $\omega_0$ . . . . .	95
4.17	Schema a blocchi ad alto livello dell'architettura . . . . .	98
4.18	Block Diagram semplificato del VI eseguito su FPGA . . . . .	104
4.19	Pannello di configurazione per la creazione di una FIFO . . . . .	105
4.20	Esempio di Producer/Consumer implementato su LabVIEW FPGA . . . . .	106
4.21	Ingressi ed uscite di un express VI con esecuzione pipelined .	107
4.22	Computazione FFT . . . . .	108
4.23	Pannello di configurazione del FFT Express VI . . . . .	108
4.24	FFT Express VI Timing Diagram . . . . .	109
4.25	Confronto dello spazio occupato da un sommatore su FPGA con Fixed Point e Floating Point . . . . .	109
4.26	Fixed point in LabVIEW . . . . .	110
4.27	Interfaccia grafica dello strumento di misura . . . . .	113
4.28	Schema architetturale per DMA . . . . .	114



# Elenco delle tabelle

3.1	Valori di utilizzo di area dell'FPGA . . . . .	71
-----	------------------------------------------------	----



# Introduzione

Il lavoro di tesi qui presentato trae origine dall'esperienza svolta presso il "Laboratorio di Misure Ottiche ed Elettroniche - MOLES" del Dipartimento di Elettronica Informazione e Bioingegneria del Politecnico di Milano, nell'ambito dello studio e del progetto di un sistema di misura laser di distanza mediante tecnica interferometrica a retroiniezione. Lo strumento in questione deriva dalle conoscenze acquisite con un'attività di ricerca che si sviluppa da diversi anni [17].

Grazie alla scarsa invasività delle sorgenti laser e alla loro elevata adattabilità ai vari ambienti di lavoro, il loro utilizzo è richiesto in numerose applicazioni, che spaziano dagli ambiti biomedicali alle telecomunicazioni, fino ad arrivare alla pura sensoristica. Sebbene nel mercato ci siano diverse tipologie di misuratori di distanza ottici, sfruttati grazie alla loro capacità di misurazione senza perturbazioni o interventi meccanici, la tecnica interferometrica a retroiniezione consente caratteristiche e prestazioni differenti. È una tecnica recente che permette di effettuare una misura di distanza assoluta utilizzando solamente un laser, un fotodiodo e una lente. Il costo dei componenti è esiguo grazie alle tecnologie elettroniche analogiche e digitali moderne e alla semplicità del sistema ottico.

L'obiettivo di questa tesi è lo sviluppo di una versione dello strumento che si prefigge di raggiungere il massimo delle prestazioni ottenibili e di raffinare altri aspetti come affidabilità, qualità hardware e software. In quanto note a priori le problematiche da affrontare e le specifiche che ogni componente avrebbe dovuto soddisfare, è stato possibile svolgere il lavoro in maniera ordinata e precisa.

L'attività è stata ripartita con un altro laureando, Samuele Disegna, che si è occupato della parte elettronica e ottica dello strumento, mentre questo lavoro tratta la parte software.

Gli argomenti sviluppati sono organizzati in 5 capitoli principali.

Nel **Capitolo 1** è presente una descrizione delle caratteristiche fisiche e ottiche delle sorgenti laser. Nel **Capitolo 2** sono descritti i principi base

dell'interferometria, con particolare attenzione a quella utilizzata, la retroiniezione. Nei **Capitoli 3 e 4** sono descritte l'architettura hardware e software dello strumento. Nel **Capitolo 5**, infine, sono illustrate le prove sperimentali.

Milano, Dicembre 2015

*Leonardo Cavagnis  
Diego Rondelli*

# Capitolo 1

## Principi di Laser e Telemetria

*In questo Capitolo verranno richiamati i concetti fondamentali relativi al funzionamento delle sorgenti laser. Verranno quindi descritte le principali tipologie di sorgenti e le classi di sicurezza che ne regolamentano l'utilizzo. Successivamente, verrà esposto lo stato dell'arte dei misuratori ottici di distanza assoluta, ponendo particolare attenzione alle tecniche di misura a triangolazione, a tempo di volo e a onda continua. In conclusione, quest'ultime verranno messe a confronto con una tecnica alternativa, l'interferometria a retroiniezione, al fine di motivare l'utilizzo di tale tecnica per questo lavoro di Tesi.*

### 1.1 Principi di funzionamento del laser

LASER é l'acronimo di *Light Amplification by Stimulated Emission of Radiation*, cioè amplificazione della luce mediante il fenomeno dell'emissione stimolata di radiazione [42]. In generale, un laser é un dispositivo elettronico in grado di emettere un fascio di luce. Un fascio di luce é un'onda elettromagnetica costituita da particelle chiamate fotoni. Il fotone é il quanto di energia della radiazione elettromagnetica.

Il fascio di luce emesso da un dispositivo laser possiede tre caratteristiche principali:

1. Coerenza
2. Monocromaticitá
3. Direzionalitá

Per quanto riguarda la prima caratteristica ci si riferisce in particolare a due aspetti differenti: la coerenza spaziale e la coerenza temporale. La

coerenza spaziale esprime la correlazione tra i valori che il campo elettromagnetico emesso assume in due punti diversi dello spazio, mentre per coerenza temporale si definisce la correlazione tra i valori che il campo elettromagnetico assume in due istanti di tempo diversi. Un onda é coerente spaziale se esiste una differenza di fase costante tra due punti qualunque sul fronte d'onda. Mentre, la coerenza temporale é strettamente legata al tempo di coerenza. Esso é l'intervallo medio di tempo nel quale l'onda compirá un certo numero di oscillazioni prima di cambiare fase.

La seconda proprietá garantisce che tutti i fotoni generati per effetto dell'emissione stimolata risultino iso-frequenziali (aventi tutti la stessa frequenza). Essa é strettamente correlata alla coerenza temporale.

La terza e ultima proprietá, invece, afferma che la radiazione emessa dalla sorgente si propaga nello spazio in un'unica direzione ben definita con piccoli angoli di divergenza del fascio parallelo e perpendicolare, piú precisamente, l'angolo solido sotteso da un fascio laser é estremamente piccolo. Essa é strettamente correlata alla coerenza spaziale.

Queste proprietá fondamentali di una sorgente laser sono dovute al principio di funzionamento stesso: l'emissione stimolata di radiazione.

### 1.1.1 Emissione stimolata di radiazione

Il fenomeno che permette il funzionamento del laser é l'emissione stimolata, per questo é importante capire di cosa si tratta e in che modo essa é differente dall'emissione spontanea. Per fare ciò é conveniente richiamare alcuni concetti di base.

Come accennato in precedenza, la propagazione della luce nello spazio si sviluppa tramite particelle dette fotoni (o quanti), aventi ciascuna una energia pari a:

$$E = h\nu \quad (1.1)$$

dove  $h = 6.62 \times 10^{-34} \text{ Js}$  é la costante di Planck, mentre  $\nu$  é la frequenza della radiazione. A sua volta la frequenza  $\nu$  é in relazione alla lunghezza d'onda  $\lambda$  tramite la relazione:

$$\nu = \frac{c}{\lambda} \quad (1.2)$$

dove  $c = 2.99 \times 10^8 \text{ m/s}$  rappresenta la velocitá della luce nel vuoto.

Un laser é un sistema ottico costituito da due specchi separati da un mezzo attivo, il quale puó essere un solido, un liquido, un gas o un semiconduttore. Si consideri il mezzo attivo come un insieme di atomi (sistema atomico) e per semplicitá di trattazione si ipotizzi che il materiale attivo in questione, investito dalla radiazione, abbia due soli livelli energetici dotati

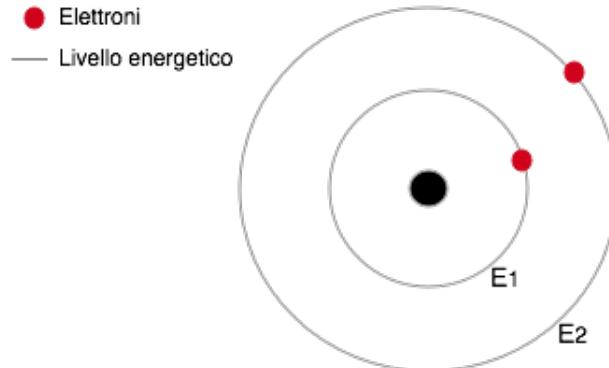


Figura 1.1: Sistema atomico composto da due livelli

rispettivamente di energia  $E_1$  e  $E_2$  con  $E_2 > E_1$ . Definiamo lo stato ad energia inferiore  $E_1$  come stato fondamentale e lo stato ad energia superiore  $E_2$  come stato eccitato.

Quando la luce investe un materiale, si possono verificare due differenti tipi di transizione:

- *Assorbimento*: Quando il fascio di luce investe un materiale, parte dell'energia posseduta dal fascio viene ceduta. L'assorbimento consiste nel cedere, da parte del fotone, la propria energia al sistema atomico permettendo ad un singolo elettrone di passare da uno stato fondamentale  $E_1$  ad uno stato eccitato  $E_2$ .
- *Emissione*: Il sistema atomico cede energia al campo, in questo caso si possono avere due possibili scenari:
  - *Spontanea*: L'emissione spontanea si verifica quando un elettrone torna dal livello  $E_2$  al livello  $E_1$ , provocando l'emissione di un fotone, senza nessun campo di radiazione incidente su di esso. Essa viene definita anche emissione incoerente: l'energia viene emessa con fase e direzione casuale.
  - *Stimolata*: L'emissione stimolata, invece, si ottiene quando un fotone incidente causa la discesa di un elettrone dal livello  $E_2$  al livello  $E_1$  ottenendo così due fotoni alla stessa frequenza. Quindi, in breve, un fotone colpisce l'atomo e due fotoni lo lasciano. Essa viene definita anche emissione coerente: l'energia viene emessa con la stessa fase, frequenza e direzione.

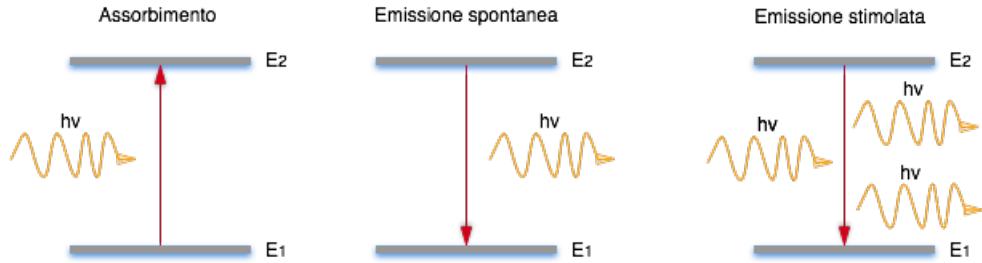


Figura 1.2: Schema di funzionamento di un atomo eccitato: assorbimento, emissione spontanea ed emissione stimolata

L'importanza dell'emissione stimolata sta nel fatto che essendo prodotti due fotoni iso-frequenziali, di fatto avviene un'amplificazione ottica, la quale consente il funzionamento del sistema laser.

Per comprendere i meccanismi che regolano lo scambio di energia tra la radiazione ed il sistema atomico, è necessario introdurre la statistica di Boltzmann [24]. Attraverso tale statistica è possibile definire la popolazione, in termini di densità, di un livello energetico all'equilibrio termico mediante la seguente relazione:

$$N = N_0 e^{-\frac{E}{kT}} \quad (1.3)$$

dove  $N_0$  è la popolazione iniziale in un dato livello energetico e  $k = 1.38 \cdot 10^{-23} \frac{J}{K}$  è la costante di Boltzmann.

Indicando, quindi, con  $N_1$  e  $N_2$  il numero di atomi per unità di volume (popolazione atomica) per i rispettivi livelli  $E_1$  e  $E_2$ , possiamo ricavare il rapporto tra il numero di atomi nello stato fondamentale e quello nello stato eccitato, all'equilibrio termodinamico alla temperatura  $T$ , tramite la seguente equazione:

$$\frac{N_2}{N_1} = e^{-\frac{\Delta E}{kT}} \quad (1.4)$$

dove  $\Delta E = E_2 - E_1$ .

La condizione necessaria per avere emissione stimolata è l'inversione di popolazione tra i due livelli energetici, ovvero, il numero di atomi presenti nel livello eccitato deve essere maggiore di quello fondamentale ( $N_2 > N_1$ ).

In un sistema a due livelli questa condizione non è realizzabile poiché, come possiamo osservare dall'equazione 1.4 a causa della presenza dell'esponenziale negativo,  $N_1$  risulta sempre maggiore di  $N_2$ , ovvero gli atomi a energia minima sono maggiori rispetto a quelli eccitati. È evidente dunque che, per avere una maggior probabilità di emissione rispetto all'assorbimento, sia necessaria un'inversione di popolazione, ovvero  $N_2 > N_1$ .

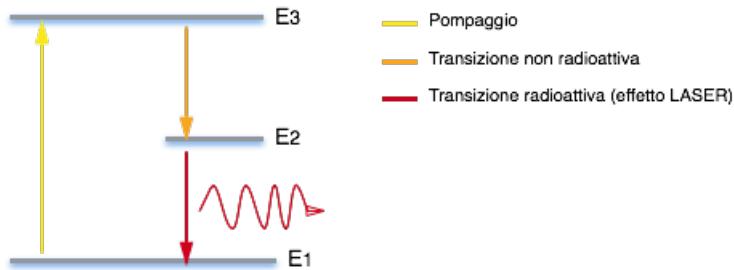


Figura 1.3: Sistema atomico a tre livelli energetici

Per ottenere la condizione di inversione di popolazione é quindi necessario l'utilizzo di un sistema atomico con almeno tre livelli energetici.

### 1.1.2 Inversione di popolazione

L'idea di base dell'inversione di popolazione consiste nello sfruttare i diversi tempi di vita medi dei differenti stati energetici.

Considerando un sistema ad esattamente tre livelli energetici ( $E_1$ ,  $E_2$  e  $E_3$ ) schematizzato in Figura 1.3, si sottopone il sistema atomico ad una radiazione luminosa di frequenza  $\nu_{31}$ , corrispondente al gap energetico  $E_3 - E_1$ , in modo tale che gli elettroni dallo stato  $E_1$  si eccitino raggiungendo lo stato instabile  $E_3$ . Questa prima fase è chiamata *pompaggio*.

Successivamente si manifesta un decadimento dallo stato instabile  $E_3$  allo stato  $E_2$  in tempi molto rapidi e privi di emissione spontanea. Solitamente l'energia rilasciata viene trasferita sotto forma di moto vibrazionale al materiale circostante e non come fotone emesso.

Dal livello  $E_2$  si verifica l'emissione laser alla frequenza  $\nu_{21}$  e la conseguente regressione dal livello  $E_2$  al livello  $E_1$ . La condizione necessaria, sui tempi di vita medi, per il funzionamento del laser è  $\tau_{32} \ll \tau_{21}$ , ovvero che il decadimento da  $E_3$  a  $E_2$  sia piú rapido di quello da  $E_2$  a  $E_1$ . Tale condizione permette di avere inversione di popolazione, ovvero  $N_2 > N_1$ , cosí da poter innescare l'amplificazione ottica e, quindi, l'effetto laser alla frequenza  $\nu_{21}$ .

Questo metodo é inefficiente perché richiede un pompaggio elevato: é necessario fornire un numero elevato di elettroni al livello  $E_3$  in modo tale che possa donarli al livello  $E_2$  permettendo cosí l'inversione di popolazione. Un struttura piú efficiente é quella a quattro livelli schematizzata in Figura 1.4. In questa disposizione, il pompaggio degli elettroni avviene da  $E_1$  a  $E_4$  a cui segue una transizione rapida e senza radiazione verso  $E_3$  che consente di popolare il livello energetico. A questo punto avviene la transizione lenta

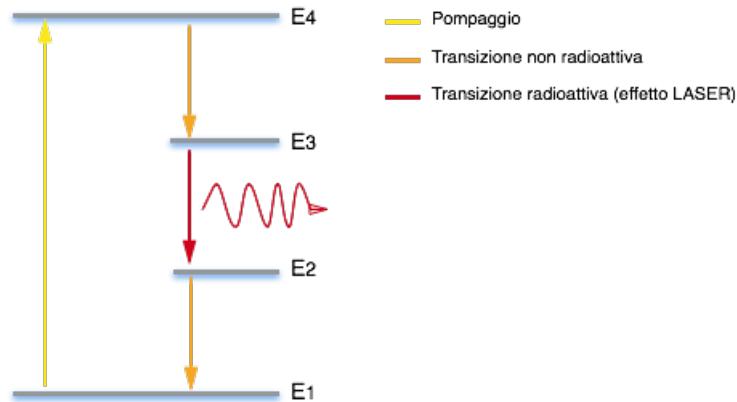


Figura 1.4: Sistema atomico a quattro livelli energetici

e l'emissione tramite azione laser. Infine avviene una rapida transizione da  $E_2$  allo stato fondamentale  $E_1$ .

A differenza della struttura a tre livelli, questa ha il vantaggio di avere il livello  $E_3$  popolato mentre il livello  $E_2$  vuoto ( $N_2 = 0$ ), in quanto gli elettroni decadono rapidamente dallo stato  $E_2$  allo stato  $E_1$ , aiutando a conservare l'inversione di popolazione  $N_3 \gg N_2$ .

Oltre al meccanismo appena descritto, altre caratteristiche risultano cruciali nel funzionamento del laser: la cavitá ottica e il materiale attivo.

### 1.1.3 Cavitá ottica e materiale attivo

Nel paragrafo precedente abbiamo visto come realizzare un materiale amplificatore, che sfrutta l'inversione di popolazione nei livelli energetici. Per generare il fascio laser é necessario, però, inserire il materiale attivo all'interno di una cavitá ottica. Una semplice cavitá ottica può essere rappresentata dalla cavitá a specchi piani e paralleli, nota anche come cavitá di Fabry-Perot, mostrata in Figura 1.5. Il fascio di luce viaggia avanti-indietro riflettendosi negli specchi e amplificandosi nel passaggio all'interno del materiale attivo. Il fascio laser in uscita si ottiene rendendo uno dei due specchi della cavitá ottica parzialmente trasparente, in modo tale che parte della radiazione esca dalla cavitá.

Per definire il guadagno ottico del materiale attivo é necessario introdurre però alcuni concetti preliminari. Indicando con  $I$  l'intensità luminosa, definiamo l'intensità della luce all'interno della cavitá ottica con la seguente relazione:

$$I(l) = I(0)e^{\sigma(N_2 - N_1)l} \quad (1.5)$$

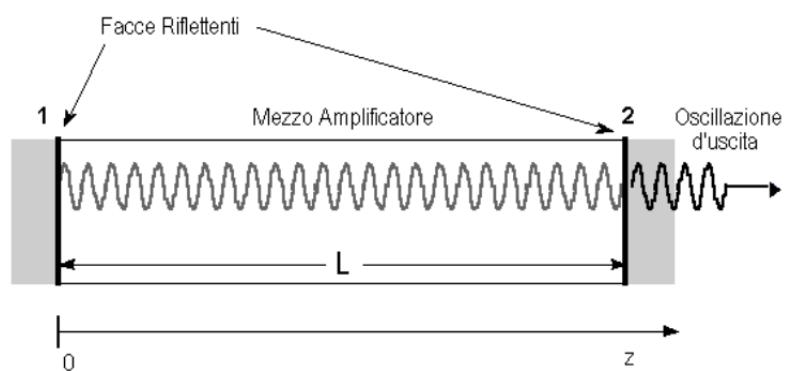


Figura 1.5: Cavità di Fabry-Perot

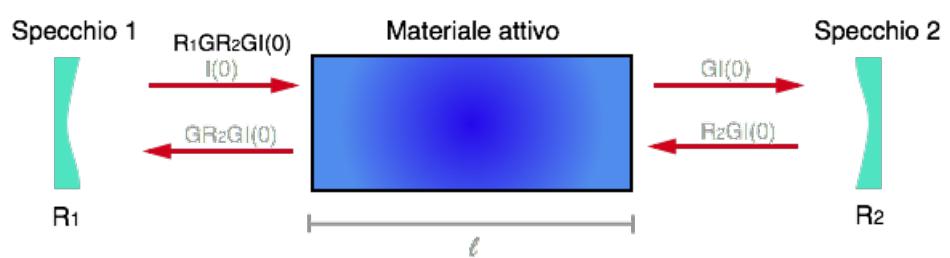


Figura 1.6: Schema del Roundtrip ottico nella cavità ottica

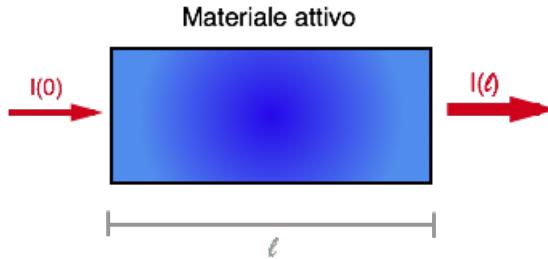


Figura 1.7: Propagazione della radiazione all'interno della cavità laser

dove  $\sigma$  é la la *cross section* di emissione e  $l$  é la lunghezza della cavitá ottica. Possiamo definire il guadagno ottico del mezzo attivo come:

$$G = \frac{I(l)}{I(0)} = e^{\sigma(N_2 - N_1)l} \quad (1.6)$$

sulla quale graveranno le limitazioni dovute alle perdite del materiale e alla perdita utile di potenza dovuta al laser uscente.

La cavitá ottica e il mezzo attivo formano quindi un oscillatore ottico, ossia un amplificatore che viene retro-azionato positivamente attraverso specchi riflettenti posti ai lati del materiale attivo. Quindi, per avere un'azione laser è necessario raggiungere una situazione in cui il guadagno di amplificazione sia tale da compensare tutte le perdite presenti.

Considerando come perdite solamente la riflettività parziale degli specchi  $R_1$  e  $R_2$ , l'oscillazione ottica si innesca quando il guadagno del materiale attivo supera le perdite della cavitá in un giro completo, o *round trip*:

$$G^2 = \frac{1}{R_1 R_2} \quad (1.7)$$

Quest'ultimo è chiamato guadagno critico o inversione critica.

## 1.2 Laser a semiconduttore

Un dispositivo laser a semiconduttore utilizza una giunzione p-n come materiale attivo all'interno della cavitá ottica. Il pompaggio avviene tramite la ricombinazione di elettroni e lacune, che produce fotoni ad una lunghezza d'onda dipendente dal *gap* tra i livelli energetici del dispositivo: la frequenza della radiazione emessa è legata quindi al tipo di materiali impiegati [42].

In commercio esistono diverse tipologie di laser a semiconduttore. Esse differiscono sulla base di due aspetti fondamentali: il tipo di cavitá ottica

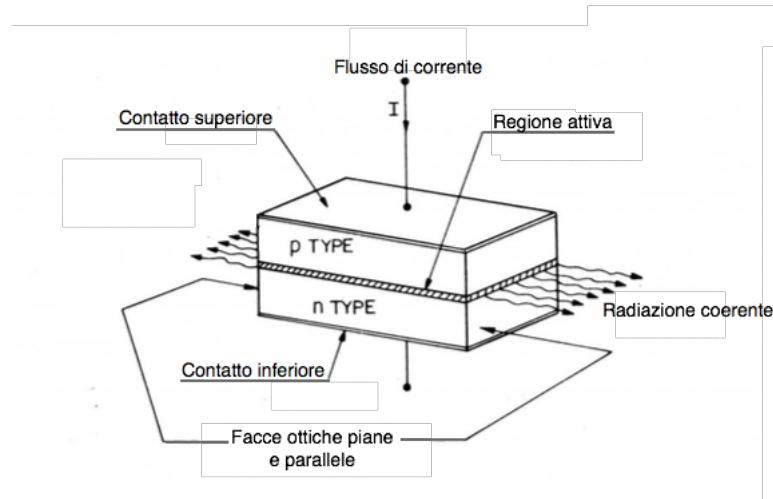


Figura 1.8: Laser Fabry-Perot

impiegata e la forma degli specchi semi-riflettenti necessari a realizzare la retroazione ottica.

Sulla base di queste due caratteristiche si possono individuare tre categorie principali di laser:

1. Laser Fabry-perot
2. Laser DFB
3. Laser VCSEL

### 1.2.1 Laser Fabry-Perot

I laser Fabry-Perot hanno la classica configurazione con cavità orizzontale a specchi piani, come mostrato in Figura 1.8.

Il principio di funzionamento é analogo a quello di un oscillatore ottico, descritto per esteso nel paragrafo precedente.

In particolare, la luce presente in cavità stimola alcuni atomi, già eccitati dalla corrente di pompa, ad emettere un fotone isofrequenziale con stessa fase di quello incidente. Gli specchi agli estremi della cavità rendono possibile la retroazione positiva, infatti fanno in modo che si possano instaurare alcuni modi di oscillazione stabili, o talvolta solo uno.

Questo può rappresentare uno svantaggio in applicazioni come l'interferometria.

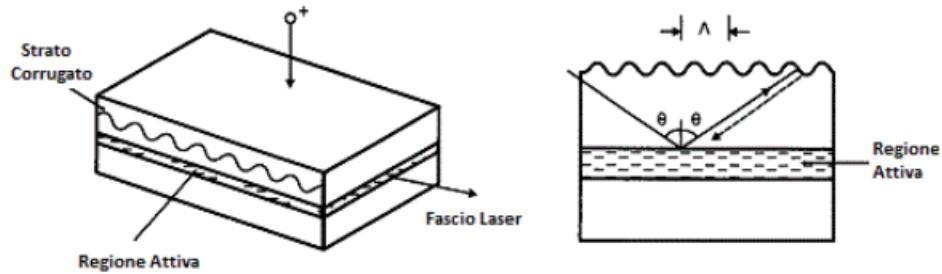


Figura 1.9: Laser DFB

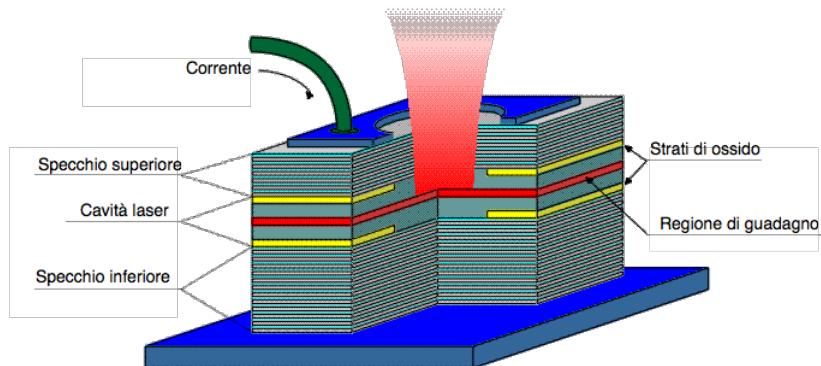


Figura 1.10: Laser VCSEL

### 1.2.2 Laser DFB

DFB è l'acronimo di *Distributed Feedback Laser*. Il principio di funzionamento è simile a quello dei laser Fabry-Perot, ma con la importante differenza che non sono presenti due specchi separati a formare la cavità ottica ma è inserito uno strato corrugato adiacente allo strato attivo.

Questo strato aggiuntivo è fabbricato in maniera tale da riflettere solo una banda stretta di lunghezze d'onda, così da garantire un singolo modo longitudinale.

Per questa caratteristica i laser DFB, sono più stabili dei Fabry-Perot, e più adatti ad applicazioni connesse alle fibre ottiche.

### 1.2.3 Laser VCSEL

VCSEL è l'acronimo di *Vertical Cavity Surface-Emitting Laser*. Questi dispositivi, a differenza delle precedenti due categorie, presentano una cavità

ottica verticale.

La cavità ottica ha una lunghezza molto inferiore alle dimensioni laterali del dispositivo, ciò implica che la luce del fascio laser viene emessa dalla superficie piuttosto che dai bordi laterali.

Poiché la cavità risulta essere più ridotta rispetto alle altre tipologie, a parità di guadagno  $G$  occorre aumentare la riflettività degli specchi per ottenere un guadagno d'anello maggiore di uno. A sfavore vi è una limitazione in potenza di emissione dovuta agli specchi altamente riflettenti.

### 1.3 Classi di sicurezza dei laser

Durante lo sviluppo di un sistema che include l'utilizzo di sorgenti laser è di fondamentale importanza garantire lo svolgimento del lavoro in sicurezza. Infatti, avendo a che fare con sorgenti laser, è importante conoscere i danni che esse possono provocare a cose e persone. In particolare, nei casi più gravi, le sorgenti laser possono procurare danni alla retina dell'occhio e ustioni alla pelle.

Per poter scegliere in modo appropriato il laser da impiegare nel dispositivo è indispensabile conoscere le norme legislative che ne regolano l'utilizzo [23]. A livello europeo vige la normativa *IEC/EN 60825-1* che fissa una classificazione dei fasci laser in funzione della lunghezza d'onda e della potenza ottica emessa.

La suddetta normativa classifica la pericolosità di una sorgente laser suddividendola in 4 classi:

- *Classe 1*: Questa classe include le sorgenti laser che non arrecano danni a cose e/o persone
- *Classe 2*: Questa classe comprende i laser che emettono nell'intervallo di lunghezza d'onda compreso tra  $400nm$  e  $700nm$ . Possono provocare danni alla retina dell'occhio in caso di osservazione prolungata.
- *Classe 3*: Questa classe comprende due sotto-classi.
  - *Classe 3A*: Questa classe include tutte le sorgenti laser che risultano sicure per una visione naturale e non mediante strumenti ottici (es. microscopi)
  - *Classe 3B*: Questa classe include tutte le sorgenti laser che risultano pericolose per una visione naturale. È necessario utilizzare protezioni per gli occhi e disporre delle segnalazioni di avvertimento nella zona in cui si utilizza il laser.

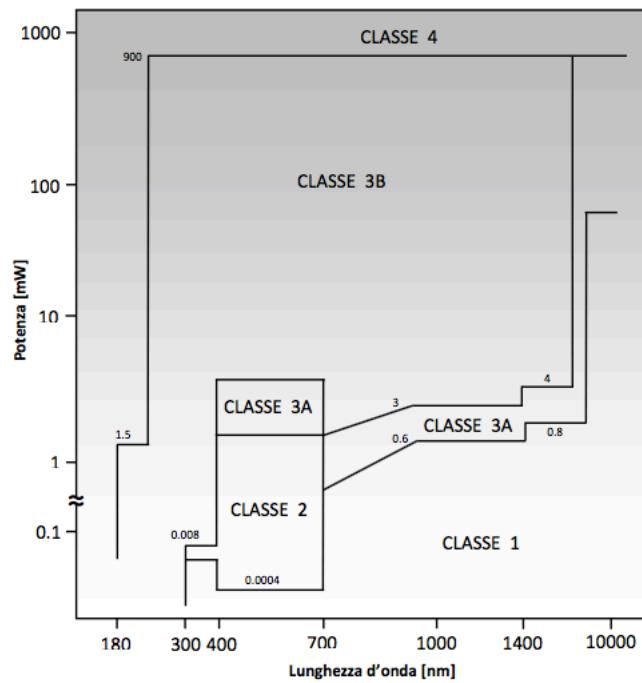


Figura 1.11: Classi di sicurezza dei laser

- **Classe 4:** Questa classe include le sorgenti laser più pericolose. Possono causare lesioni alla pelle e costituiscono pericolo d'incendio.

La classificazione appena descritta è rappresentata in Figura 1.11, insieme alle lunghezze d'onda corrispondenti a ciascuna classe.

## 1.4 Telemetri ottici

La telemetria è un insieme di metodi di osservazione aventi lo scopo di fornire la misura della distanza di un oggetto dall'osservatore.

Lo strumento che effettua la misura è chiamato telemetro: esso rileva la distanza assoluta tra lo strumento stesso e un oggetto remoto, chiamato bersaglio. In commercio esistono diverse tipologie di telemetri che si basano sull'impiego di sorgenti laser.

Le tipologie più diffuse sono tre:

1. A triangolazione
2. A tempo di volo

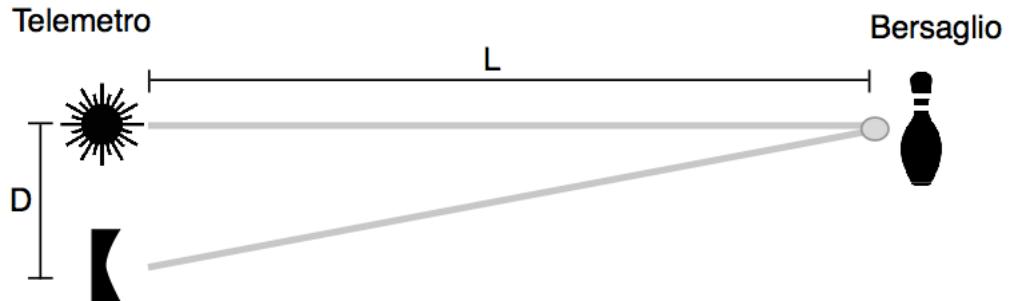


Figura 1.12: Schema di funzionamento di un telemetro a triangolazione

### 3. A onda continua

#### 1.4.1 Telemetri a triangolazione

I telemetri a triangolazione utilizzano una sorgente laser e un'ottica di ricezione posti perpendicolarmente ad una distanza  $D$  tra loro. Inoltre, la sorgente laser è posizionata ad una distanza  $L$  dal bersaglio. Si dice, quindi, che il bersaglio è "triangolato" da due punti a distanza  $D$  su una stessa linea di base. Lo schema è mostrato in Figura 1.12.

Questo tipo di telemetro si basa sul principio della triangolazione, quindi valuta la distanza del bersaglio, conoscendo l'angolo  $\alpha$  con cui il fascio laser viene riflesso sull'ottica di ricezione.

Tramite questo angolo, infatti, è possibile ricavare la seguente relazione:

$$\frac{D}{L} = \tan \alpha \approx \alpha \quad (1.8)$$

Questa approssimazione è valida solo se  $\alpha \ll 1$ .

L'angolo  $\alpha$  si trova valutando la posizione in cui il fascio colpisce il fotorivelatore, secondo la relazione:

$$\alpha = \frac{x}{f_{rec}} \quad (1.9)$$

dove  $x$  è la distanza del punto di incidenza sul fotorivelatore dall'asse ottico della lente di ricezione, mentre  $f_{rec}$  è la lunghezza focale dell'ottica di ricezione.

La misura della distanza si ricava facilmente dalle precedenti relazioni:

$$L = \frac{D}{x} f_{rec} \quad (1.10)$$

Differenziando quest'ultima equazione si trova l'errore di misura assoluto  $\Delta L$  dovuto al minimo spostamento misurabile  $\Delta x$  sul rivelatore:

$$\Delta L = -\frac{D}{x^2} f_{rec} \Delta x \quad (1.11)$$

Ricavando quindi un errore di misura relativo pari a:

$$\frac{\Delta L}{L} = -\frac{\Delta x}{x} = -\frac{\Delta \alpha}{\alpha} \quad (1.12)$$

Il principale svantaggio di questa tecnica è che, con l'aumentare della distanza  $L$ , l'angolo  $\alpha$  diminuisce, e ciò comporta l'aumento dell'incertezza relativa  $-\frac{\Delta \alpha}{\alpha}$ .

Per tale motivo, questa tipologia di telemetro è utilizzata per la misura di brevi distanze ( $0.1 \div 10m$ ).

#### 1.4.2 Telemetri a tempo di volo

I telemetri a tempo di volo utilizzano un laser impulsato. Essi emettono potenza solo per un brevissimo istante di tempo  $\tau$ .

Con un bersaglio posto a distanza  $L$  dalla sorgente laser, il fascio laser percorre un cammino pari a  $2L$  (Andata e Ritorno) in un tempo  $T$  viaggiando a velocità della luce  $c \approx 3 \cdot 10^8 \frac{m}{s}$ , ricavando così la seguente relazione:

$$L = \frac{c}{2} T \quad (1.13)$$

Differenziando la precedente equazione si ottiene:

$$\Delta L = \frac{c}{2} \Delta T \quad (1.14)$$

$$\frac{\Delta L}{L} = \frac{\Delta T}{T} \quad (1.15)$$

Si può osservare come la misura  $\Delta L$  dipenda unicamente dal  $\Delta T$  che si riesce a risolvere. In pratica, la misura viene effettuata utilizzando un contatore che conta quanti tempi di clock  $\Delta T$  sono intercorsi tra l'istante di partenza del raggio laser e il momento in cui ritorna all'ottica di rivelazione.

Il vincolo di questi telemetri è che la durata dell'impulso  $\tau$  deve soddisfare la condizione  $\tau \ll \Delta T$ . Per tale motivo sono utilizzati per misure di distanze medio-lunghe (fino a  $10Km$ ).

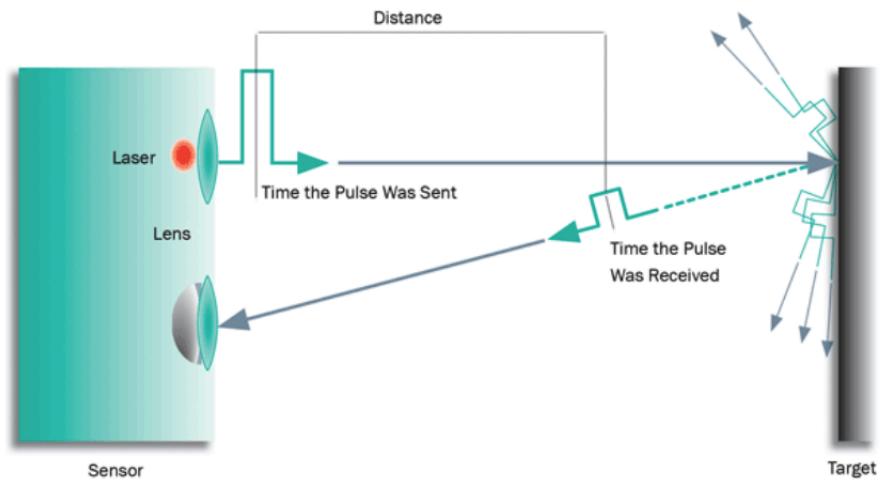


Figura 1.13: Schema di funzionamento di un telemetri a tempo di volo

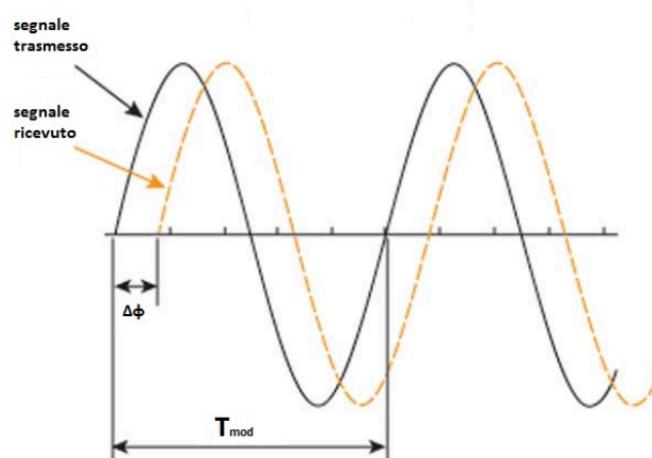


Figura 1.14: Potenza trasmessa e potenza riflessa in un telemetro a onda continua

### 1.4.3 Telemetri a onda continua

I telemetri a onda continua hanno un funzionamento analogo ai telemetri a tempo di volo, ma a differenza di questi ultimi, la potenza ottica viene modulata sinusoidalmente a frequenza  $f_{mod}$  ottenendo il segnale in Figura 1.14.

La potenza trasmessa dal laser vale quindi:

$$P(t) = P_0[1 + m\sin(2\pi f_{mod}t)] \quad (1.16)$$

Nella pratica, la misura non viene effettuata con un contatore elettronico, come per i telemetri a tempo di volo, ma avviene mediante la rilevazione del ritardo di fase  $\Delta\phi$  tra il segnale ricevuto  $P_r$  e il segnale trasmesso  $P_t$ .

Considerando la relazione:

$$\frac{\Delta\phi}{2\pi} = \frac{\Delta t}{T_{mod}} \quad (1.17)$$

dove  $\Delta t = \frac{2L}{c}$  e  $T_{mod} = \frac{1}{f_{mod}}$ , si ottiene così la misura di distanza assoluta:

$$L = \frac{c}{2} \frac{1}{2\pi f_{mod}} \Delta\phi = \frac{\Delta\phi}{S} \quad (1.18)$$

Dall'ultima equazione si può notare come la sensibilità  $S$  della misura migliori all'aumentare della frequenza di modulazione. Tuttavia un aumento eccessivo della frequenza di modulazione comporta una riduzione della massima distanza rilevabile.

Questo tipo di telemetro si utilizza per misure comprese tra  $1 \div 1000m$ , con risoluzioni dell'ordine del millimetro.

## 1.5 Una tecnica alternativa per la misura

Come già discusso nell'Introduzione, lo scopo di questo lavoro di Tesi è realizzare il prototipo di un misuratore di distanza assoluta che sia a basso costo e con ingombro limitato per poter essere più maneggevole nella misura di medio-brevi distanze.

Da quanto esposto nel precedente paragrafo, emerge che i telemetri descritti non sono in grado di raggiungere tali obiettivi. Infatti tali strumenti sono adatti per la misura di distanze dell'ordine della decina di metri. Per quanto riguarda i telemetri a triangolazione, invece, esistono in commercio strumenti in grado di misurare distanze brevi con buona sensibilità. Tuttavia questi dispositivi, oltre ad avere un costo elevato, necessitano di un'ottica specifica per la ricezione della radiazione riflessa e di un bersaglio cooperativo.

Da queste osservazioni si conclude che, per raggiungere l'obiettivo preposto, è necessario utilizzare una tecnica di misura che si differenzia da quelle comuni, ovvero l'**interferometria a retroiniezione**. Questa tecnica verrà ampiamente discussa nel Capitolo 2.



## Capitolo 2

# Interferometria a retroiniezione

*In questo Capitolo verranno trattati i principi di base dell'interferometria, in particolare verrà presentata la tecnica utilizzata dello strumento di misura sviluppato in questo lavoro di Tesi: l'interferometria a retroiniezione. Dopo aver descritto il principio di funzionamento dell'interferometria a retroiniezione, saranno esposti i principali vantaggi, svantaggi e limitazioni di questa tecnica. In conclusione, verranno descritti i campi di applicazione della tecnica a retroiniezione ponendo particolare attenzione al campo d'applicazione dello strumento sviluppato: la misura di distanza assoluta.*

### 2.1 Principi di interferometria convenzionale

L'interferometria convenzionale è una tecnica che si basa sulla sovrapposizione di due o più fasci ottici, emessi dalla stessa sorgente, al fine di ottenere una frequenza di battimento (frequenza risultante dalla sovrapposizione) che contiene informazioni sui differenti cammini percorsi dai due fasci. Questa tecnica è in accordo con la teoria ondulatoria della luce che attribuisce alla propagazione della luce le caratteristiche della propagazione delle onde (eletromagnetiche) elastiche. Essa si basa sul fenomeno dell'interferenza e in particolare si sfrutta il principio secondo cui un'onda risultante dalla combinazione di due onde differenti mantiene proprietà che dipendono dalle onde generatrici.

I campi di applicazione di tale tecnica spaziano dall'astronomia alla metrologia ottica. Essa, in generale, trova applicazioni in ambiti dove l'ambiente di lavoro è critico (ad esempio superfici calde) o l'oggetto da misurare è difficilmente raggiungibile. La configurazione ottica interferometrica classica, mostrata in Figura 2.1, è definita interferometro di *Michelson* [16]. Questa

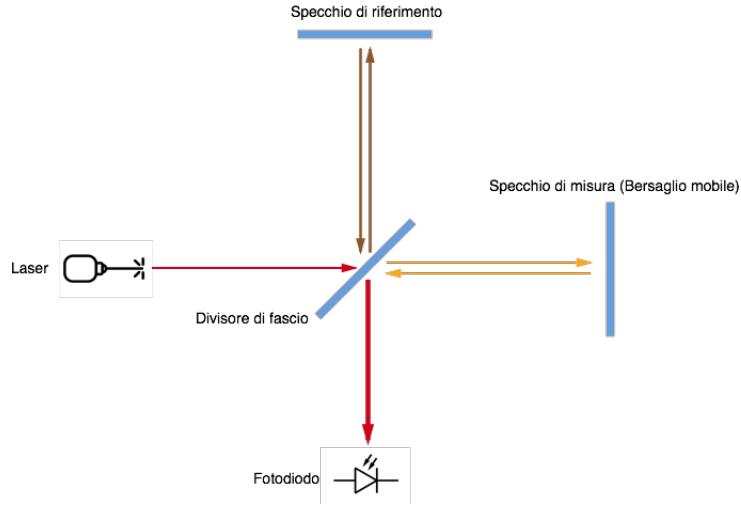


Figura 2.1: Interferometro di Michelson

configurazione è costituita da una sorgente laser, un divisore di fascio (*beam splitter*), due specchi e un fotodiodo.

Il funzionamento dell'interferometro di *Michelson* consiste nel duplicare il fascio ottico emesso dalla sorgente laser, tramite uno specchio semiriflettente (divisore di fascio), in due cammini ottici distinti, di cui uno noto (di riferimento) e uno di misura. Entrambi i fasci, vengono riflessi da due specchi posti all'estremità dei due cammini. Lo specchio posto nel ramo di riferimento è fisso, mentre quello posto sul ramo di misura è mobile. Successivamente i fasci di riferimento e di misura vengono sovrapposti e indirizzati, attraverso il *beam splitter*, verso il fotodiodo. Il fascio risultante è la combinazione di due fasci iso-frequenziali, ma sfasati a causa dei diversi cammini percorsi.

Infine, il fotodiodo produce una corrente, proporzionale all'intensità del fascio laser incidente su di esso, che contiene l'informazione sullo spostamento del bersaglio. Tale corrente fotogenerata è associata alla potenza incidente che è proporzionale alla somma dei campi elettrici dei rispettivi cammini.

Indicando con  $E_m$  il campo elettrico relativo al cammino di misura e con  $E_r$  il campo elettrico relativo al cammino di riferimento, la corrente fotogenerata  $I_{ph}$  segue la relazione:

$$\begin{aligned}
 I_{ph} &= \sigma |E_r + E_m|^2 = \sigma |E_r e^{i\phi_r} + E_m e^{i\phi_m}|^2 \\
 &= \sigma \{E_r^2 + E_m^2 + 2E_r E_m \operatorname{Re}[e^{i(\phi_m - \phi_r)}]\} \\
 &= I_r + I_m + 2\sqrt{I_r I_m} \cos(\phi_m - \phi_r)
 \end{aligned} \tag{2.1}$$

dove  $\sigma$  è il coefficiente di conversione del fotodiodo,  $E_r$  e  $E_m$  sono rappresen-

tati come vettori rotanti di ampiezza  $|E_{m,r}|$  e fase  $\phi_{m,r}$  e la terza componente  $2\sqrt{I_r I_m} \cos(\phi_m - \phi_r)$  costituisce la fase interferometrica.

Per come è costruito l'interferometro di *Michelson*, si avrà che la fase  $\phi_r$  sarà sempre costante, mentre quella relativa a  $\phi_m$  varierà in funzione dello spostamento dell'ostacolo. Quindi ciò che si ottiene sarà una corrente fotogenerata che dipenderà dal coseno della fase  $\phi_m$ .

Indicando con  $k = \frac{2\pi}{\lambda}$  il numero d'onda (numero di oscillazione nell'unità di lunghezza) è possibile esplicitare l'argomento del coseno ottenendo:

$$\phi = ks = \frac{2\pi}{\lambda} s \quad (2.2)$$

dove  $\lambda$  è la lunghezza d'onda della sorgente laser e  $s$  è lo spostamento.

Essendo  $\phi_r = ks_r$  e  $\phi_m = ks_m$ , si può ricavare la variazione di fase totale, ottenendo:

$$\Delta\phi = \phi_m - \phi_r = k(s_m - s_r) = \frac{2\pi}{\lambda}(s_m - s_r) \quad (2.3)$$

dove  $s_m$  e  $s_r$  sono rispettivamente i cammini ottici di misura e di riferimento.

La variazione di fase del segnale interferometrico contiene quindi l'informazione sullo spostamento del bersaglio rispetto al cammino di riferimento. Il segnale interferometrico è quindi periodico per sfasamenti totali pari a  $2\pi$ , che corrispondono ad uno spostamento  $s_m$  pari a  $\frac{\lambda}{2}$ . La misura avviene tramite il semplice conteggio delle frange interferometriche del segnale nel tempo. La risoluzione dello strumento di misura risulta quindi essere pari a  $\frac{\lambda}{2}$ .

### 2.1.1 Svantaggi dell'interferometria convenzionale

L'interferometro di *Michelson* è molto vantaggioso dal punto di vista della semplicità di realizzazione perché vi è un impiego ridotto di componenti, ma presenta tuttavia notevoli svantaggi nella pratica:

- L'allineamento degli specchi, per far convergere i due fasci di luce coerenti nello stesso punto, e il posizionamento del *beam splitter* richiedono accuratezza elevata e sono quindi di difficile realizzazione.
- Richiede l'utilizzo di un bersaglio cooperativo e di una particolare ottica di collimazione. Non realizzabile in caso di misura non invasiva.
- Non permette di discriminare il verso di spostamento del bersaglio con conseguente ambiguità del movimento misurato. Tale ambiguità è causata dalla risposta cosinusoidale, come mostrato in Figura 2.2.
- La misura subisce alterazioni in caso di retroiniezione non voluta di luce esterna all'interno della cavità ottica del laser.

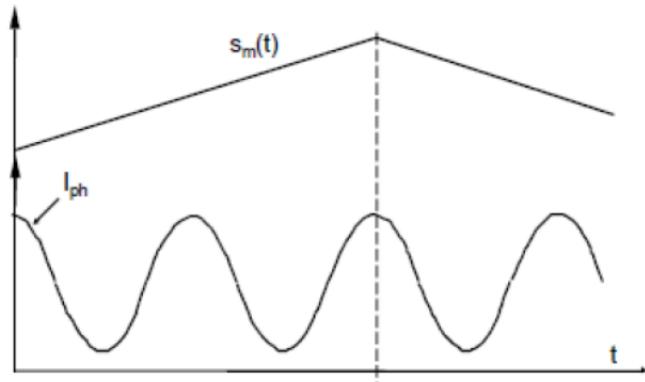


Figura 2.2: Ambiguità sul verso di spostamento

- L'utilizzo di sorgenti laser a gas comporta elevati costi di realizzazione dello strumento.

Tutto ciò ha fatto emergere la necessità di uno strumento che presentasse una semplicità di utilizzo e un costo contenuto.

## 2.2 Interferometria a retroiniezione

È stato esposto, nel paragrafo precedente, come l'interferometria convenzionale presenti notevoli svantaggi. Per tale motivo viene presentata una differente tecnica interferometrica: si tratta dell'interferometria a retroiniezione, chiamata anche interferometria a *self-mixing* [20].

La tecnica interferometrica a retroiniezione rende possibile la realizzazione di uno strumento di misura tramite il semplice utilizzo di una sorgente laser e di una lente per focalizzare il raggio ottico. Questa configurazione interferometrica risolve, quindi, i problemi dell'interferometria convenzionale, presentando una ridotta complessità di realizzazione e un costo contenuto. Per tale motivo il misuratore realizzato in questo lavoro di tesi sfrutta codesta tecnica.

La configurazione base del misuratore a *self-mixing* è formata da un diodo laser, posto a distanza  $s$  dal bersaglio, un'ottica collimatrice e un fotodiodo, che può essere anche quello di monitor integrato nello stesso package del laser. Lo schema base è riportato in Figura 2.3.

La modalità *self-mixing*, a differenza dell'interferometro convenzionale, non utilizza porzioni di fascio ottico come riferimento. Per effettuare la mi-

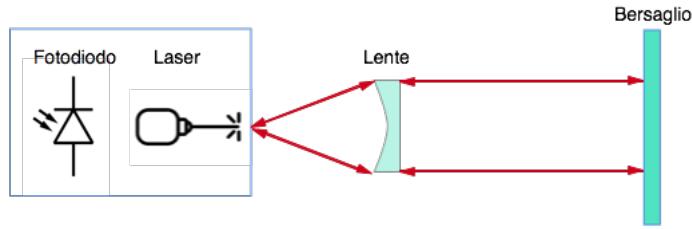


Figura 2.3: Schema di principio di un interferometro a retroiniezione

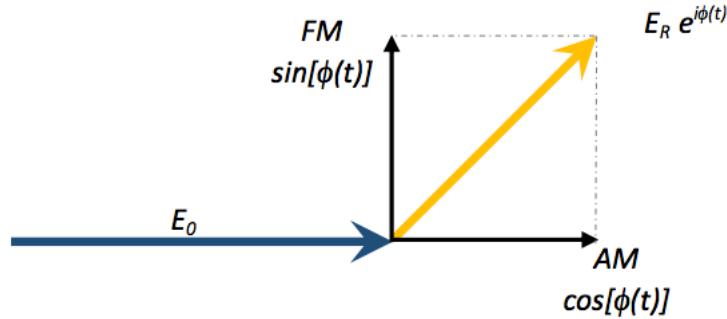


Figura 2.4: Rappresentazione vettoriale dell'interferenza all'interno della cavità laser

sura viene sfruttata una porzione di luce riflessa dalla superficie del bersaglio che, rientrando nella cavità laser attraverso la lente, genera un battimento ottico con l'onda laser già presente. La luce riflessa dal bersaglio arriva con verso opposto al precedente cammino e con una potenza che è una frazione di quella emessa, a causa dell'attenuazione data dal tragitto laser-ostacolo-laser.

È possibile esplicitare la potenza della luce riflessa dal bersaglio con la relazione:

$$P_r = \frac{P_0}{A} \quad (2.4)$$

dove  $P_0$  è la potenza ottica della radiazione emessa e  $A$  è il coefficiente di attenuazione.

All'interno della cavità ottica del laser si verifica un fenomeno di interferenza: il campo elettrico  $E_0$  della radiazione emessa si combina con il campo elettrico  $E_r$  della luce retroiniettata. Come descritto nel paragrafo precedente, anche in questa situazione,  $E_r$  risulta avere uno sfasamento ottico che è pari a:

$$\phi = 2ks \quad (2.5)$$

dove  $k = \frac{2\pi}{\lambda}$  è il numero d'onda e  $s$  è la distanza. Come mostrato in fi-

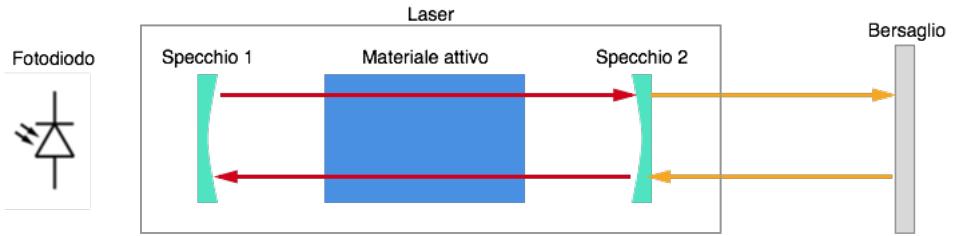


Figura 2.5: Round trip ottico del campo elettrico all'interno della cavità laser

gura 2.4 è possibile scomporre  $E_r$ , operando un'analisi vettoriale, nelle sue componenti in fase ed in ampiezza. Ciò permette, nella cavità ottica, sia la modulazione in frequenza (FM) che la modulazione in ampiezza (AM) del campo elettrico emesso originariamente dalla sorgente  $E_0$ .

La componente che opera la modulazione d'ampiezza è  $E_r \cos[\phi(t)]$ , mentre la componente di modulazione di frequenza è  $E_r \sin[\phi(t)]$ . Le due componenti sono sfasate di  $90^\circ$  e quindi la discriminazione dei due segnali in quadratura permette di ricavare senza ambiguità il verso dello spostamento del bersaglio, al contrario dell'interferometro di *Michelson*.

L'equazione che governa la corrente nel fotodiodo sarà:

$$I_{ph} = I_0(1 + m_{AM}) \cos [(1 + m_{FM})\omega t] \quad (2.6)$$

dove  $m_{AM}$  e  $m_{FM}$  sono le profondità di modulazione.

Se si utilizza un laser a semiconduttore, come nel caso dello strumento presentato in questa Tesi, la componente FM presente nel fotodiodo è impossibile da estrarre in quanto presenta frequenze dell'ordine delle decine di *MHz*. Quindi, l'unica modulazione visibile sulla corrente del fotodiodo è quella AM:

$$I_{ph} = I_0(1 + m_{AM}) \cos [\omega t] \quad (2.7)$$

Il funzionamento di un diodo laser a singolo modo longitudinale soggetto a retroiniezione è descritto dalle equazioni differenziali sviluppate da Lang&Kobayashi [28]; tuttavia, la risoluzione analitica di tali equazioni non è necessaria ai fini di questo lavoro di tesi.

Per questo motivo nel paragrafo successivo è presentata soltanto una soluzione qualitativa delle equazioni di Lang&Kobayashi, con lo scopo di determinare i regimi di retroiniezione della tecnica *self-mixing*.

### 2.2.1 Regimi di retroiniezione tramite risoluzione qualitativa

Un approccio qualitativo per determinare i regimi di retroiniezione è quello di considerare il *round trip* del campo elettrico di Figura 2.5. In figura sono

stati evidenziati i *round trip* all'interno della cavità e tra il bersaglio e la cavità stessa.

Il campo elettrico totale presente all'interno della cavità è dato dalla somma di due campi elettrici: quello dovuto alla riflessione dello specchio interno posto al lato opposto della cavità e quello dovuto alla riflessione del bersaglio. Il campo elettrico risultante è quindi:

$$E' = ER_1R_2e^{2\gamma L}e^{2jkL} + E\alpha e^{2jks} \quad (2.8)$$

dove:

- $R_1$  e  $R_2$  sono rispettivamente le riflettività dello specchio in cavità e del bersaglio
- $\gamma$  è il guadagno netto per unità di lunghezza
- $L$  è la lunghezza della cavità
- $s$  è la distanza tra il secondo specchio ed il bersaglio
- $\alpha$  è il fattore di riflessione (o diffusione) del bersaglio

È possibile, quindi, ricavare facilmente dall'equazione precedente il guadagno d'anello:

$$G_{loop} = \frac{E}{E'} = R_1R_2e^{2\gamma L}e^{2jkL} + \alpha e^{2jks} \quad (2.9)$$

Per far sì che il sistema sia in grado di produrre oscillazioni spontanee che si mantengano nel tempo con ampiezza costante è necessario rispettare il criterio di *Barkhausen*, il quale afferma che il sistema non deve modificare l'ampiezza del segnale e non deve introdurre sfasamento complessivo. Tali condizioni sono riassunte come segue:

$$\begin{cases} |G_{loop}| = 1 \\ \Phi_{loop} = 0 \end{cases} \quad (2.10)$$

Per studiare le soluzioni del sistema, si divide l'analisi del problema in due casi:

1. In assenza di retroazione
2. In presenza di retroazione

Nel primo caso, ovvero con coefficiente  $\alpha$  nullo, si trovano le stesse equazioni di un normale laser in cui il guadagno e le perdite sono uguali, ovvero:

$$\begin{cases} G_{loop} = R_1 R_2 e^{2\gamma L} \\ \Phi_{loop} = 2kL = N2\pi \end{cases} \quad (2.11)$$

con:

$$k = 2\pi n_l \frac{\nu_0}{c} \quad (2.12)$$

$$\nu_0 = N \frac{c}{2n_l L} \quad (2.13)$$

dove  $k$  è il numero d'onda,  $\nu_0$  sono i modi di risonanza e  $n_l$  rappresenta l'indice di rifrazione del mezzo attivo.

Se la frequenza reale si discosta da quella di risonanza propria, la frazione di fase  $2kL$  in eccesso rispetto ad un multiplo di  $2\pi$  può essere scritta come:

$$2kL = 4\pi n_l L \frac{\nu - \nu_0}{c} \quad (2.14)$$

Infatti, in un interferometro con laser Fabry-Perot, la frequenza di risonanza diminuisce all'aumentare della lunghezza  $L$  della cavità. Tale affermazione è resa valida dalla relazione differenziale:

$$\frac{\Delta L}{L} = \frac{\Delta \lambda}{\lambda} = -\frac{\Delta \nu}{\nu} \quad (2.15)$$

Se invece si è in presenza di retroiniezione (secondo caso), ovvero con coefficiente  $\alpha$  non nullo, l'equazione 2.9 diventa:

$$R_1 R_2 e^{2\gamma L} \sin \left( 4\pi n_l L \frac{\nu - \nu_0}{c} \right) + \alpha \sin (2ks) = 0 \quad (2.16)$$

Ipotizziamo il termine di variazione di frequenza  $(\nu - \nu_0)$  abbastanza piccolo da poter approssimare  $\sin(x)$  con  $x$ . Mettendo l'equazione 2.16 a sistema con le equazioni presenti in 2.11 e considerando l'approssimazione e la sostituzione dei parametri, si ottiene la condizione di risonanza:

$$2ks = 4\pi s \frac{\nu}{c} \approx 4\pi s \frac{\nu_0}{c} \quad (2.17)$$

$$(\nu - \nu_0) + \left[ \frac{c}{4\pi n_l L} \alpha \sin \left( \frac{4\pi s \nu_0}{c} \right) \right] = 0 \quad (2.18)$$

Indicando, infine, con  $\nu' = (\nu - \nu_0)$  lo scostamento della frequenza reale rispetto a quella ideale, la modulazione della frequenza vale:

$$\nu' = \frac{c}{4\pi n_l L} \alpha \sin \left( \frac{4\pi s \nu_0}{c} \right) \quad (2.19)$$

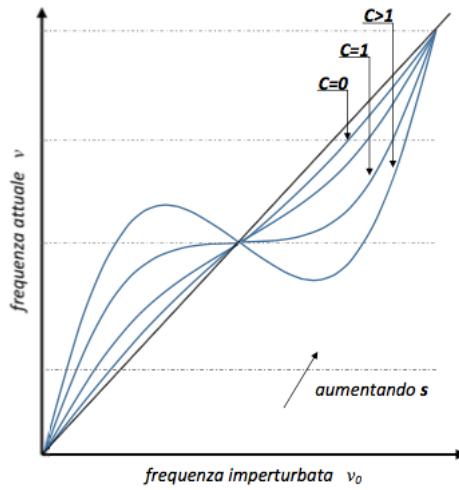


Figura 2.6: Perturbazione della frequenza reale rispetto alla frequenza ideale

In Figura 2.6 è rappresentata la relazione tra la frequenza reale  $\nu$  e la frequenza imperturbata  $\nu_0$  all'aumentare dello spostamento  $s$  del bersaglio. Dalla relazione si ottiene una sinusode sovrapposta alla bisettrice del primo quadrante periodica di  $2\pi$ .

Ogni salto è equivalente a spostamenti  $s$  pari a  $\frac{\lambda}{2}$ , quindi la distanza del bersaglio si può esprimere come un multiplo intero di  $\frac{\lambda}{2}$  sommato allo scarto  $\Delta s$ :

$$s = N \frac{\lambda}{2} + \Delta s \quad (2.20)$$

assumendo  $\Delta s$  minore di  $\frac{\lambda}{2}$ .

Le intersezioni tra le linee tratteggiate e le curve sinusoidali, mostrate in Figura 2.6, indicano i punti di lavoro del sistema: spostando il bersaglio della quantità  $\delta s$ , la frequenza reale assume il valore dato dall'intersezione tra la linea tratteggiata e la curva.

In base all'ampiezza della sinusode si possono distinguere due regimi di funzionamento:

1. *Regime di bassa retroiniezione*
2. *Regime di alta retroiniezione*

Si dice regime di bassa iniezione quando l'ampiezza della sinusode sovrapposta è piccola e vi è un unico punto di intersezione che corrisponde alla frequenza reale  $\nu$ .

Si dice regime di alta iniezione, invece, quando l'ampiezza della sinusoide è grande e vi sono tre o più punti di intersezione rappresentanti la frequenza reale  $\nu$ . Per distinguere la frequenza effettiva, in regime di alta iniezione, è necessario conoscere le condizioni del laser prima dello spostamento  $\Delta s$  [34].

Il limite tra i due regimi sta nel punto centrale del grafico in cui vi è un flesso a tangente orizzontale. Tale condizione può essere espressa matematicamente con il sistema:

$$\begin{cases} \frac{d(y=x+Asin(Bx))}{dx} = 0 \\ Bx = \pi \end{cases} \quad (2.21)$$

dove  $A = -\frac{c}{4\pi n_l L} \alpha$  e  $B = \frac{4\pi s}{c}$ . Dal sistema 2.21 si ricava  $AB = 1$  e sostituendo i parametri fisici si ottiene:

$$\frac{c}{4\pi n_l L} \alpha \frac{4\pi s}{c} = \frac{\alpha s}{n_l L} = 1 \quad (2.22)$$

Arrivati a questo punto si è in grado di definire il fattore  $C$  come indice della retroiniezione del sistema interferometrico a *self-mixing*:

$$C = \frac{\alpha s}{n_l L} \quad (2.23)$$

Nel caso in cui la sorgente laser considerata utilizzi come materiale attivo un semiconduttore, il parametro  $C$  diventa:

$$C = \alpha s \frac{\sqrt{1 + \alpha_{en}^2}}{n_l L} \quad (2.24)$$

dove  $\alpha_{en}$  è il fattore di allargamento di riga che tipicamente assume valori tra 2 e 6 nel caso di laser a stato solido.

Come descritto in precedenza non verrà trattata la risoluzione matematica delle equazioni di Lang&Kobayashi, ma si analizzeranno solo i risultati. Per una trattazione completa si faccia riferimento alla letteratura [28].

Risolvendo le equazioni di Lang&Kobayashi si ottiene che con lo spostamento del bersaglio, non è modulata unicamente la frequenza propria di oscillazione del laser, ma anche la potenza ottica emessa dal laser, che in condizioni di retroazione è pari a:

$$P(\Phi) = P_0(1 + mF(\Phi)) \quad (2.25)$$

dove:

- $P_0$  è la potenza emessa del laser in assenza di retroiniezione

- $m$  rappresenta l'ampiezza del segnale a frange sovrapposto al termine costante (profondità di modulazione)
- $\Phi = 2ks$  è lo sfasamento tra l'onda emessa e quella retroiniettata
- $F(\Phi)$  è una funzione di periodo  $2\pi$  che assume valori di ampiezza compresi tra  $-1$  e  $+1$ :

$$F(\Phi(t)) = F(2ks(t)) = F\left(2\frac{2\pi}{\lambda}s(t)\right) \quad (2.26)$$

L'andamento della funzione  $F(\Phi)$  è influenzato dal fattore  $C$ . La conseguenza è che  $F(\Phi)$  dipende da 3 fattori:

1. distanza del bersaglio
2. riflettività del bersaglio
3. parametri del laser

Si possono, quindi, identificare quattro possibili regimi di funzionamento per un interferometro a *self-mixing* sulla base del parametro  $C$ :

1. *Regime di retroiniezione molto debole* ( $0 < C \leq 0.1$ ): la quantità di potenza ottica che rientra in cavità è molto ridotta a causa di un'elevata riflettività degli specchi o di bassi coefficienti di riflessione della superficie del bersaglio. La funzione  $F(\Phi)$  è un coseno di ampiezza molto ridotta, tipicamente  $10^{-4}$  rispetto al valore in continua.
2. *Regime di retroiniezione debole* ( $0.1 < C < 1$ ): La funzione  $F(\Phi)$  inizia a distorcere e non è più simmetrica. La maggior parte dei casi pratici rientra in questo regime di funzionamento.
3. *Regime di retroiniezione moderata* ( $1 < C \leq 4.6$ ): Per i valori di  $C$  presenti in questo intervallo ci sono tre punti di intersezione in Figura 2.6, che indicano bruschi salti di potenza ottica.
4. *Regime di retroiniezione forte* ( $C > 4.6$ ): Avviene quando la luce retroiniettata in cavità è al massimo pari a quella incidente, nel quale ci sono cinque o più punti di equilibrio.

In Figura 2.7 è mostrata la forma della funzione  $F(\Phi)$  nei 4 possibili regimi di funzionamento. Nella Figura 2.8 è mostrato, invece, il segnale di comando del bersaglio e i rispettivi segnali interferometrici a frange al variare del parametro  $C$ .

Il segnale risultante è periodico in  $\Phi$  e mostra le tipiche frange interferometriche ogni volta che la fase varia di  $2\pi$ . Ciascuna frangia corrisponde ad uno spostamento  $\frac{\lambda}{2}$  del bersaglio [17].

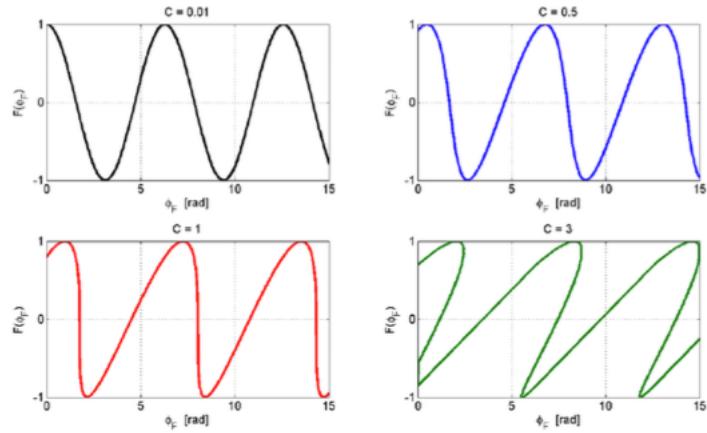


Figura 2.7: Regimi di retroiniezione: funzione  $F(\Phi)$  al variare del parametro  $C$

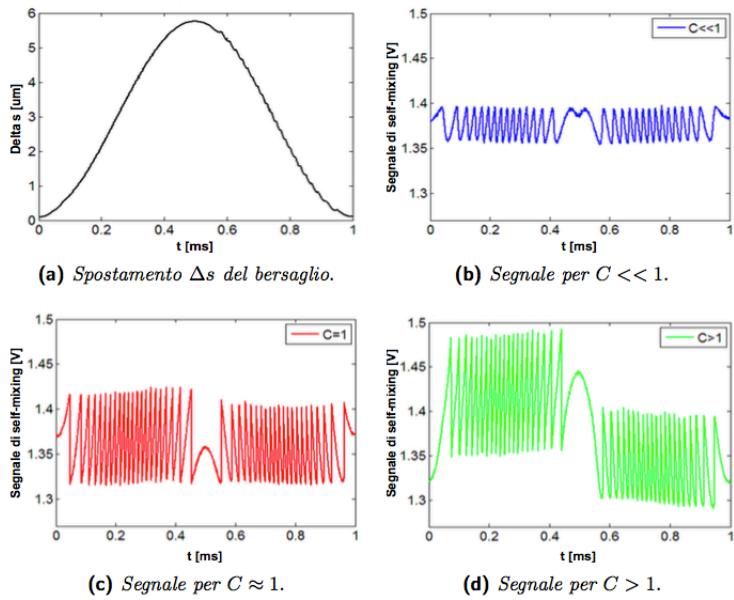


Figura 2.8: Esempi di segnale interferometrico per i differenti valori del parametro  $C$

## 2.2.2 Vantaggi e svantaggi dell'interferometria a retroiniezione

La tecnica *self-mixing* ha diversi vantaggi rispetto agli strumenti basati su altri principi, come i telemetri visti nel Capitolo precedente.

I principali vantaggi sono:

- Semplicità strutturale: Il sistema interferometrico non necessita dell'uso di un canale ottico di riferimento o di divisorì di fascio. L'informazione dello spostamento è ricavata direttamente dalla potenza emessa dal laser.
- Semplicità di rilevamento: Il rilevamento si basa sull'osservazione della distorsione delle frange, più semplice del rilevamento nell'interferometro di *Michelson*. Inoltre, l'allineamento ottico del bersaglio non è particolarmente critico ed è possibile eseguire misure valide su molteplici tipologie di superficie del bersaglio.
- Performante: La semplicità di rilevamento permette di ottenere uno strumento di misura performante.
- Non ambiguità del verso dello spostamento: Il problema dell'ambiguità del verso di spostamento non è presente perché tale informazione è intrinseca nella forma d'onda  $F(\Phi)$ . Per tale motivo non si rende necessario l'utilizzo di un secondo canale di misura interferometrico.
- Elevata sensibilità: Si basa su un sistema ottico coerente che estrae le informazioni dalla sorgente di luce, a differenza della tecnica di triangolazione.

Tuttavia, l'interferometria a retroiniezione presenta anche diversi svantaggi:

- Salti di modo: Sono dovuti a variazioni di corrente o di temperatura nei laser a semiconduttore. Essi influenzano negativamente la misura dell'interferometro, degradandone le prestazioni.
- Dipendenza della misura da variabili esterne: Variabili esterne come temperatura e pompaggio di corrente incidono sulla lunghezza d'onda d'emissione del laser, grandezza dalla quale dipende direttamente la misura, causando una ridotta accuratezza della misura.
- Decadimento della luce coerente: È possibile che la potenza retroiniettata passi a regimi di alta iniezione compromettendo le misure o, in casi limite, danneggiando irreparabilmente il laser.

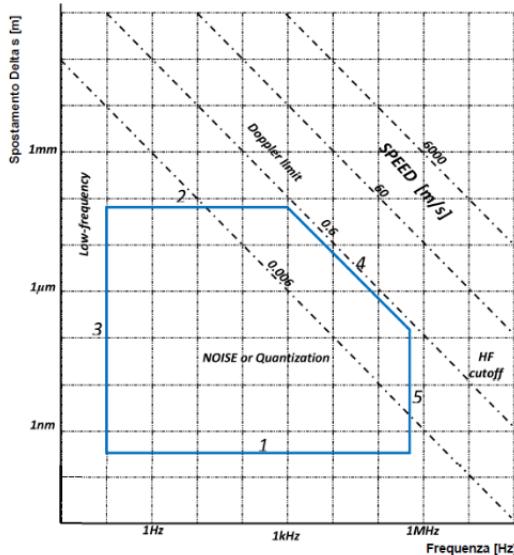


Figura 2.9: Diagramma di Wegel

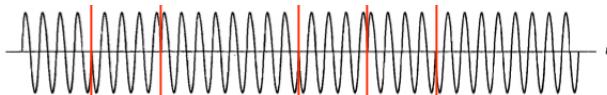


Figura 2.10: Esempio di salti di fase dell'onda nel tempo

### 2.3 Principali limitazioni della tecnica interferometrica

Indipendentemente dalla configurazione di interferometro scelta, le prestazioni della tecnica interreferometrica presentano limiti dovuti a molteplici cause. Le principali sono descritte di seguito.

- Limitazioni nel piano spostamento-frequenza: Il diagramma di Wegel, rappresentato in Figura 2.9, è utilizzato per analizzare la qualità di un interferometro in termini di prestazioni e limiti. Il diagramma presenta 5 segmenti che delimitano un'area chiamata "zona di funzionamento". Maggiore sarà l'area della zona di funzionamento maggiore risulterà il campo d'azione dello strumento di misura.

I 5 segmenti rappresentano, in particolare, 5 limitazioni qui di seguito riportate:

- *Minimo spostamento misurabile*: Il segmento inferiore 1 definisce la sensibilità dello strumento al minimo spostamento misurabile

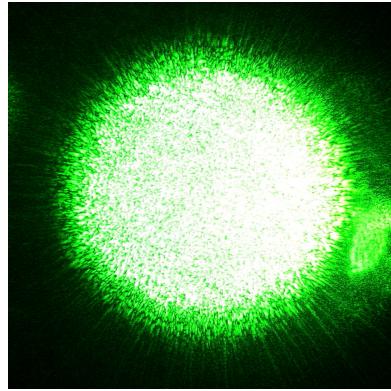


Figura 2.11: Speckle causato da una superficie diffusiva

in base al rumore elettronico di fondo.

- *Massimo spostamento misurabile*: Il segmento superiore 2 impone il limite massimo di spostamento del bersaglio affinché lo strumento funzioni correttamente.
- *Frequenza minima dello spostamento*: Il segmento laterale sinistro 3 indica la minima frequenza di vibrazione in accordo con la minima banda di segnale dell'elettronica di elaborazione.
- *Frequenza massima dello spostamento*: Il segmento laterale destro 5 indica la massima frequenza di vibrazione in accordo con la massima banda di segnale dell'elettronica di elaborazione.
- *Velocità di spostamento massima*: Il segmento obliquo 4 mostra la velocità limite di spostamento massimo, che è proporzionale al prodotto tra la frequenza di oscillazione e la sua ampiezza.
- Coerenza temporale: Un qualunque fascio coerente generato da un laser ha una lunghezza di coerenza limitata, ovvero dopo un certo tempo dalla sua emissione la fase dell'onda cambia ed assume un valore casuale. Dunque il campo emesso da un laser presenta salti di fase a intervalli casuali, come mostrato in Figura 2.10. Indicando con  $T_{coh}$  la coerenza temporale e con  $L_{coh}$  la lunghezza di coerenza, è possibile scrivere:

$$\begin{cases} L_{coh} = cT_{coh} \\ T_{coh} = \frac{1}{\Delta\nu} \end{cases} \quad (2.27)$$

dove  $c$  è la velocità della luce nel vuoto e  $\Delta\nu$  è la larghezza di riga in frequenza.

Dalle precedenti relazioni è possibile ricavare la distanza massima dell'ostacolo affinché non si perda la proprietà di coerenza:

$$s_{max} = |s_m - s_r|_{max} = \frac{c}{\Delta\nu} = L_{coh} \quad (2.28)$$

Se la distanza dell'ostacolo è maggiore di  $L_{coh}$  non si ha segnale interferometrico ma solo rumore dunque non è possibile effettuare misure valide non potendo contare correttamente le frange interferometriche. Questo rumore è definito come rumore di fase. Solitamente il rumore di fase è il contributo dominante nella determinazione del minimo spostamento misurabile.

Tuttavia esiste un altro contributo che limita le prestazioni: il rumore quantico. Tale rumore risulta essere quasi sempre trascurabile perché il rumore di fase è dominante rispetto ad esso.

- Speckle-Pattern: È un fenomeno dovuto alla superficie del bersaglio. In particolare, se un fascio di luce coerente colpisce un bersaglio avente superficie diffondente la luce retrodiffusa avrà una distribuzione di potenza granulare e non omogenea. Il fenomeno è mostrato in Figura 2.11.

La luce retrodiffusa è generata dalla combinazione di due fasci di luce: una porzione di fascio riflessa in modo speculare alla direzione di incidenza e l'altra porzione di fascio diffusa uniformemente nello spazio. Quest'ultima è causata da imperfezioni superficiali del bersaglio.

Ogni imperfezione superficiale, chiamata avvallamento, si comporta come un diffusore di luce secondario debolmente correlato ad un avvallamento adiacente. Il campo risultante, in un punto  $P$  difronte alla superficie dell'ostacolo, è formato dalla sovrapposizione delle onde generate da ciascun avvallamento.

L'effetto di tale sovrapposizione è il risultato dell'interferenza costruttiva e distruttiva di onde aventi fase diversa. Ogni granulo ottico così formato prende il nome di *speckle*. Geometricamente si definisce *speckle* un ellissoide avente variazioni di fase contenute nel 50% e dimensioni trasversali ( $S_t$ ) e longitudinali ( $S_l$ ) pari a:

$$S_t = \lambda \left( \frac{z_0}{D} \right)^2 \quad (2.29)$$

$$S_l = \lambda \left( 2 \frac{z_0}{D} \right)^2 \quad (2.30)$$

dove  $z_0$  è la distanza tra il punto di osservazione e il bersaglio e  $D$  è la dimensione di macchia su di esso.

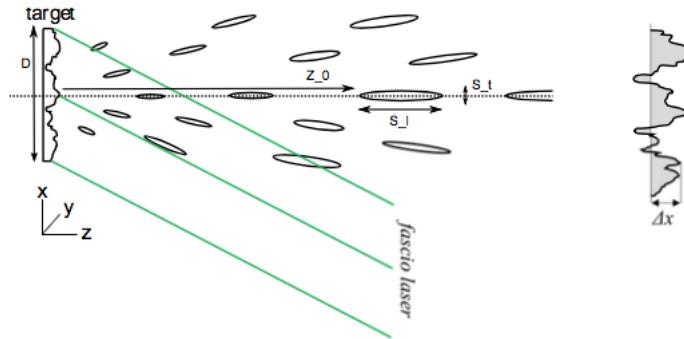


Figura 2.12: Rappresentazione esemplificata del fenomeno dello speckle-pattern

Nelle misure basate sulla tecnica di interferometria a retroiniezione, lo *speckle-pattern* rappresenta il fattore limitante più grande. A causa di tale disturbo, il segnale interferometrico risulterà interrotto ogni volta che il fascio di luce incontrerà uno *speckle*. È possibile limitare questa problematica aumentando la potenza ottica retro-iniettata o adottando il sistema ottico più adatto per l'applicazione in questione.

- Dispersione del mezzo: Il fascio ottico che si propaga fuori cavità subisce delle alterazioni dovute alle caratteristiche di ogni mezzo che esso attraversa, che ne modifica la lunghezza d'onda. I fattori che incidono maggiormente sono temperatura e pressione del mezzo. Per minimizzare il contributo dispersivo di questi fattori vengono impiegati stabilizzatori di temperatura e pressione ambientale.

## 2.4 Applicazioni dell'interferometria a retroiniezione

La tecnica interferometrica a retroiniezione può essere utilizzata in quattro principali campi di applicazione:

- Misura di distanza assoluta: Tecnica per la misurazione di distanza tra lo strumento di misura e un oggetto remoto (bersaglio).
- Velocimetria: Tecnica per la misurazione di velocità a distanza.
- Vibrometria: Tecnica per la misurazione delle vibrazioni. Il vibrometro laser è utile quando non è possibile installare un accelerometro sull'oggetto da misurare. Si basa sul principio di aggancio di fase [30].

- Misura di spostamenti: Tecnica per la misurazione dello spostamento di un bersaglio. Si basa sul conteggio delle frange interferometriche di periodicità  $\frac{\lambda}{2}$  al variare della distanza dal bersaglio.

La misura di distanza assoluta, oggetto del seguente lavoro di Tesi, verrà discussa ampiamente nel paragrafo successivo.

#### 2.4.1 Misura di distanza assoluta

Come descritto in precedenza, un segnale interferometrico è caratterizzato dalla seguente relazione:

$$\phi = 2ks = 2\frac{2\pi}{\lambda}s \quad (2.31)$$

che rappresenta lo sfasamento del campo elettrico retroiniettato rispetto al campo elettrico emesso.

Da tale relazione si nota che per generare un segnale interferometrico è necessario uno spostamento  $s$  del bersaglio o una variazione della lunghezza d'onda  $\lambda$  del fascio emesso dal laser. Considerando come campo di applicazione la misura di distanza assoluta è necessario agire solo sulla lunghezza d'onda  $\lambda$  e non sullo spostamento  $s$  dato che il bersaglio è immobile.

Un metodo per variare la lunghezza d'onda  $\lambda$  è quello di modulare la corrente di pompa della sorgente laser:

$$\lambda_{new} = \lambda_{old} + \chi\Delta I \quad (2.32)$$

dove  $\chi$  è la variazione di lunghezza d'onda rispetto alla variazione di corrente. La variazione di lunghezza d'onda  $\chi$  è pari a  $\chi = \frac{\Delta\lambda}{\Delta I}$  ed è espressa in  $[\frac{\text{pm}}{\text{mA}}]$ . Tale parametro varia molto da dispositivo a dispositivo. Esso risulta, inoltre, fortemente non lineare in alcune zone di funzionamento della sorgente laser.

Differenziando l'equazione 2.32, rispetto alla lunghezza d'onda  $\lambda$ , è possibile ottenere la variazione di fase del segnale interferometrico:

$$\frac{d\phi}{d\lambda} = -2\frac{2\pi}{\lambda^2}s \quad (2.33)$$

da cui si può ricavare la misura di distanza assoluta  $s$ :

$$s = -\frac{d\phi}{d\lambda} \frac{\lambda^2}{4\pi} \quad (2.34)$$

Infine, la misura di distanza assoluta  $s$  può essere compiuta in due modi:

1. Conteggio del numero di frange: Questo metodo consiste nel contare il numero di frange visibili nel segnale interferometrico. Il numero di

frange ad una distanza fissata  $s$  dipende dalla variazione della lunghezza d'onda. Nota la variazione della lunghezza d'onda, più precisamente il periodo in cui viene fornita la variazione di corrente che genera la variazione di lunghezza d'onda, è possibile misurare la distanza assoluta del bersaglio con la seguente equazione:

$$s = \frac{N_F \lambda^2}{\Delta\lambda} \frac{1}{2} \quad (2.35)$$

dove  $N_F$  rappresenta il numero di frange pari a:

$$N_F = \frac{\Delta\phi}{2\pi} \quad (2.36)$$

Questo metodo ha una scarsa accuratezza poiché la risoluzione massima ottenibile è data dalla singola frangia.

2. Misura del tempo di frangia: Questo metodo consiste nella misura del periodo di frangia. Definendo con  $t_{frangia}$  la lunghezza del periodo di frangia, è possibile scrivere la seguente proporzione:

$$\Delta\lambda_{2\pi} : \Delta\lambda = t_{frangia} : \Delta t \quad (2.37)$$

dove  $\Delta\lambda_{2\pi}$  è la variazione di lunghezza d'onda che produce una variazione di fase interferometrica  $\Delta\phi$  pari a  $2\pi$ . Dalla precedente relazione è possibile ricavare la variazione di lunghezza d'onda  $\Delta\lambda_{2\pi}$ :

$$\Delta\lambda_{2\pi} = \frac{t_{frangia}}{\Delta t} \Delta\lambda = \frac{\Delta\lambda}{\Delta I} \frac{\Delta I}{\Delta t} t_{frangia} \quad (2.38)$$

Per ricavare la distanza assoluta  $s$  è necessario fare riferimento alla relazione:

$$\left( 2\frac{2\pi}{\lambda} - 2\frac{2\pi}{\lambda + \Delta\lambda_{2\pi}} \right) s = 2\pi \quad (2.39)$$

che eguaglia la differenza di fase causata da una variazione di lunghezza d'onda  $\Delta\lambda_{2\pi}$  a  $2\pi$ .

Infine, per ricavare la distanza  $s$ , è necessario approssimare  $(\lambda + \Delta\lambda_{2\pi}) \approx \lambda$  poiché  $\lambda \gg \Delta\lambda_{2\pi}$ :

$$s \approx \frac{\lambda^2}{2\frac{\Delta\lambda}{\Delta I} \frac{\Delta I}{\Delta t} t_{frangia}} = \frac{\lambda^2}{2\frac{\Delta\lambda}{\Delta I} \frac{\Delta I}{\Delta t}} f_{frangia} \quad (2.40)$$

dove  $f_{frangia} = \frac{1}{t_{frangia}}$  è il tono fondamentale del segnale interferometrico.

Lo strumento di misura descritto in questo elaborato, deve essere in grado di effettuare la misura anche con ostacolo in movimento. Per

tal motivo occorre scegliere un'opportuna forma d'onda per realizzare la modulazione in corrente.

Come già detto in precedenza, uno spostamento dell'ostacolo provoca una variazione di fase del segnale interferometrico. Richiamando che la variazione di fase causata dalla modulazione equivale a:

$$\frac{d\phi_{mod}}{dt} = 2\pi f_{mod} \quad (2.41)$$

e derivando la relazione 2.31 rispetto allo spazio:

$$\frac{d\phi_s}{ds} = 2\frac{2\pi}{\lambda} \quad (2.42)$$

è possibile ottenere:

$$\frac{d\phi_s}{ds} = 2\pi f_s \quad (2.43)$$

dove  $f_s$  è la frequenza media delle frange prodotte a causa dello spostamento.

Da questo si deduce che: se il bersaglio è in movimento la variazione di fase complessiva del segnale interferometrico è data dalla somma di due componenti:  $\phi_{mod}$  e  $\phi_s$ . Per compensare il contributo indesiderato  $\phi_s$  è necessario realizzare una modulazione di corrente che vada a incrementare e decrementare la lunghezza d'onda della sorgente laser. Per fare ciò, si utilizza un segnale a forma triangolare con ampiezza opportuna.

Un accorgimento atto a compensare e minimizzare gli sfasamenti consiste nell'esecuzione, durante il processo di misura, di una media tra il periodo di frangia valutato nella fase ascendente e il periodo di frangia valutato nella fase discendente della triangolare di modulazione allo scopo di aumentare la sensibilità dello strumento rispetto allo spostamento.

Da questa osservazione si ottiene la relazione:

$$s \propto \frac{f_{rise} + f_{fall}}{2} = \frac{\left| \frac{d\phi_{rise}}{dt} + \frac{d\phi_{fall}}{dt} \right|}{2} \quad (2.44)$$

dove  $f_{rise}$  e  $f_{fall}$  sono rispettivamente il tono fondamentale del segnale interferometrico nella rampa di salita e discesa dell'onda triangolare.

Sapendo che la fase complessiva del segnale interferometrico è formata dai due contributi discussi in precedenza è possibile sostituire nella relazione 2.44 la fase totale  $\phi_{tot} = \phi_{mod} + \phi_s$ :

$$s \propto \frac{\left| \frac{d\phi_{mod}}{dt} + \frac{d\phi_s}{dt} \right| + \left| \frac{d\phi_{mod}}{dt} - \frac{d\phi_s}{dt} \right|}{2} = \frac{d\phi_{mod}}{dt} \quad (2.45)$$

Infine, in questo modo è possibile misurare la distanza assoluta eliminando così il contributo indesiderato  $\phi_s$  causato da un eventuale spostamento del bersaglio.

Esprimendo le grandezze  $f_{rise}$  e  $f_{fall}$  attraverso le seguenti relazioni:

$$f_{rise} = \left| \frac{1}{2\pi} \frac{\delta\phi}{\delta t} \right| = \left| -\frac{2s}{\lambda^2} \frac{\delta\lambda}{\delta t} + \frac{2}{\lambda} \frac{\delta s}{\delta t} \right| = \left| -\frac{2sk}{\lambda^2} \frac{\Delta I}{\Delta T} + \frac{2}{\lambda} v \right| \quad (2.46)$$

$$f_{fall} = \left| \frac{1}{2\pi} \frac{\delta\phi}{\delta t} \right| = \left| \frac{2s}{\lambda^2} \frac{\delta\lambda}{\delta t} + \frac{2}{\lambda} \frac{\delta s}{\delta t} \right| = \left| \frac{2sk}{\lambda^2} \frac{\Delta I}{\Delta T} + \frac{2}{\lambda} v \right| \quad (2.47)$$

dove  $v$  è la velocità del bersaglio e  $T$  è il semiperiodo della triangolare di modulazione, è possibile definire l'equazione della distanza assoluta  $s$  in maniera più precisa:

$$s = \frac{f_{rise} + f_{fall}}{2} \left[ \frac{\lambda^2}{2 \left( \frac{\Delta I}{\Delta T} \frac{\Delta \lambda}{\Delta I} \right)} \right] \quad (2.48)$$

Questa tecnica risulta più accurata della precedente perché è limitata solamente dalla precisione con la quale è possibile misurare il periodo e dalla capacità della sorgente di generare un segnale interferometrico ripetibile.



## Capitolo 3

# Architettura Hardware dello strumento

*In questo Capitolo verrà descritta l'architettura hardware dello strumento realizzato. Prima verrà analizzata brevemente la parte analogica dello strumento. Successivamente verrà descritta la parte di conversione analogico-digitale e digitale-analogico soffermandoci prima sulle architetture dei convertitori presenti in letteratura e poi su quelle presenti nello strumento. Infine verrà presentata la parte digitale motivando la scelta dell'uso di una scheda di prototipazione contenente FPGA e microncontrollore.*

### 3.1 Struttura complessiva dello strumento

Lo strumento di misura progettato in questo elaborato è un misuratore *contactless* di distanza. Lo scopo è realizzare un sistema *embedded* in grado di gestire in modo autonomo tutta la misura, dal controllo della sorgente laser all'acquisizione ed elaborazione del segnale interferometrico.

Per sistema *embedded*, si intende un sistema di elaborazione progettato appositamente per una specifica applicazione (*special purpose*). Come già introdotto, l'applicazione in questione è la misura di distanza.

Il principio fisico per il quale è possibile effettuare una misura è stato descritto nel Capitolo 2.

La Figura 3.1 mostra il risultato finale del misuratore progettato e realizzato per questo lavoro di Tesi. Dalla figura si possono distinguere le tre parti principali che compongono lo strumento:

- Parte analogica: La parte analogica è costituita da una sezione ottica e una sezione analogica. La sezione ottica è composta dal package della sorgente laser, che comprende laser, fotodiodo e lente collimatrice. La

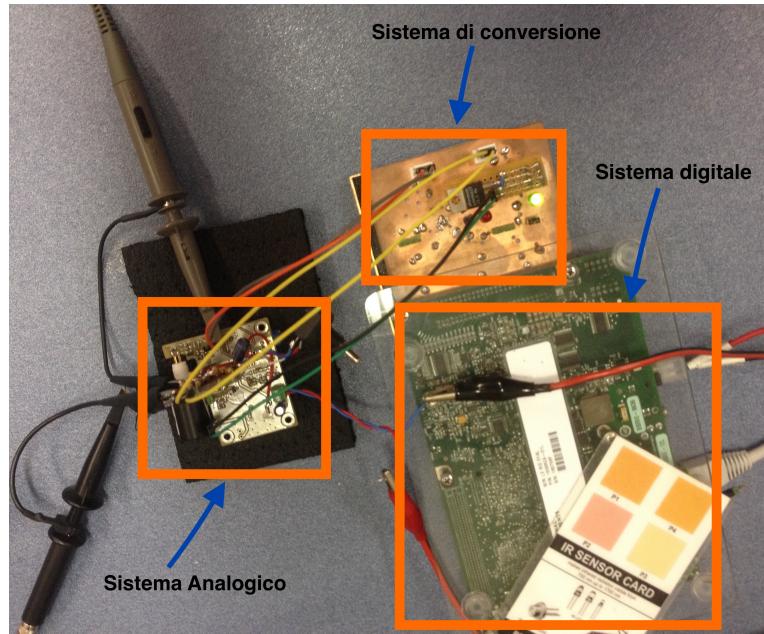


Figura 3.1: Risultato finale del misuratore realizzato

sezione analogica comprende il circuito di alimentazione del laser e gli stadi di condizionamento e amplificazione del segnale interferometrico generato dal fotodiodo.

- Parte di conversione: La parte di conversione è formata da due sezioni: la sezione di conversione analogico-digitale (ADC) e la sezione di conversione digitale-analogico (DAC). La prima sezione provvede ad effettuare la conversione analogico-digitale del segnale interferometrico in uscita dalla parte analogica. Mentre, la seconda sezione effettua la conversione digitale-analogica dal segnale di modulazione del laser in uscita dalla parte digitale.
- Parte digitale: La parte digitale è costituita da un dispositivo *embedded* di prototipazione composto da un microprocessore e un FPGA. I compiti principali della parte digitale sono la generazione del segnale di modulazione del laser, l'elaborazione del segnale interferometrico per l'estrazione della frequenza di frangia e il calcolo della misura di distanza. Un computer è collegato con la parte digitale tramite connessione *Ethernet*, attraverso questo collegamento è possibile la comunicazione con gli elementi programmabili del sistema.

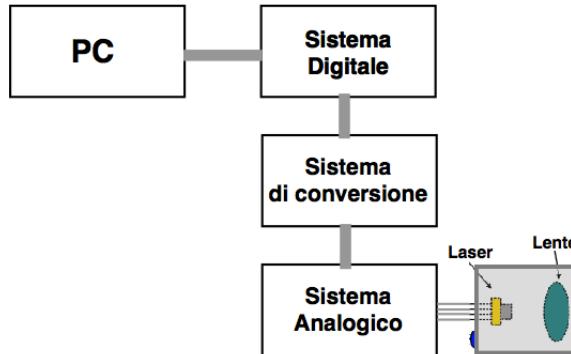


Figura 3.2: Schema a blocchi dell'architettura complessiva del misuratore

L'architettura complessiva del sistema progettato è schematizzata in Figura 3.2. La Figura mostra lo schema concettuale dello strumento, evidenziando le singoli parti e le connessioni fra quest'ultime.

Nei paragrafi successivi verrà fornita un'analisi dettagliata delle singoli parti dello strumento.

## 3.2 Parte analogica

La parte analogica è formata da due sotto-sistemi:

1. Sistema ottico e sorgente laser
2. Circuito di interfacciamento con il modulo laser

### 3.2.1 Sistema ottico e sorgente laser

In lavori precedenti sono state provate diverse sorgenti [8] [41]. Tra queste è stato scelto il modulo laser *WLSD-1550-020m-1-PD*.

Si tratta di una sorgente laser di tipo DFB (*Distributed FeedBack*) prodotta dalla *Wavespectrum*, la cui struttura è mostrata in Figura 3.3. Essa emette ad una lunghezza d'onda di 1550nm.

La sorgente è allineata alla lente per la collimazione, questi due componenti alloggiano in un supporto di alluminio. La lente, codice prodotto *C230TMD-C* realizzata da *Thorlabs*, ha la funzione di raccogliere tutto il fascio laser della sorgente focalizzandolo a piacere. È possibile infatti regolarne la posizione avvitandola o svitandola.

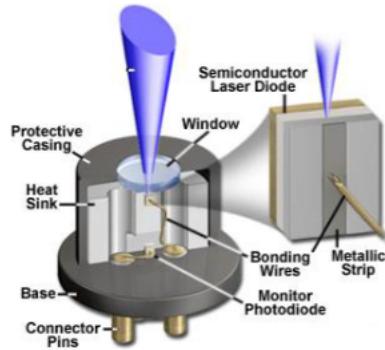


Figura 3.3: Struttura del dispositivo WLSD-1550-020m-1-PD

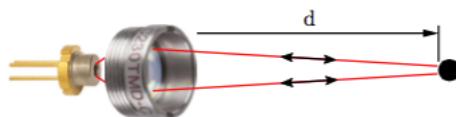


Figura 3.4: Sistema ottico con laser e lente

Di seguito vengono riassunti i punti chiave che hanno portato alla scelta di tale sorgente laser:

- Possiede un buon rapporto segnale/rumore
- È facile modularne la lunghezza d'onda
- Essendo un laser DFB riflette solo una banda ristretta di lunghezze d'onda
- Ha un costo accessibile
- Appartiene alla classe di sicurezza 1
- Ha una debole riflettività dello specchio permettendo così una forte sensibilità alla retroiniezione

Tale sorgente però, emettendo nel non visibile, rende complicati i processi di allineamento. In figura 3.4 è mostrato il sistema ottico senza supporto. Il bersaglio è schematizzato con un pallino nero e  $d$  è la distanza misurata.

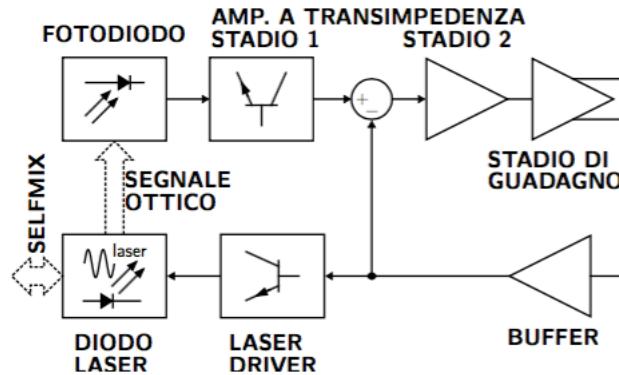


Figura 3.5: Schema a blocchi del circuito di interfacciamento

### 3.2.2 Circuito di interfacciamento

Come ampiamente descritto nel Capitolo precedente, è necessario generare una corrente di modulazione sovrapposta a quella di polarizzazione per il diodo laser. Per tale motivo, questo circuito si occupa dell'elaborazione dei segnali analogici, rispettivamente della corrente di pilotaggio del laser e della corrente di uscita dal fotodiodo.

Lo schema logico del circuito è mostrato in Figura 3.5.

Il segnale di modulazione proveniente dalla parte di conversione attraversa un circuito di filtraggio e viene convertito in corrente (corrente di pilotaggio del laser). Il segnale di corrente dovuto all'effetto interferometrico è minore di  $2\mu A$ , molto più ampio invece è quello associato alla modulazione (circa  $1mA$ ). Per poter estrarre solamente il segnale interferometrico si effettua quindi una differenza del segnale totale rilevato con quello di modulazione del laser. Infine, il risultato della differenza è ulteriormente amplificato e reso disponibile alla parte di conversione in formato differenziale.

Una descrizione più dettagliata della parte analogica e dell'attività di progetto del sistema è trattata nel lavoro di tesi sviluppato dal laureando Samuele Disegna con cui abbiamo collaborato [15].

## 3.3 Parte di Conversione

La scheda utilizzata per la parte di conversione del misuratore è la *SCO Board*, progettata e prodotta nel laboratorio di "Misure Ottiche ed Elettroniche - MOLES" del Politecnico di Milano. Anch'essa è stata progettata e realizzata da Samuele Disegna nel contesto del suo lavoro di tesi.

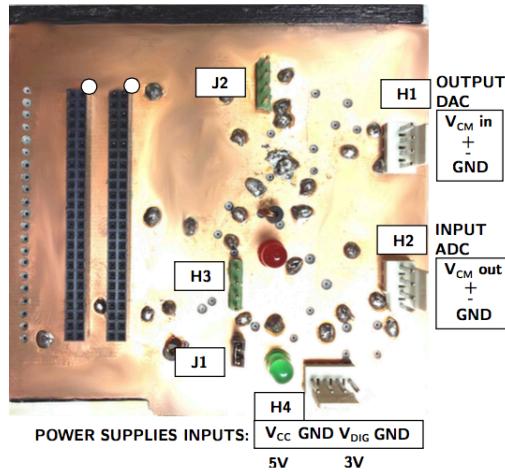


Figura 3.6: Foto della scheda di conversione *SCO Board* e identificazione delle connessioni

Essa possiede due interfacce di conversione: un’interfaccia Analogico-Digitale (AD), che riceve in ingresso il segnale interferometrico in arrivo dalla sorgente laser, e un’interfaccia Digitale-Analogica (DA), che converte il segnale digitale di modulazione in arrivo dalla scheda di prototipazione. Le conversioni AD e DA vengono svolte in parallelo.

Le specifiche fornite per la frequenza massima di campionamento dei segnali sono  $50MS/s$  con 12 bit di quantizzazione, essi garantiscono di avere una buona dinamica di misura della distanza e una buona dinamica di ampiezza del segnale misurato con rumore di quantizzazione trascurabile. In Figura 3.6 e 3.7 sono mostrate rispettivamente la *SCO Board* e il relativo schema a blocchi del circuito.

I due amplificatori operazionali completamente differenziali si occupano della gestione dei segnali analogici. Uno pilota l’uscita differenziale corrispondente al DAC e porta la dinamica di uscita a  $1V_{pp}$ , l’altro prepara il segnale di ingresso differenziale per il campionamento. I filtri passa basso svolgono la funzione di antialiasing per l’ADC e di ricostruzione del segnale del DAC.

I convertitori presenti sulla *SCO Board* e le loro caratteristiche fondamentali verranno descritte in dettaglio nei paragrafi successivi.

Invece, una descrizione più dettagliata del circuito è fornita in letteratura [15].

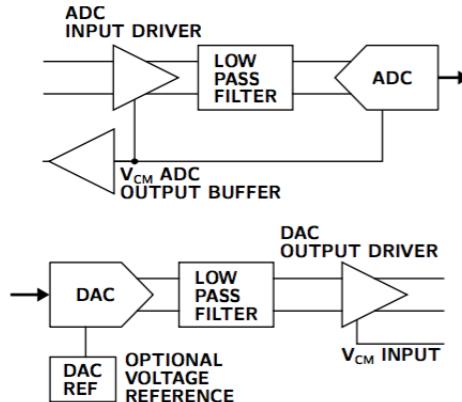


Figura 3.7: Schema a blocchi del sistema di conversione A/D - D/A, SCO Board

### 3.3.1 Convertitori

I convertitori Digitale-Analogico (DAC) e Analogico-Digitale (ADC) fanno parte della famiglia dei convertitori dati (*Data Converters*).

Le caratteristiche fondamentali di un convertitore dati sono:

- Risoluzione: Descrive il numero di valori (livelli di quantizzazione) che il convertitore riesce a distinguere. L'unità di misura è il numero di bit  $n$ . I livelli di quantizzazione  $N$  sono definiti come:

$$N = 2^n \quad (3.1)$$

- Dinamica: Descrive la massima ampiezza del segnale analogico. L'unità di misura è il volt [V].
- Tempo di conversione: Indica il tempo impiegato dal convertitore per eseguire la conversione. I tempi impiegati per la conversione differiscono notevolmente a seconda della tipologia di convertitore utilizzato, anche se, in generale, la conversione DA è più veloce dell'operazione inversa. Di solito è più conveniente specificare il numero di campioni che possono essere convertiti in un secondo, piuttosto che il tempo di conversione. Tale grandezza è chiamata frequenza di campionamento (o *sampling rate*) e si esprime in Hertz [Hz] o campioni per secondo [Sa/s].

Come descritto nel paragrafo precedente, le specifiche fornite per la frequenza massima di campionamento dei segnali sono 50MS/s con 12 bit di risoluzione.

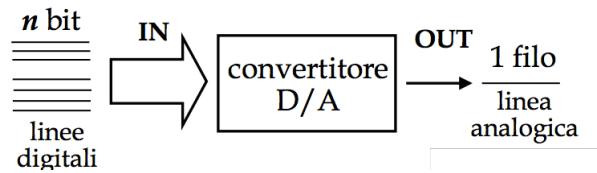


Figura 3.8: Schema di funzionamento di un convertitore Digitale-Analogico (DAC)

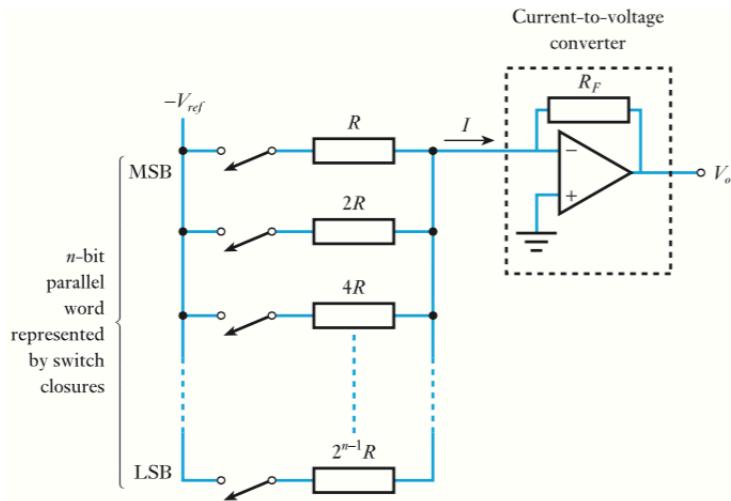


Figura 3.9: Struttura circuitale di un DAC a resistor pesati

Nei paragrafi successivi verrà trattato lo stato dell'arte dei convertitori dati e verranno descritte le architetture dei convertitori presenti sulla scheda di conversione utilizzata.

### Convertitore DAC

Il processo di ricostruzione di un segnale analogico consiste nel prelevare valori digitali e convertirli nei loro equivalenti analogici. Questa operazione è svolta dal convertitore Digitale-Analogico (DAC) [40].

**Architetture DAC** Esistono due architetture comuni di DAC:

1. A resistor pesati (o *Binary-weighted resistor*)
2. A scala R-2R (o *R-2R resistor chain*)

**DAC a resistori pesati** La Figura 3.9 illustra lo schema circuitale di un generico DAC a resistori pesati a  $n$  bit. Il circuito, mostrato in figura, è costituito da  $n$  interruttori pilotati dagli  $n$  bit della parola in ingresso,  $n$  resistori pesati su un valore di riferimento  $R$ , e un convertitore corrente-tensione invertente.

Ogni ingresso controlla un interruttore (*switch*) che collega un resistore a un riferimento di tensione. Quando il generico bit dell'ingresso è a 1 il corrispondente interruttore viene collegato al riferimento di tensione  $-V_{ref}$ , mentre quando il bit è a 0 l'interruttore viene collegato a massa (0V).

Data la corrente passante nell'  $i$ -esimo interruttore controllato dal bit  $b_i$  della parola:

$$I_i = -\frac{V_{ref}}{2^{(n-1)-i}R}b_i = -\frac{V_{ref}}{2^{(n-1)}R}2^i b_i \quad (3.2)$$

si ricava la corrente complessiva della rete di resistori pesati:

$$I = \sum_{k=1}^{n-1} I_k = -\frac{V_{ref}}{2^{(n-1)}R} \sum_{k=1}^{n-1} 2^k b_k \quad (3.3)$$

Infine, per effetto del convertitore corrente-tensione si ha:

$$V_0 = -R_f I \quad (3.4)$$

dove  $R_f$  è il resistore di feedback. L'equazione definisce la relazione ingresso-uscita del DAC.

I vantaggi di questa configurazione sono la semplicità di realizzazione del circuito e l'utilizzo di un ristretto numero di resistenze ( $n$  bit,  $n$  resistenze). Per tale motivo le risoluzioni tipiche per questo metodo sono inferiori ai 10 bit.

Lo svantaggio maggiore è la difficoltà nel realizzare resistenze con valori differenti e perfettamente calibrati, in modo tale che i loro rapporti tra esse siano precisi. Per tale motivo questa configurazione è tuttavia poco usata nella pratica.

**DAC a scala R-2R** Il DAC a resistenze pesate visto nella paragrafo precedente presenta problemi di realizzazione e di funzionamento, dovuti sostanzialmente all'uso di resistori molto differenti fra di loro. Questi problemi possono essere risolti usando un'altra architettura circuitale: il DAC a scala R-2R.

La figura 3.10 illustra lo schema circuitale di un generico DAC a scala R-2R a  $n$  bit. Il circuito, mostrato in figura, è costituito da  $n$  interruttori pilotati dagli  $n$  bit della parola in ingresso,  $n$  resistori e un convertitore corrente-tensione invertente.

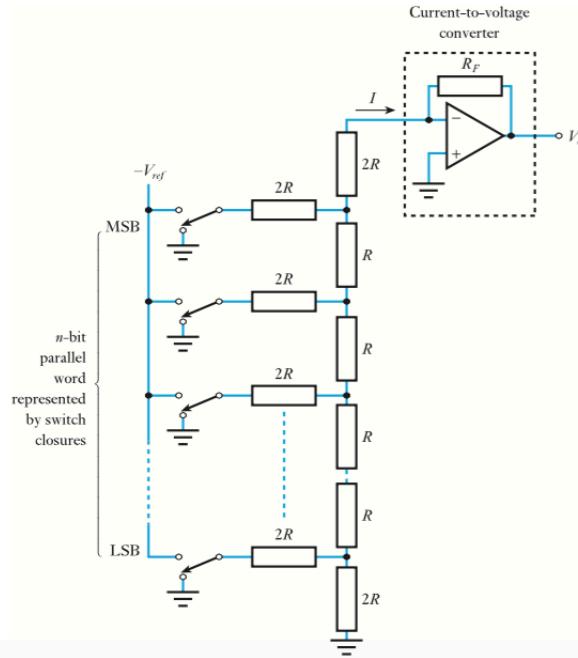


Figura 3.10: Struttura circuitale di un DAC a scala R-2R

La differenza in questo caso è che tutti i resistori collegati agli interruttori hanno lo stesso valore  $2R$ . Inoltre, l'estremità di ogni resistore è collegata ad una catena di resistenze che va dall'ingresso invertente del convertitore corrente-tensione a massa.

Il circuito è realizzato in modo tale che le correnti che fluiscono attraverso i resistori collegati agli interruttori vedano una resistenza  $2R$  guardando in entrambe le direzioni lungo la catena delle resistenze. Pertanto, metà della corrente andrà in ciascuna direzione.

Allo stesso modo, le correnti che scorrono lungo la catena vedono resistenze uguali in entrambe le direzioni e ad ogni nodo vengono nuovamente divise.

Riassumendo, ciascun interruttore contribuisce con metà della corrente fornita dall'interruttore sopra, e tale corrente viene ripetutamente dimezzata da ogni nodo incontrato nel percorso verso l'amplificatore operazionale.

Pertanto, le correnti generate dagli interruttori sono pesate, come nel metodo precedente, ma senza l'uso di un ampio intervallo di valori di resistenza. Dunque questa architettura presenta il vantaggio di utilizzare solo due valori resistivi risultando più facile da realizzare.

I tempi di conversione dei metodi precedentemente descritti sono simili.

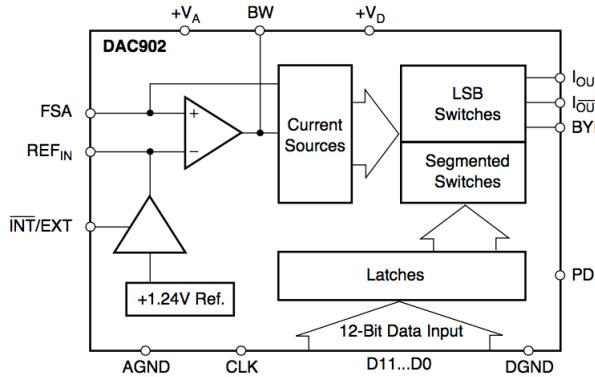


Figura 3.11: Schema a Blocchi interno del DAC - DAC902

Essi sono determinati dal tempo impiegato dagli interruttori per cambiare stato e dal tempo di risposta dell'amplificatore operazionale. In generale il tempo di conversione cresce all'aumentare della risoluzione. Un DAC a 8 bit per applicazioni generiche (*general-purpose*) possiede tempi di conversione compresi tra i  $100\text{ns}$  e  $1\text{ms}$ , mentre un convertitore a 16 bit può avere un tempo di conversione nell'ordine dei microsecondi.

Per applicazioni specifiche, come nel caso del misuratore oggetto di questo lavoro di tesi, sono necessari convertitori ad alta velocità con tempi di conversione nell'ordine dei nanosecondi. Per raggiungere tali prestazioni l'uso di una singola architettura non basta, pertanto è necessario combinare due o più architetture allo scopo di raggiungere le prestazioni richieste. Il procedimento è noto come "segmentazione" e questa architettura è chiamata "DAC segmentato (*Segmented DAC*)".

**Convertitore DAC presente sulla scheda di conversione** Il convertitore presente sulla scheda di conversione *SCO Board* è il *DAC902*. È un convertitore DAC Segmentato prodotto dalla *Texas Instruments* con una risoluzione di 12 bit e una velocità di campionamento massima pari a  $165\text{MHz}$ . Tale architettura rispetta pienamente le specifiche prefissate con una frequenza di campionamento massima ben superiore a quella richiesta.

Questo convertitore utilizza la tecnica della somma di correnti vista nelle precedenti due architetture, ma al contrario di quest'ultime i generatori di corrente sono implementati con transistor PMOS invece che con resistori e tensioni di riferimento. L'architettura segmentata utilizzata in questo integrato sfrutta due architetture: un DAC gestisce i bit più significativi (MSB)

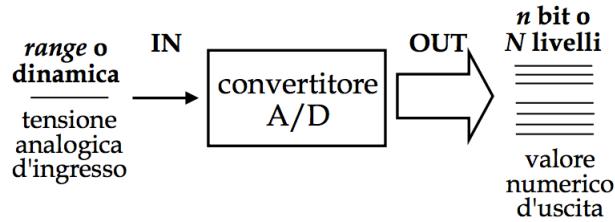


Figura 3.12: Schema di funzionamento di un convertitore Analogico-Digitale (ADC)

e un altro gestisce i bit meno significativi (LSB). Lo schema a blocchi è mostrato in figura 3.11.

Una descrizione più dettagliata dell'integrato è fornita sul sito del costruttore [6].

### Convertitore ADC

Il processo di campionamento di un segnale analogico consiste nel prelevare una lettura istantanea della sua grandezza e di convertirla in una forma digitale. Questa operazione è svolta dal convertitore Analogico-Digitale (ADC) [40].

**Architetture ADC** Esistono cinque architetture comuni di ADC:

1. A conteggio (*Counter*)
2. Ad approssimazioni successive (*Successive Approximation*)
3. A Doppia rampa (*Dual Slope*)
4. Flash
5. Pipeline

**ADC a conteggio** L'ADC a conteggio è una delle forme più semplici di convertitore ADC. Lo schema di funzionamento è mostrato in Figura 3.13.

Il cuore dell'architettura è costituito da un contatore incrementale. La conversione viene effettuata iniziando un nuovo conteggio a partire da zero. L'uscita del contatore viene convertita in analogico dal DAC e quindi confrontata con la tensione di ingresso mediante un comparatore. Un comparatore è un dispositivo che produce un'uscita booleana a seconda di quale dei due ingressi è maggiore. Quando il valore prodotto dal DAC supera la

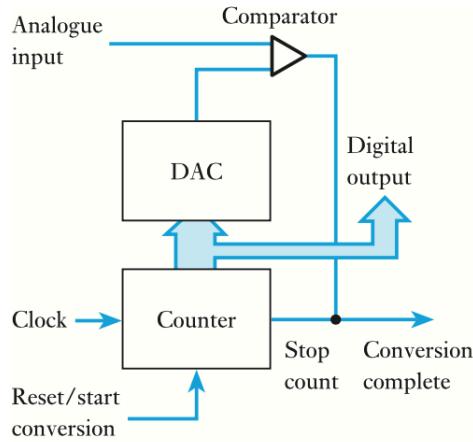


Figura 3.13: Schema di funzionamento di un ADC a conteggio

tensione di ingresso, il conteggio viene bloccato e tale valore rappresenta il valore di tensione di uscita.

Questo metodo è di semplice realizzazione, ma i principali difetti sono la lentezza e il tempo di conversione non costante. Per un ADC a  $n$  bit, la conversione può richiedere nel caso pessimo  $2^{n-1}$  cicli di clock. Il tempo di conversione è solitamente nell'ordine dei millisecondi (ca. 500Sa/s).

Una versione migliorata dell'ADC a conteggio è l'ADC a inseguimento (*servo ADC*): il contatore incrementale viene sostituito con un contatore incrementale/decrementale. Pertanto, l'uscita del comparatore è utilizzata come segnale di comando del contatore, forzando il contatore a "inseguire" segnale di ingresso analogico. Esso risulta più veloce dell'ADC a conteggio, a condizione che i valori di tensione da convertire siano abbastanza vicini fra loro.

**ADC ad approssimazioni successive** L'ADC ad approssimazioni successive è basato su una logica di controllo costituita da un registro ad approssimazioni successive chiamato SAR (*Successive Approximation Register*). Lo schema di funzionamento è mostrato in Figura 3.14.

Questa architettura è simile a quella dell'ADC a conteggio, fatta eccezione che il semplice contatore incrementale è sostituito dalla logica di controllo SAR. Come negli ADC a conteggio, la conversione avviene confrontando l'uscita di un convertitore DAC con la tensione in ingresso.

Il DAC viene comandato dalla parola digitale prodotta dalla logica di controllo. Inizialmente, tutti i bit della parola digitale sono impostati a 0

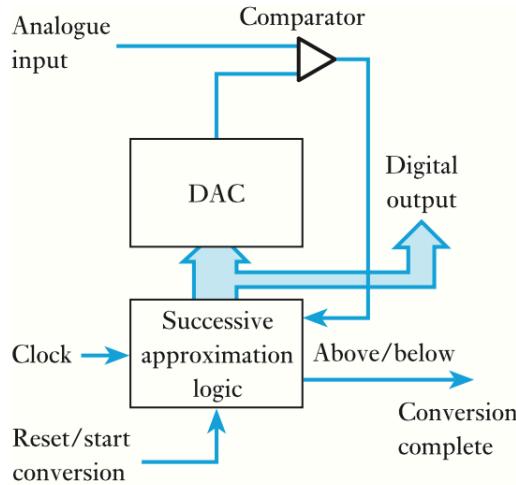


Figura 3.14: Schema di funzionamento di un ADC ad approssimazioni successive

mentre il bit più significativo (MSB, *Most Significant Bit*) è impostato a 1. Questo valore viene convertito e confrontato con il segnale di ingresso analogico utilizzando il comparatore ed il risultato del comparatore viene inviato al SAR. Se la tensione di ingresso è maggiore della parola digitale corrente, il SAR mantiene l'MSB a 1 e carica un altro 1 nel bit immediatamente successivo, altrimenti pone l'MSB a 0 e carica sempre un 1 nel bit immediatamente successivo. I passi appena descritti vengono reiterati fino al bit meno significativo (LSB, *Least Significant Bit*). Il metodo appena descritto è chiamato metodo di *bisezione*.

Il tempo di conversione dell'ADC è costante qualunque sia il valore della tensione in ingresso. Indicando con  $T_{ck}$  il periodo del clock e con  $n$  bit il numero di bit del convertitore, il tempo di conversione  $T_{conv}$  è pari a:

$$T_{conv} = nT_{ck} \quad (3.5)$$

Questa architettura raggiunge velocità di conversione migliori dell'ADC a conteggio, perdendo però in semplicità di realizzazione.

Solitamente i convertitori SAR possiedono risoluzioni comprese tra gli 8 e i 12 bit con tempi di conversione nell'ordine dei microsecondi ( $50 \div 400 KSa/s$ ).

**ADC a Doppia rampa** L'ADC a doppia rampa è un convertitore basato sull'utilizzo di un integratore. Lo schema di funzionamento è mostrato in Figura 3.15.

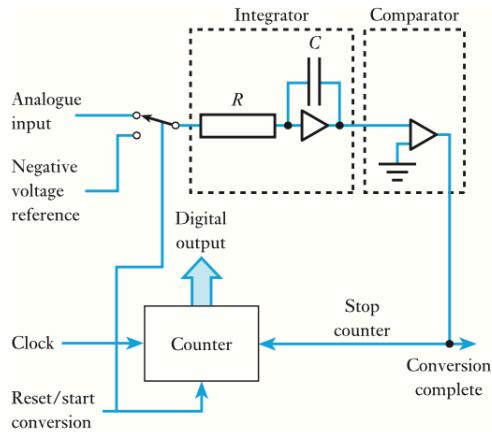


Figura 3.15: Schema di funzionamento di un ADC a doppia rampa

L'amplificatore operazionale integra il segnale di ingresso per un determinato periodo di tempo, producendo una carica sul condensatore proporzionale alla tensione di ingresso. L'integratore viene poi collegato ad un riferimento di tensione negativo che permette di scaricare il condensatore ad una velocità costante.

Il tempo impiegato per scaricare il condensatore viene misurato da un contatore. Il tempo di scarica è proporzionale alla quantità di carica presente nel condensatore e quindi alla tensione di ingresso. Tale conteggio corrisponde al valore digitale prodotto in uscita.

Questa tecnica si utilizza in quelle applicazioni in cui il segnale da convertire varia lentamente nel tempo, privilegiando la precisione rispetto alla rapidità di conversione. Con questi convertitori è possibile raggiungere risoluzioni maggiori di 20 bit portando però il tempo di conversione sull'ordine dei secondi ( $10 \div 100 Sa/s$ ).

**ADC Flash** L'ADC Flash è basato sull'utilizzo di compratori in parallelo. Lo schema di funzionamento è mostrato in Figura 3.16.

Il segnale di ingresso di un ADC Flash a  $n$  bit, viene confrontato con  $2^n$  tensioni di riferimento, tipicamente generate con una stringa resistiva realizzata con  $2^n$  resistori, tramite  $2^n$  comparatori. Il risultato è che tutti i comparatori che hanno tensioni superiori alla tensione di ingresso produrranno un'uscita di tensione positiva, mentre quelli collegati a tensioni al di sotto della tensione di ingresso produrranno tensioni di senso opposto. La logica combinatoria viene quindi utilizzata per determinare il valore digitale di uscita.

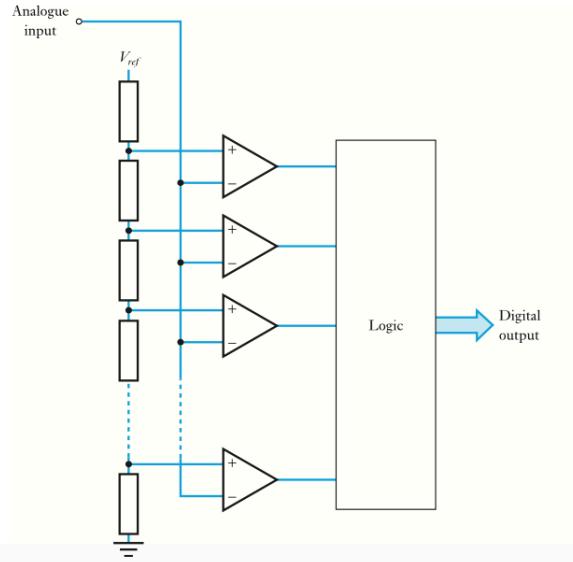


Figura 3.16: Schema di funzionamento di un ADC Flash

Il grosso vantaggio di questa tecnica è la velocità di conversione, essa permette di raggiungere tempi di conversione nell'ordine dei nanosecondi (fino a  $40GSa/s$ ): è la tecnica ADC più veloce.

Lo svantaggio è l'alto costo di realizzazione ( $n$  bit implica  $2^n$  comparatori). Pertanto, le risoluzioni tipiche sono di 8 bit.

**ADC Pipeline** L'ADC Pipeline è un convertitore che sfrutta il principio della pipeline. Lo schema di funzionamento è mostrato in Figura 3.17.

Una pipeline è un insieme di stadi di elaborazione connessi in serie, in cui l'uscita di un elemento è l'ingresso del successivo. Gli stadi sono identici, quindi è sufficiente spiegare il funzionamento del primo per comprendere il funzionamento di tutta l'architettura.

Il segnale analogico d'ingresso viene convertito da un ADC SAR a  $n$  bit. Il segnale digitale così prodotto costituisce l'uscita dello stadio. Il segnale ottenuto dal SAR, oltre che essere l'uscita dello stadio, costituisce l'ingresso di un DAC, che fornisce in uscita nuovamente un segnale analogico, che però differisce da quello originale in quanto affetto da errore di quantizzazione introdotto dal SAR.

Il campione così ottenuto va in ingresso a un sommatore che ne fa la differenza col segnale analogico originale, ottenendo così come risultato l'errore

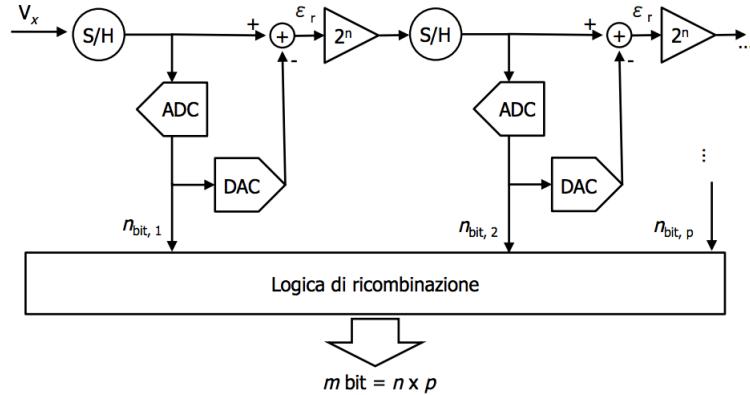


Figura 3.17: Schema di funzionamento di un ADC Pipeline

di quantizzazione. Successivamente l'errore di quantizzazione va in ingresso a un amplificatore di guadagno  $2^n$ , in modo da poter sfruttare al massimo l'intervallo di conversione del ADC SAR. Infine, gli stadi successivi al primo convertono l'errore di quantizzazione. I bit ottenuti in uscita dai diversi stadi vengono poi riallineati tramite opportuni registri, in modo da costituire la parola digitale di uscita.

La risoluzione di un convertitore Pipeline risulta limitata dalla precisione dei convertitori AD e DA presenti nell'architettura. La massima risoluzione ottenibile si aggira da un minimo di 8 bit a un massimo di 24 bit con velocità di conversione sull'ordine dei nanosecondi ( $1 \div 200MSa/s$ ).

**Convertitore ADC presente sulla scheda di conversione** Il convertitore presente sulla scheda di conversione *SCO Board* è l'*ADS807*. È un convertitore ADC Pipeline prodotto dalla *Texas Instruments* con una risoluzione di 12 bit e una velocità di campionamento massima pari a  $53MHz$ . Tale convertitore rispetta pienamente le specifiche prefissate.

La figura 3.18 mostra lo schema a blocchi interno del *ADS807*. Il circuito è costituito da quattro componenti principali:

- **Circuito di Track & Hold:** L'ingresso analogico consiste in un circuito differenziale di *Track & Hold*. Tale circuito è utilizzato per mantenere il segnale campionato costante durante tutto il periodo di conversione. A differenza di un *Sample & Hold*, il *Track & Hold* trascorre la maggior parte del tempo seguendo l'ingresso ed è posto in modo hold solo per un breve intervallo. Nei sistemi di acquisizione che operano ad elevate velocità, come in questo caso (superiori a  $1MHz$ ), i termini *Sample & Hold* e *Track & Hold* perdono la loro distinzione.

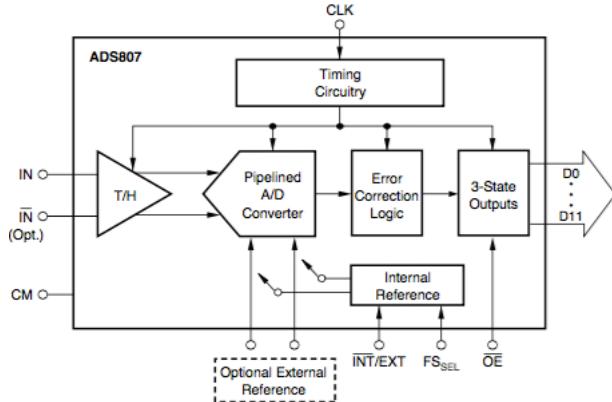


Figura 3.18: Schema di funzionamento del ADS807

- ADC Pipeline: L'architettura a Pipeline è formata da 12 stadi interni con una latenza di 6 cicli di clock. I dati in uscita diventano validi sul fronte di salita del clock.
- Logica per la correzione dell'errore (Error Correction Logic): La logica di correzione dell'errore è quella descritta nel paragrafo precedente dove il campione digitale prodotto ogni stadio della pipeline viene convertito da un DAC e sottratto al valore analogico di ingresso producendo così un errore di quantizzazione che verrà amplificato e inviato allo stadio successivo.
- 3 State Outputs: L'uscita digitale del convertitore possiede una logica a 3 stati. Nella logica *tri-state* è presente un terzo stato, detto ad alta impedenza, oltre ai due livelli logici già presenti nella logica binaria. Nello stato ad alta impedenza l'uscita si comporta come se fosse elettricamente disconnessa dal resto del circuito.

Una descrizione più dettagliata dell'integrato è fornita sul sito del produttore [5].

### 3.4 Parte digitale

La parte digitale dello strumento è composta da una scheda *embedded* di prototipazione.

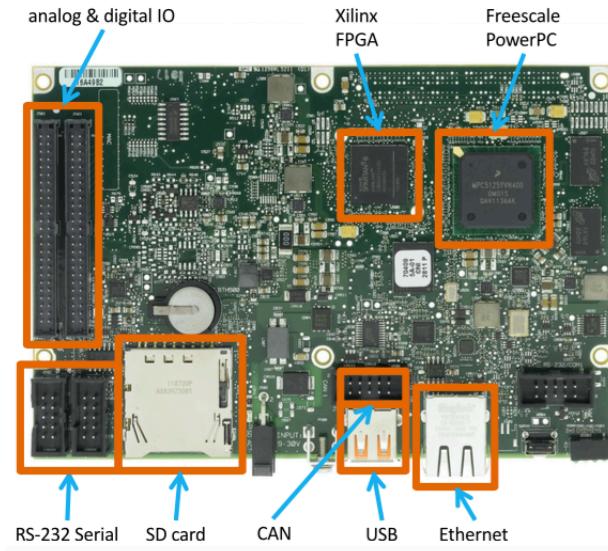


Figura 3.19: Scheda di elaborazione digitale, National Instruments Single-Board RIO (sbRIO) 9636

### 3.4.1 Scheda di prototipazione utilizzata

Per lo svolgimento della fase di prototipazione dello strumento è stata utilizzata la scheda di prototipazione *Single Board RIO 9636*, progettata e prodotta dalla *National Instruments*, Figura 3.19.

La principale motivazione che ha spinto all'utilizzo di un dispositivo *embedded* di prototipazione è la possibilità di raggiungere elevate prestazioni di calcolo. Inoltre, l'estrema ri-configurabilità di tale dispositivo ha permesso una rapida prototipazione dello strumento.

La scheda integra su un singolo PCB molteplici componenti elettronici; sono infatti presenti un microprocessore, in grado di eseguire software *Real-Time*, una FPGA, *Xilinx Spartan-6 LX45*, un DAC ed un ADC. Inoltre la scheda integra 4 output e 16 input analogici, a 16-bit, e 28 canali digitali, programmabili per essere utilizzati come input o output.

Sono presenti infine un connettore *10/100Base-T Ethernet* per la comunicazione con un PC, una USB, un lettore di SDHC, un connettore CAN e due connettori seriali (RS232 e RS485). Delle numerose componenti presenti sulla scheda, per il progetto presentato in questa tesi sono stati sfruttati, oltre al microprocessore ed all'FPGA, soltanto la connessione *Ethernet*, per comunicare con un PC host, e 24 dei 28 pin digitali (12 in ingresso e 12 in uscita) per la comunicazione con i convertitori della scheda di conversione

*SCO Board.*

A causa dei vincoli stringenti sulla temporizzazione non è stato possibile sfruttare i convertitori integrati nella *sbRIO*, che sono stati sostituiti con i componenti precedentemente descritti. Una descrizione più dettagliata della scheda è fornita sul sito di *National Instruments* [2].

Poiché la parte centrale dell'architettura software è stata sviluppata per essere eseguita da un'FPGA, in questo capitolo si darà ampio spazio ad una spiegazione generale della nascita e delle principali tecnologie impiegate nella costruzione di FPGA.

### 3.4.2 FPGA

Con il termine FPGA, acronimo di *Field Programmable Gate Array*, si intende un circuito integrato le cui funzionalità sono programmabili via software [27]. Rispetto ad una soluzione specializzata (ASIC, ovvero *Application Specific Integrated Circuit*) offrono numerosi vantaggi, infatti per la realizzazione di un'ASIC possono trascorrere mesi e la spesa è dell'ordine di centinaia di migliaia di dollari solo per la realizzazione del primo prototipo; al contrario un'FPGA può essere riconfigurata in pochi secondi e i suoi costi non superano qualche migliaio di dollari.

I vantaggi in termini di costo e velocità di sviluppo sono compensati da una maggiore area occupata, tipicamente dalle 20 alle 35 volte, una computazione 3-4 volte più lenta e un consumo energetico circa 10 volte maggiore [26].

Nonostante questi svantaggi le FPGA sono preferite in ambiti in cui i volumi di produzione non sono tali da giustificare la spesa per la realizzazione di un ASIC, e spesso rappresentano l'unica alternativa economicamente sostenibile.

#### Storia delle FPGA

L'origine delle moderne FPGA può essere ricondotta allo sviluppo dei circuiti integrati negli anni '60.

I primi dispositivi programmabili avevano architetture regolari e funzionalità flessibili. Infatti tipicamente si avevano architetture composte da array bidimensionali di porte logiche, con connessioni punto-punto, detti *cellular array*.

Questi primi array contenevano celle logiche che potevano essere programmate, attraverso la "metallizzazione", durante il processo produttivo per creare funzioni logiche a due ingressi. A metà degli anni '60 si è raggiunta la capacità di programmare i circuiti sul campo, ovvero dopo il processo

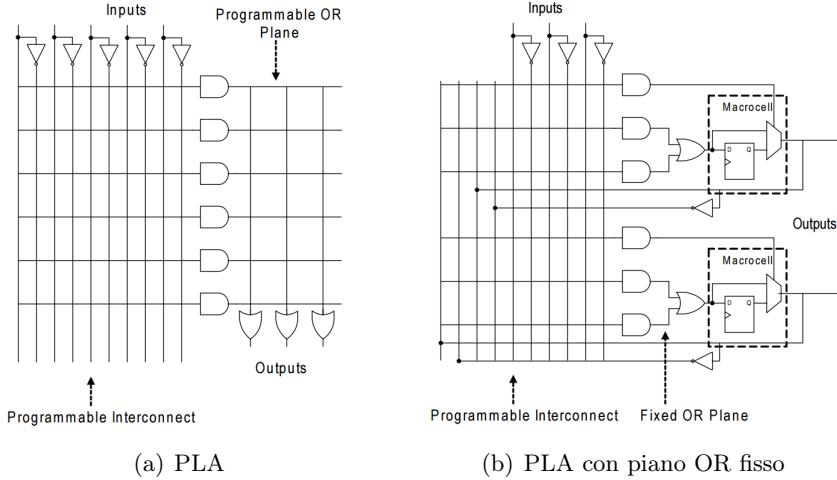


Figura 3.20: Tipologie di PLA

di produzione, attraverso l'introduzione di "punti di taglio" [32] nei *cellular array*. Le connessioni tra i diversi elementi logici erano ancora fisse, ma la funzionalità di ogni cella logica era modificabile grazie all'introduzione di fusibili programmabili. Questi fusibili erano programmabili sul campo attraverso l'uso di correnti di programmazione.

Negli anni '70 sono state introdotte le memorie PROM, che hanno portato nuovi modi di implementare funzioni logiche. Nonostante una ROM a  $N$  indirizzi possa implementare qualsiasi funzione logica con  $N$  ingressi, l'efficienza rispetto all'area si rivela essere un problema. Infatti l'area dipende in modo esponenziale dal numero di ingressi  $N$  e quindi, anche con valori piccoli di  $N$ , il chip avrà dimensioni elevate.

I primi dispositivi programmabili, chiamati PLA (*Programmable Logic Array*) si basavano su questa architettura, con una struttura a due livelli, AND e OR. Un esempio di PLA è mostrato in Figura 3.20(a).

Questa architettura si è poi evoluta in un piano AND programmabile, seguito da un piano OR fisso (Figura 3.20(b)). Questa seconda architettura mantiene comunque una buona flessibilità, ma al tempo stesso riduce la complessità del circuito.

Quest'architettura è stata introdotta sul mercato nel 1977 da *Monolithic Memories Incorporated* (MMI) [11].

Per realizzare una funzione logica con questo tipo di architettura è necessario usare uno o due livelli della struttura. Gli input e le combinazioni intermedie alimentano l'array attraverso un'interconnessione programmabile, tipicamente una *crossbar*. Questo tipo di implementazione dell'interconnes-

sione porta ad una crescita dell'area molto rapida, soprattutto per circuiti con molti ingressi.

La prima memoria basata su FPGA (*SRAM-based FPGA*) è stata introdotta da *Wahlstrom* nel 1967 [45].

Questa architettura usa uno stream di bits sia per la logica che per l'interconnessione. Al contrario della *cellular array* ogni cella logica può implementare sia una funzione logica che una memoria. Inoltre, la connessione tra celle logiche può essere facilmente modificata per permettere l'implementazione di diverse topologie del circuito. Nonostante la memoria statica (SRAM) offra una flessibilità maggiore nella programmabilità, richiede un'area maggiore rispetto all'implementazione ROM.

Questo problema ha ritardato l'introduzione in commercio di prodotti basati su memoria statica fino alla metà degli anni '80, quando il costo per transistor era diventato sufficientemente basso.

La prima FPGA moderna è stata introdotta da *Xilinx* nel 1984 [44], e conteneva la classica configurazione ad array di elementi logici configurabili. In particolare era formata da 64 blocchi logici e 58 input e output.

Dal primo esempio commerciale le FPGA sono cresciute enormemente, infatti una moderna FPGA può contenere oltre 300000 elementi logici equivalenti, ed oltre 1000 input ed output, in aggiunta ad elementi logici specializzati, che hanno permesso alle FPGA di incrementare le loro funzionalità e quindi le loro possibili applicazioni.

## Tecniche di implementazione delle FPGA

Come già accennato in precedenza, una FPGA consiste in un array di blocchi logici programmabili, potenzialmente di tipo differente, che includono logica generica, celle di memoria e moltiplicatori, circondati da una rete di routing programmabile, che permette di modificare l'interconnessione via software.

L'array è circondato da una serie di blocchi input/output programmabili, che connettono il chip con il mondo esterno. La struttura semplificata del chip è presentata in Figura 3.21.

Ogni FPGA si affida alla presenza di switch programmabili, che sono usati per dare al chip la sua caratteristica di programmabilità.

Nel corso degli anni sono state introdotte diverse tecniche per rendere programmabile un chip, partendo dalle EPROM, EEPROM, per arrivare alle tecniche *static-memory*, *flash* ed *anti-fuse*.

Di tutte le tecnologie introdotte soltanto le ultime 3 sono attualmente implementate nei moderni prodotti commerciali. In seguito saranno trattate soltanto le tecnologie attualmente in uso.

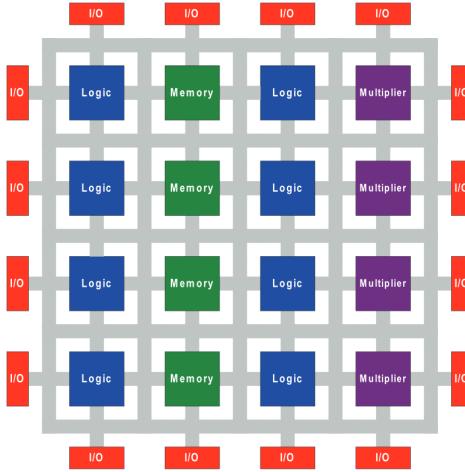


Figura 3.21: Struttura semplificata del chip FPGA

**Static-memory** Le celle statiche di memoria sono la base per la tecnologia SRAM. In questi dispositivi, le celle di memoria statica (Figura 3.22(a)) sono distribuite all'interno del chip FPGA, per garantire la programmabilità.

In questo tipo di architettura le SRAM hanno due principali utilizzi; alcune celle di memoria svolgono la funzione di *multiplexer*, per guidare i segnali attraverso le interconnessioni, mentre la maggior parte delle celle implementano le *Look Up Table* (LUT), che sono utilizzate tipicamente per implementare funzioni logiche. Le Figure 3.22(b) ed 3.22(c) illustrano questi possibili utilizzi.

La tecnologia SRAM è la più utilizzata nell'industria di FPGA per due principali vantaggi che porta, l'uso del processo produttivo CMOS standard e la riprogrammabilità. Una cella di memoria SRAM può essere riprogrammata un numero illimitato di volte. Un circuito dedicato presente nel chip inizializza tutti i bit delle SRAM all'avvio, sulla base di una configurazione fornita dall'utente.

A differenza delle altre tecnologie, l'uso di SRAM non richiede particolari lavorazioni sul circuito integrato, ma, come già detto, permette l'uso della tecnica CMOS standard. Questo porta come conseguenza diretta la possibilità di beneficiare dell'aumento di concentrazione di transistor, velocità e consumo energetico portati dell'uso del più recente processo produttivo disponibile sul mercato.

Tuttavia l'uso della tecnologia SRAM porta ad un certo numero di svantaggi:

- Area: Una cella SRAM richiede 5-6 transistor, e l'elemento riprogram-

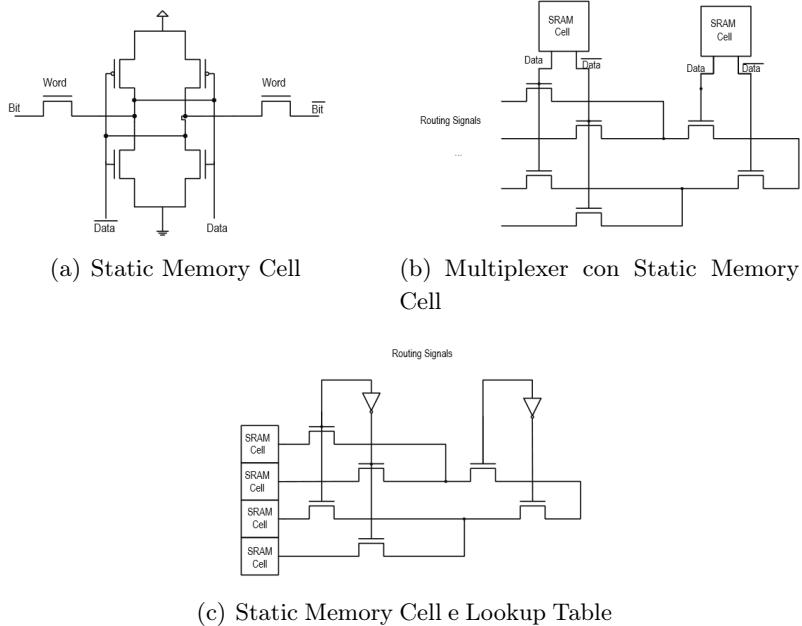


Figura 3.22: *Static Memory*

mabile usato per indirizzare i segnali richiede un transistor aggiuntivo.

- Volatilità: Le celle di SRAM sono volatili; ciò significa che è necessario usare un dispositivo esterno per salvare la configurazione in modo permanente quando il dispositivo è spento. Questi dispositivi di memoria portano ad un'incremento di costo nell'FPGA.
- Sicurezza: È possibile che la configurazione sia intercettata e rubata, a causa della necessità di caricare il contenuto delle SRAM all'avvio del dispositivo.
- Proprietà elettriche dei transistor di passaggio: Le FPGA basate su SRAM utilizzano transistor di passaggio per implementare i multiplexer. Tuttavia questi transistor non possono essere considerati switch ideali, poiché hanno una resistenza ed una capacità non trascurabili, che possono portare a problemi nell'uso di processi produttivi piccoli.

**Flash/EEPROM** Una tecnologia per superare alcuni degli svantaggi presenti nella tecnologia SRAM è basata sull'uso di transistor con tecnologia *floating gate*.

Questo approccio sta alla base dell'implementazione delle celle di memoria flash ed EEPROM. Queste celle di memoria hanno la caratteristica

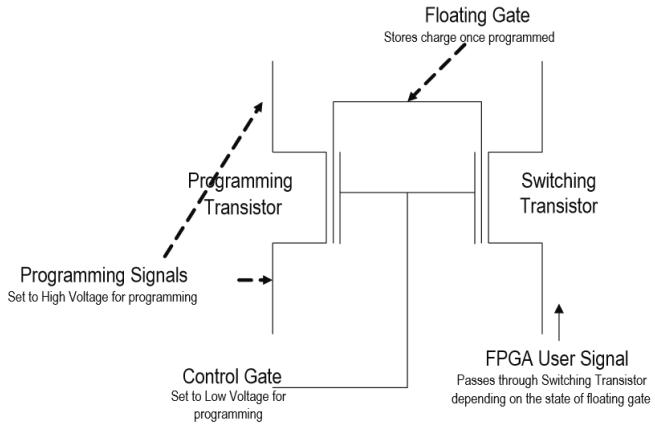


Figura 3.23: Implementazione del floating gate

di non essere volatili, quindi i dispositivi basati su questa architettura non hanno necessità di salvare la configurazione fornita dall’utente in memorie separate.

Storicamente le EEPROM non erano usate per controllare gli switch sui segnali dell’FPGA, ma erano usate per implementare le funzioni AND nei dispositivi PLD [35]. Questa tecnica non è più usata, tranne che nei dispositivi a piccola capacità [46], a causa dell’eccessivo consumo statico dei dispositivi con questa architettura.

I moderni processi produttivi hanno permesso di utilizzare le celle di memoria *floating gate* direttamente come switch. In particolare attualmente sono usate celle di memoria flash a causa della loro efficienza rispetto all’area. L’uso diffuso di memorie flash come memoria non volatile del chip assicura vantaggi nella diminuzione della dimensione geometrica dei transistor.

La Figura 3.23 mostra l’implementazione del *floating gate* nei dispositivi della linea *ProASIC* di *Actel* [13].

Il transistor più piccolo, nella Figura 3.23 *programming transistor*, è usato per programmare il *floating gate*, iniettando carica che rimane anche quando il dispositivo è spento, mentre il transistor più grande, nella figura *switching transistor*, funge da switch programmabile. Questa implementazione basata su memoria flash offre molteplici vantaggi, il più importante è la non volatilità. Inoltre, a causa della non volatilità, un dispositivo flash funzionerà immediatamente all’accensione, non dovendo aspettare il caricamento della configurazione.

Tuttavia la logica di programmazione introduce un overhead in area non presente nell’implementazione SRAM; questo costo è comunque modesto ed

è ammortizzabile su un gran numero di elementi programmabili.

Paragonata alla tecnologia *anti-fuse*, un’architettura non volatile alternativa che verrà approfondita nel paragrafo successivo, i dispositivi flash hanno il vantaggio di essere riconfigurabili e di poter essere programmati senza essere rimossi dal circuito stampato.

L’uso dei transistor *floating gate* richiede una particolare attenzione nella fase di progettazione, per assicurare che la tensione *source-drain* rimanga sufficientemente bassa da prevenire iniezione di carica nel *floating gate* [29]. Tuttavia, con l’uso di processi produttivi sempre più piccoli, e di conseguenza tensioni sempre meno elevate, questo problema perde di importanza.

I principali svantaggi dell’architettura flash sono l’impossibilità di riprogrammare il dispositivo un numero infinito di volte. La carica presente nello strato di ossido ad un certo punto non permette più di cancellare e riprogrammare il dispositivo correttamente [33]. I dispositivi attuali permettono di essere riprogrammati fino a 500 volte, un numero che nella maggior parte dei casi si rivela sufficiente.

Un altro svantaggio della tecnologia flash è l’uso di un processo CMOS non standard. Infine, come già spiegato per l’architettura SRAM, anche l’architettura flash soffre della presenza di resistenze e capacità non trascurabili causate dall’uso di transistor come switch. Un trend che sta emergendo ultimamente è l’uso di memorie flash in combinazione con tecnologie riconfigurabili SRAM. In alcuni dispositivi prodotti da *Altera*, *Xilinx* e *Lattice* la memoria flash è usata per garantire la non volatilità, mentre la SRAM fornisce la riprogrammabilità. Questo tipo di dispositivi, da un punto di vista architetturale, non sono diversi dai puramente SRAM.

Rispetto ad un dispositivo puramente SRAM si guadagna quindi la non volatilità, ma si va a perdere il vantaggio nell’uso di un processo produttivo standard e si introduce un overhead in termini di area a causa della replicazione dei bit, salvati sia in memorie flash che SRAM.

**Anti-fuse** Una tecnologia alternativa a SRAM e ai transistor *floating gate* è la tecnologia programmabile *anti-fuse*. Le interconnessioni di questa tecnologia sono strutture che in circostanze nominali mostrano una resistenza molto alta, ma che possono essere programmaticamente connessi per creare canali a bassa resistenza. A differenza delle precedenti tecniche, il collegamento è permanente. L’elemento programmabile, detto *anti-fuse*, è usato direttamente per trasmettere i segnali all’interno dell’FPGA.

Per implementare gli anti-fuse sono usati due approcci:

1. Dielettrico: Gli anti-fuse dielettrici sono composti da un dielettrico posizionato tra lo strato  $N+$  e il polisilicio [22]. L'applicazione di alte tensioni causano la rottura del dielettrico e formano un link conduttore con una resistenza tipicamente tra i 100 e i  $600\Omega$  [21].
2. Metal-to-metal: Gli anti-fuse *metal-to-metal* hanno largamente sostituito l'implementazione dielettrica. Questi *anti-fuse* sono formati da due strati di metallo, separati da uno strato di isolante (tipicamente silicio amorofo o ossido di silicio). Anche in questo caso l'applicazione di alte tensioni causa la rottura dell'*anti-fuse* e lo rende conduttivo. Il vantaggio di questa implementazione è la resistenza molto più bassa, tipicamente tra i 20 e i  $100\Omega$  [37].

Il principale vantaggio della tecnologia *anti-fuse* è il basso utilizzo di area. In particolare, con l'implementazione *metal-to-metal*, non è richiesta area in silicio per creare le connessioni; ciò diminuisce l'overhead in termini di area causato dalla programmabilità. Tuttavia questo decremento in termini di area è compensato dalla necessità di introdurre transistor di programmazione molto grandi, che devono fornire correnti elevati per programmare gli *anti-fuse*. Anche se questa area può essere distribuita tra molti *anti-fuse* con una progettazione opportuna, gli *anti-fuse* hanno un vantaggio aggiuntivo: hanno resistenze e capacità parassite più basse rispetto alle altre tecnologie.

L'area minore, la capacità e la resistenza ridotte permettono di includere più switch per dispositivo che nelle altre tecnologie. La non volatilità, come per i dispositivi basati su flash, significa che il dispositivo è pronto all'uso all'accensione; questo abbassa i costi, poiché non è necessaria una memoria aggiuntiva, e rende la tecnologia utilizzabile in quegli ambiti in cui il dispositivo deve essere pronto all'uso una volta acceso. Dato che la programmazione dell'FPGA va fatta una sola volta è possibile eseguirla in un ambiente controllato, migliorando così la sicurezza nel design dell'FPGA. Attualmente anche le altre tipologie di programmazione forniscono una modalità sicura, che disabilita l'accesso all'interfaccia di programmazione una volta che il dispositivo è programmato.

La tecnologia *anti-fuse* porta anche notevoli svantaggi. In particolare, l'uso di un processo CMOS non standard comporta l'uso di processi produttivi molto meno all'avanguardia rispetto all'architettura SRAM. Inoltre, la tecnica adottata per la programmazione, che implica cambiamenti significativi nelle proprietà dell'*anti-fuse*, porta a problemi di scaling. La tecnologia *anti-fuse* più avanzata (nel 2005) utilizza processi produttivi a  $150nm$  [12], che è diverse generazioni indietro rispetto alle più moderne tecnologie CMOS ( $90nm$  nel 2004 e  $65nm$  nel 2006).

L'impossibilità di riprogrammazione rende inutilizzabile questo tipo di dispositivi in ambiti dove sono necessari cambiamenti di configurazione, ma li rende ideali in ambiti in cui il rischio di corruzione delle memorie è elevato, come ad esempio l'ambito delle esplorazioni spaziali.

A differenza delle altre tecnologie, per l'*anti-fuse* non è possibile la programmazione su circuito stampato. Al contrario è necessario utilizzare specifici dispositivi di programmazione per programmare il chip prima che sia montato sul prodotto finito.

Infine la caratteristica di singola programmazione non consente ai produttori di testare gli *anti-fuse* per individuare possibili errori di produzione. Alcuni difetti possono essere individuati solo dopo la programmazione.

### 3.4.3 Xilinx Spartan-6 LX45

L'FPGA integrata nella scheda di prototipazione usata è la *Spartan-6 LX45*, prodotta da *Xilinx*.

In particolare questo prodotto integra al suo interno 54576 registri *Flip Flop* e 27288 *Look Up Table* (LUT) a 6 ingressi raggruppati in 6822 *Slices* (blocchi logici che contengono 8 *Flip Flop* e 4 LUT). Inoltre sono integrati 58 *DSP48* (che contengono un moltiplicatore  $18 \times 18$ , un sommatore ed un accumulatore), 5 canali DMA, 2088 Kbits di block RAM e 358 moduli di input/output riconfigurabili.

La scheda possiede un'oscillatore a  $40MHz$ , da cui possono essere derivati clock in un intervallo compreso tra  $2.5MHz$  e  $320MHz$ . La tecnologia usata per implementare lo switching è SRAM.

Una descrizione più dettagliata dell'FPGA utilizzata può essere trovata nella documentazione fornita da *Xilinx*[7].

## Utilizzo area FPGA

Il codice sviluppato per lo strumento di misura descritto in questo elaborato ha portato ad un notevole utilizzo di area del FPGA. I valori di utilizzo sono riportati in tabella 3.1. Inoltre, sono stati sfruttati due diversi clock di frequenza  $30MHz$  e  $60MHz$ .

### 3.4.4 Microcontrollore

Un microcontrollore è un dispositivo elettronico integrato in un singolo chip, utilizzato in alternativa ad un microprocessore nei sistemi *embedded*, ed in particolare per applicazioni specifiche di controllo digitale.

	Utilizzati	Totali	Utilizzo [%]
<b>Slices</b>	5560	6822	81,5
<b>Registri (FF)</b>	13465	54576	24,7
<b>Look Up Table (LUT)</b>	18003	27288	66
<b>Block RAM</b>	57	116	49,1
<b>Digital Signal Processor (DSP)</b>	57	58	98,3
<b>DMA Channel</b>	5	5	100

Tabella 3.1: Valori di utilizzo di area dell'FPGA

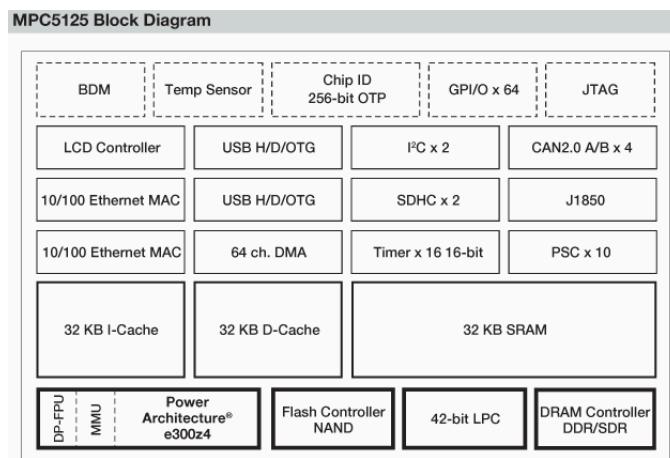


Figura 3.24: Schema a blocchi dell'architettura del microcontrollore MPC5125

Il microcontrollore utilizzato per il progetto descritto in questo elaborato, codice prodotto *MPC5125*, è prodotto da *Freescale*.

Nella Figura 3.24 è presentato lo schema a blocchi del microcontrollore; si può notare la presenza dei dispositivi tipicamente integrati in un prodotto di questa categoria, come ad esempio i GPIO, un modulo JTAG e le DMA.

Il core del microcontrollore è basata su architettura *PowerPC*, ed in particolare si tratta dell'architettura *Power e300z4*. Nel corso di questo paragrafo verrà data una spiegazione più dettagliata di questa architettura.

Infine la scheda integra 256MB di memoria RAM e 512MB di memoria non volatile, su cui sono caricati i programmi utente ed il sistema operativo.

### Architettura PowerPC

L'architettura *PowerPC* è nata nel 1991 dalla collaborazione di *Apple*, *IBM* e *Motorola*.

La filosofia alla base di questa architettura è denominata RISC (*Reduced Instruction Set Computer*) e prevede lo sviluppo di un processore in grado di eseguire poche istruzioni semplici, per accelerare i tempi di esecuzione. Questa filosofia è nata in contrapposizione alla filosofia CISC (*Complex Instruction Set Computer*) che si era diffusa all'inizio dell'era dell'industria informatica per facilitare il lavoro di sviluppo di programmi, cercando di emulare istruzioni di alto livello, in quanto non erano ancora disponibili i primi compilatori.

Una definizione più precisa delle architetture RISC può essere architettura *load-store*, in quanto le architetture RISC permettono l'accesso alla memoria di sistema unicamente con queste due funzioni.

L'architettura RISC è nata alla fine degli anni '70, quando i ricercatori di *IBM* notarono che la maggior parte delle funzioni di indirizzamento integrate nelle ISA (*Instruction Set Architecture*) erano ignorate dai programmatori, a causa dell'introduzione sul mercato dei primi compilatori, che erano in grado di gestire soltanto le istruzioni più semplici dei processori.

Inoltre scoprirono che alcune istruzioni complesse CISC erano più lente del loro corrispettivo sviluppato utilizzando una serie di istruzioni generiche. Per questi motivi si svilupparono set di istruzioni più semplici, che garantissero una progettazione del processore meno complessa e una velocità di esecuzione migliore. La conseguenza dell'introduzione di questo tipo di architetture è stata la possibilità di sfruttare soluzioni di parallelizzazione delle istruzioni, come la *pipeline*.

**PowerPC e300** L'architettura *e300* comprende una famiglia di processori a 32-bit sviluppata da *Freescale*, il cui principale utilizzo sono le applicazioni *embedded*.

L'architettura *e300* segue la filosofia RISC precedentemente illustrata ed implementa una pipeline a 4 stadi.

L'*e300*, inoltre, segue la filosofia *superscalar*, che prevede la possibilità di leggere dalla memoria e di completare più di un'istruzione per ciclo di clock, aumentando così il *throughput* del sistema.

La pipeline, come accennato sopra, si compone di 4 fasi principali:

1. Fetch: Si legge l'istruzione da eseguire dalla memoria di sistema e si calcola la posizione dell'istruzione successiva. Inoltre, se necessario, la BPU (*Branch Prediction Unit*) decodifica i branch e decide quale è l'istruzione successiva.
2. Dispatch: Decodifica l'istruzione proveniente dalla fase di fetch e determina quali istruzioni possono essere eseguite nel ciclo corrente. Inoltre

gli operatori delle operazioni sono letti dai registri sorgente ed inviati assieme all'istruzione alla fase successiva della pipeline.

3. Execution: In questa fase ogni unità funzionale a cui è stata assegnata un'istruzione esegue l'operazione, scrive il risultato nel registro di destinazione e notifica allo stadio finale della pipeline il termine dell'operazione. In questa fase è possibile che le operazioni richiedano più di un ciclo di clock.
4. Complete/write-back: questa fase si preoccupa di mantenere il corretto comportamento architettonico della macchina, ad esempio in presenza di interrupt la pipeline viene scaricata, i risultati delle precedenti computazioni sono scartati e il flusso di istruzioni viene indirizzato dalla sorgente corretta.

Nella fase di fetch delle istruzioni si è accennato alla presenza di una BPU. Quando si legge dalla memoria un'istruzione di branch non si può sapere a priori quale sarà l'istruzione successiva; la BPU svolge la funzione di effettuare una predizione su quale flusso sarà seguito dal programma ed in caso di successo nella predizione si ottiene come effetto l'assenza di ritardi causati dalla soluzione del branch.

Nell'architettura in esame la predizione della direzione del branch è fatta a *compile time*; all'istruzione di branch è aggiunto un bit che indica la direzione prevista per quel branch e la BPU, che contiene un sommatore al suo interno, è in grado di calcolare l'indirizzo dell'istruzione seguente. Il flusso di istruzioni si mantiene quello della previsione finché il branch non è risolto. A questo punto se la previsione era corretta il flusso di istruzioni viene mantenuto, altrimenti si provvede a scartare i risultati non utili ed il flusso di istruzioni ricomincia seguendo il verso opposto del branch.

Per ulteriori informazioni riguardo l'architettura e il microcontrollore in uso si faccia riferimento al manuale utente, incluso nella bibliografia di questo elaborato [36].



## Capitolo 4

# Architettura Software dello strumento

*In questo Capitolo verrà descritta l'architettura software dello strumento realizzato. Inizialmente verrà analizzato l'ambiente di sviluppo software. Successivamente verranno mostrati i principi alla base della programmazione per microcontrollore e FPGA. Infine, verranno illustrate e descritte le funzionalità implementate su FPGA e microcontrollore.*

### 4.1 Ambiente di sviluppo LabVIEW

L'ambiente di sviluppo scelto per questo lavoro di Tesi è NI LabVIEW.

LabVIEW, abbreviazione di *Laboratory Virtual Instrumentation Engineering Workbench*, è l'ambiente di sviluppo integrato per il linguaggio di programmazione visuale di *National Instruments*: il linguaggio G (*G-Language*, abbreviazione di *Graphical Language*).

La differenza sostanziale tra il linguaggio G e i linguaggi tradizionali risiede nella sintassi e nel controllo del flusso di programma:

- Sintassi: La sintassi del linguaggio G non è scritta ma grafica.
- Controllo del flusso di programma: Nei linguaggi tradizionali di tipo testuale, l'ordine di esecuzione delle istruzioni che costituiscono il codice del programma è determinato, a meno di ottimizzazioni portate dal compilatore, dall'ordine in cui le istruzioni sono scritte all'interno del codice stesso. Mentre, nel linguaggio G, l'ordine di esecuzione è stabilito dal "flusso di dati", ovvero ciascuna istruzione viene eseguita non appena sono disponibili i suoi dati di ingresso.

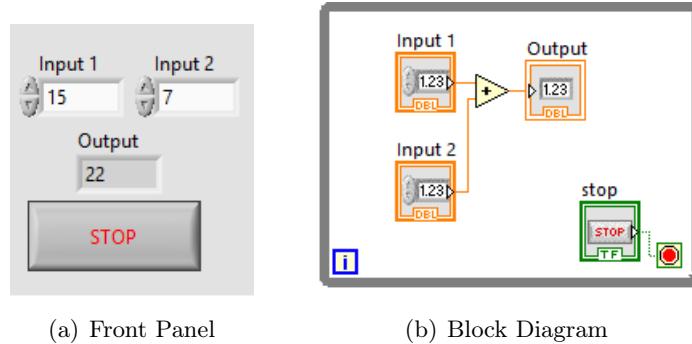


Figura 4.1: Esempio di LabVIEW VI che calcola la somma di due numeri in virgola mobile

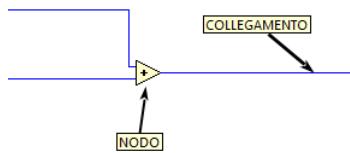


Figura 4.2: Esempio di Nodo e di Collegamento

I programmi generati da LabVIEW prendono il nome di "strumenti virtuali" (*Virtual Instrument*, VI). Un programma VI è composto da due parti fondamentali: il Pannello frontale (*Front Panel*) e il Diagramma a blocchi funzionale (*Block Diagram*).

Il pannello frontale è l'interfaccia utente del VI. Esso permette di definire ed introdurre tutte le grandezze di ingresso (input del programma) e tutte le grandezze in uscita (valori delle misure, grafici, ecc.). Si realizza con controlli e indicatori, che costituiscono, rispettivamente, i terminali interattivi d'ingresso e d'uscita.

Lo schema a blocchi è il diagramma di flusso che rappresenta il codice sorgente, in formato grafico. Esso è composto da due elementi distinti: i *nodi* e i *collegamenti*. I *nodi* sono gli elementi di elaborazione, mentre i *collegamenti* sono i fili che uniscono i vari nodi e permettono lo scambio di informazioni ovvero il flusso dei dati.

Le istruzioni, definite in fase di stesura del codice mediante il linguaggio grafico, vengono tradotte in modo trasparente in linguaggio C e successivamente compilate.

Le caratteristiche principali che hanno portato alla scelta di LabVIEW e del linguaggio G sono:

- LabVIEW un ambiente di sviluppo orientato fortemente all’acquisizione dati e all’analisi ed elaborazione numerica di segnali
- Il linguaggio G possiede un parallelismo intrinseco che è facile da utilizzare grazie alla metodologia di scrittura grafica dei programmi. La programmazione parallela consente di raggiungere elevate prestazioni di calcolo, specialmente nella programmazione per FPGA
- LabVIEW include al suo interno una serie di librerie per l’analisi dei segnali disponibili anche per la compilazione su FPGA che ha notevolmente accelerato il processo di sviluppo del codice
- LabVIEW è l’unico linguaggio di programmazione ufficialmente supportato dalla scheda di prototipazione scelta (vedi Capitolo 3)

## 4.2 Sistema Real-Time e FPGA

Come già accennato nel Capitolo precedente, lo scopo di questo lavoro di Tesi e la realizzazione di un prototipo di un misuratore di distanza basato sull’utilizzo di tecnologie FPGA e microcontrollore.

### 4.2.1 Programmazione del microcontrollore

Lo sviluppo di codice per un sistema *embedded* basato su microcontrollore si può suddividere in due approcci:

1. Bare-metal approach: consiste in una programmazione a basso livello dove il codice viene caricato direttamente sul dispositivo hardware (in questo caso microcontrollore). Questo approccio è utilizzato quando la complessità del sistema non è elevata. Pertanto garantisce un’occupazione di memoria molto piccola (*memory footprint*). Un grosso svantaggio è la poca flessibilità ai cambiamenti di specifiche.
2. Real-time Operating System approach: consiste in una programmazione basata sul supporto di un sistema operativo *Real-Time*. Questo approccio sfrutta tutti i vantaggi dell’utilizzo di un OS (programmazione *multi-thread*, astrazione dell’hardware, ecc...).

Per i motivi sopracitati si è scelto di sfruttare i vantaggi dell’utilizzo di un sistema Real-Time. Nel paragrafo successivo verranno trattate le caratteristiche dei sistemi Real-Time, i sistemi operativi Real-Time e in particolare il sistema operativo in uso sulla scheda di prototipazione scelta.

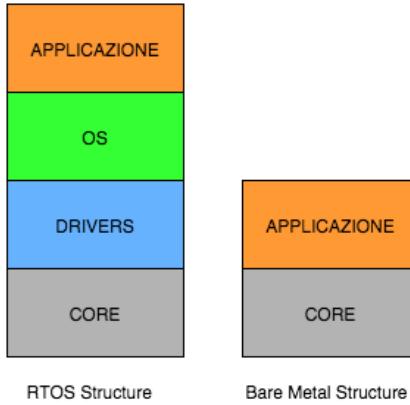


Figura 4.3: Schema di funzionamento dei due approcci: RTOS e Bare-metal

### Sistema Real-Time

Un *Real-Time System* (RTS) è un sistema in cui la correttezza del comportamento del sistema non dipende solo dall'esattezza dei risultati dei calcoli, ma anche dall'istante temporale in cui questi risultati vengono prodotti [25].

Un'applicazione Real-Time è costituita da un insieme di *task* (compiti) cooperanti. Ogni task possiede una scadenza temporale (*deadline*) entro la quale deve completare la sua esecuzione. Le deadline possono essere di tre tipologie:

1. Hard-deadline: Una deadline si dice *hard* se le conseguenze della sua violazione portano a un fallimento del sistema.
2. Firm-deadline: Una deadline si dice *firm* se i risultati del corrispondente *task* cessano di essere utili non appena la scadenza viene violata [38]. Gli effetti della sua violazione non sono catastrofici ma degradano le prestazioni del sistema.
3. Soft-deadline: Una deadline si dice *soft* se l'utilità dei risultati prodotti dal task diminuiscono nel tempo dopo la scadenza. Gli effetti della sua violazione non produce problemi.

**Sistema Operativo Real-Time** Un sistema operativo real-time (abbreviato RTOS, *Real-Time Operating System*) è un sistema operativo specializzato al supporto di applicazioni Real-Time.

Gli RTOS sono solo un elemento di un più complesso sistema real-time. I loro obiettivi sono quelli di fornire un ambiente comodo per riuscire a svilup-

pare efficacemente l'applicazione gestendo al meglio le risorse, ma soprattutto rispettando i vincoli temporali imposti [19].

Gli aspetti principali che differenziano un RTOS rispetto a un normale sistema operativo sono:

- Determinismo: è in grado di svolgere operazioni entro limiti di tempo prefissati.
- Latenza: La latenza è l'intervallo di tempo che intercorre fra il momento in cui arriva l'input al sistema ed il momento in cui è disponibile il suo output. Nei RTOS viene posto un limite massimo al tempo di latenza per garantire il determinismo.
- Controllo utente: Il controllo utente è più ampio rispetto ai normali OS. Il programmatore ha un controllo più fine sulle priorità, sull'uso di memoria dei processi e sulla gestione degli interrupt.
- Affidabilità: Gli RTOS sono progettati in modo da far fronte ai fallimenti del sistema, cercando di preservare quante più informazioni possibili. Contrariamente ai normali OS, non si notifica solo il guasto all'utente ma si cerca di risolvere il problema.

Lo scheduling nei RTOS è solitamente semplificato: politiche FIFO o *Round-Robin*. È comune l'utilizzo della prelazione (*preemption*) e delle priorità.

I Sistemi Operativi Real-Time più comuni in commercio sono: *VxWorks*, *LynkOS* e *Windows CE*. Esistono anche RTOS gratuiti e *open-source* come: *FreeRTOS* e *eCos*.

L'RTOS presente sulla scheda di prototipazione utilizzata è *VxWorks*.

**VxWorks** *VxWorks*, prodotto da *WindRiver*, è il più famoso sistema operativo per applicazioni real-time; la notorietà è dovuta al suo utilizzo da parte della NASA nelle sonde spaziali [19].

La struttura del sistema operativo è basata su *micro-kernel*. Il *micro-kernel*, contrariamente al kernel monolitico, fornisce solo le funzionalità strettamente necessarie alla gestione dei processi. Le funzionalità accessorie, come ad esempio la gestione di rete, sono rese disponibili da librerie esterne al kernel. Questa soluzione offre un ristretto utilizzo di memoria e libertà al progettista di personalizzare il sistema con le funzionalità prescelte.

Lo scheduling utilizzato da *VxWorks* utilizza la politica Round-Robin con diritto di prelazione e priorità. I task con priorità più elevata vengono

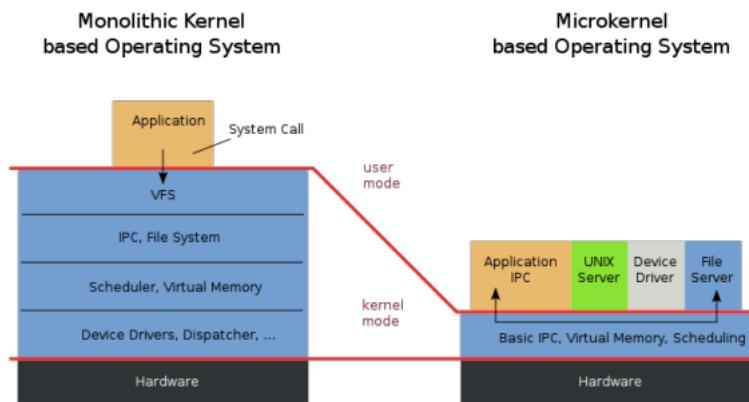


Figura 4.4: Confronto tra kernel monolitico e micro-kernel

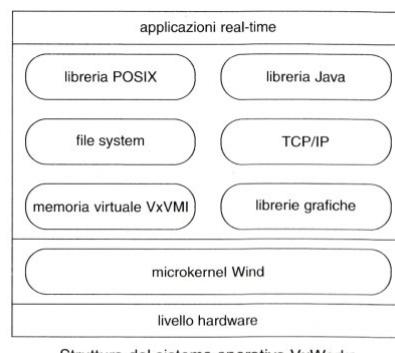


Figura 4.5: Struttura del sistema operativo VxWorks

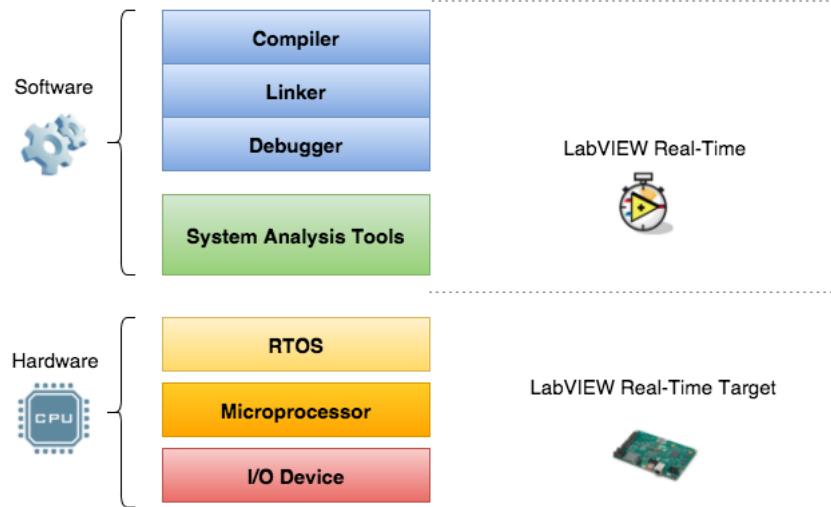


Figura 4.6: Struttura del modulo LabVIEW Real-Time

eseguiti per primi e nel caso di task con la stessa priorità l'esecuzione è Round-Robin.

La comunicazione tra processi può sfruttare una serie di meccanismi pre-definiti come: strutture dati condivisibili (es. variabili globali), semafori e code.

Maggiori informazioni sul sistema operativo *VxWorks* si possono trovare sul sito web dello sviluppatore [3].

**LabVIEW Real-Time** Per lo sviluppo dell'architettura Real-Time di questo lavoro di Tesi è stato utilizzato il modulo software LabVIEW Real-Time.

NI LabVIEW *Real-Time Module* è un componente aggiuntivo di LabVIEW utilizzato per la creazione di applicazioni Real-Time in esecuzione su dispositivi hardware *embedded*.

La Figura 4.7 mostra l'architettura software di base di un'applicazione Real-Time sviluppata in LabVIEW. Un'applicazione LabVIEW Real-Time si divide in due parti: applicazione *host* e *target*.

L'applicazione *host*, eseguita sul computer host, ha il compito di interfacciarsi con l'utente e comunicare con l'applicazione *target*.

L'applicazione *target*, invece, è l'applicazione Real-Time eseguita dal microprocessore del target computer. Nel nostro lavoro di Tesi il target computer è la scheda di prototipazione *Single Board RIO 9636* ampiamente descritta nel Capitolo 3.

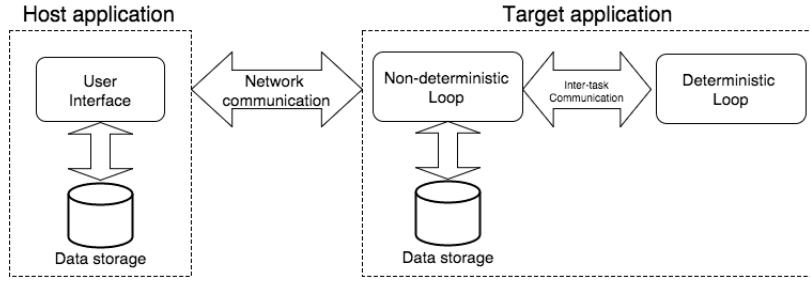


Figura 4.7: Architettura software di un'applicazione LabVIEW Real-Time

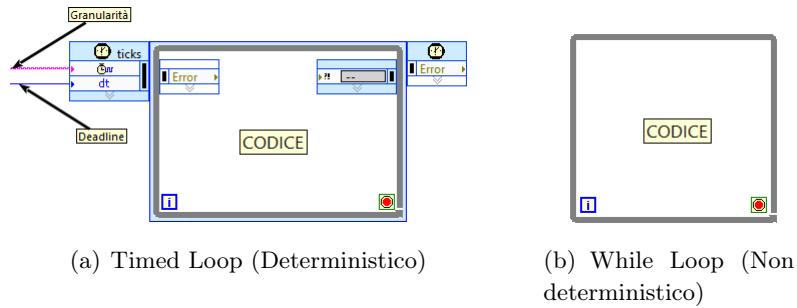


Figura 4.8: Cicli deterministici e non-deterministici

L'applicazione *target* è composta da processi. Un processo è un insieme di operazioni che si ripetono iterativamente nel tempo consumando una precisa quantità di tempo del microprocessore (*deadline*).

In LabVIEW un processo è rappresentato graficamente da un ciclo (*loop*) in cui sono racchiuse le operazioni da eseguire. Essi si suddividono in due categorie:

1. Deterministici (*Deterministic loop/process*): processi *Hard-deadline*. Sono rappresentati graficamente da Timed Loop (Figura 4.8(a)).
2. Non Deterministici (*Non-deterministic loop/process*): processi *Soft-deadline*. Sono rappresentati graficamente da While Loop (Figura 4.8(b)).

#### 4.2.2 Progettazione e programmazione dell'FPGA

FPGA, ampiamente descritto nel Capitolo 3, è un dispositivo logico le cui funzionalità sono programmabili via software.

Una delle tecniche più utilizzate per specificare la funzionalità di un dispositivo logico riprogrammabile consiste nell'uso di un linguaggio di descrizione dell'hardware (HDL, *Hardware Description Language*). Esisto-

no anche tecniche che utilizzano strumenti grafici (EDA, *Electronic Design Automation*) per definire lo schema circuitale.

L'HDL è adatto alla progettazione di grandi architetture perché permette di definire numericamente gli elementi circuitali. Tuttavia, l'utilizzo dello schema circuitale permette una lettura dell'architettura più chiara rispetto all'HDL.

### Linguaggi di descrizione dell'Hardware (HDL)

Un linguaggio per la descrizione dell'hardware o HDL (*Hardware Description Language*) è uno strumento di supporto alla progettazione dei circuiti digitali che ha lo scopo di cogliere gli aspetti funzionali e architetturali di un sistema [19].

La principale caratteristica di un HDL è la concorrenzialità: ovvero le diverse parti di un codice HDL una volta tradotte in un circuito elettronico, funzionano contemporaneamente, in quanto dispongono di hardware dedicato; al contrario, di un linguaggio software.

Esistono due linguaggi di descrizione dell'hardware attualmente utilizzati: VHDL e Verilog. La toolchain utilizzata per lo sviluppo di questo progetto, che verrà descritta in seguito, fa uso del linguaggio VHDL.

**VHDL** VHDL è l'acronimo di *VHSIC Hardware Description Language*, dove VHSIC è un altro acronimo: *Very High-Speed Integrated Circuits* [40].

La metodologia di programmazione di questo linguaggio si basa sul concetto di componente. Il componente è un'unità funzionale e il sistema è rappresentato da una rete gerarchica di componenti.

Le principali fasi di progettazione sono due:

- Entity: Un entity definisce l'interfaccia di un componente, in particolare le porte di comunicazione e altri parametri come: tempi di ritardo, larghezza del bus, ecc.
- Architecture: L'architecture definisce la funzionalità svolta dal componente. Per questa fase vengono usati di solito due stili:
  1. Behavioural: La relazione funzionale ingressi-uscite è espressa tramite un algoritmo. Vengono utilizzati i costrutti noti dei linguaggi di programmazione software come *if-then-else*.
  2. Structural: Non viene evidenziata la funzionalità ma la struttura interna del componente. La struttura è formata da componenti di basso livello (segnali, elementi di memoria, ecc.) ed i loro collegamenti (RTL, *Register Transfer Level*).

VHDL è diventato lo standard IEEE 1076 nel 1987. È stato aggiornato nel 1993 ed è conosciuto oggi come "standard IEEE 1076 1993" [39].

## Sintesi Hardware

La sintesi Hardware è il processo di compilazione che trasforma una specifica hardware, espressa in HDL o con schema circuitale, in un file di configurazione per il dispositivo hardware riprogrammabile.

Le principali fasi della sintesi hardware sono:

- Translation: Molti software per la sintesi hardware consentono al programmatore di poter scrivere la specifica hardware in diversi modi (HDL o schema circuitale). La prima fase della compilazione consiste nel combinare e tradurre le diverse specifiche in un'unica specifica completa.
- Functional simulation: Questa fase consiste nella simulazione dell'architettura progettata. Questo processo verifica la correttezza logica del circuito senza considerare i vincoli temporali.
- Optimisation: Dopo che l'architettura è stata valutata logicamente corretta, viene effettuata un ottimizzazione automatica al fine di semplificare l'implementazione. Le ottimizzazioni più comuni sono le semplificazioni aritmetico-logiche.
- Mapping: L'architettura ottimizzata viene distribuita sulle risorse disponibili del dispositivo hardware. Il processo è molto impegnativo nel caso di programmazione FPGA perché esistono svariati modi per tracciare le interconnessioni.
- Place and route: La configurazione risultante dal processo di Mapping viene sottoposta ad una dettagliata simulazione temporale. Questa fase verifica che l'architettura realizzata rispetti i vincoli temporali sul dispositivo hardware reale.
- Configuration data: Se i risultati della simulazione temporale sono soddisfacenti, viene generato un file di configurazione chiamato *bitfile*. Il *bitfile* contiene tutte le informazioni necessarie per configurare correttamente l'architettura progettata sul dispositivo hardware programmabile.

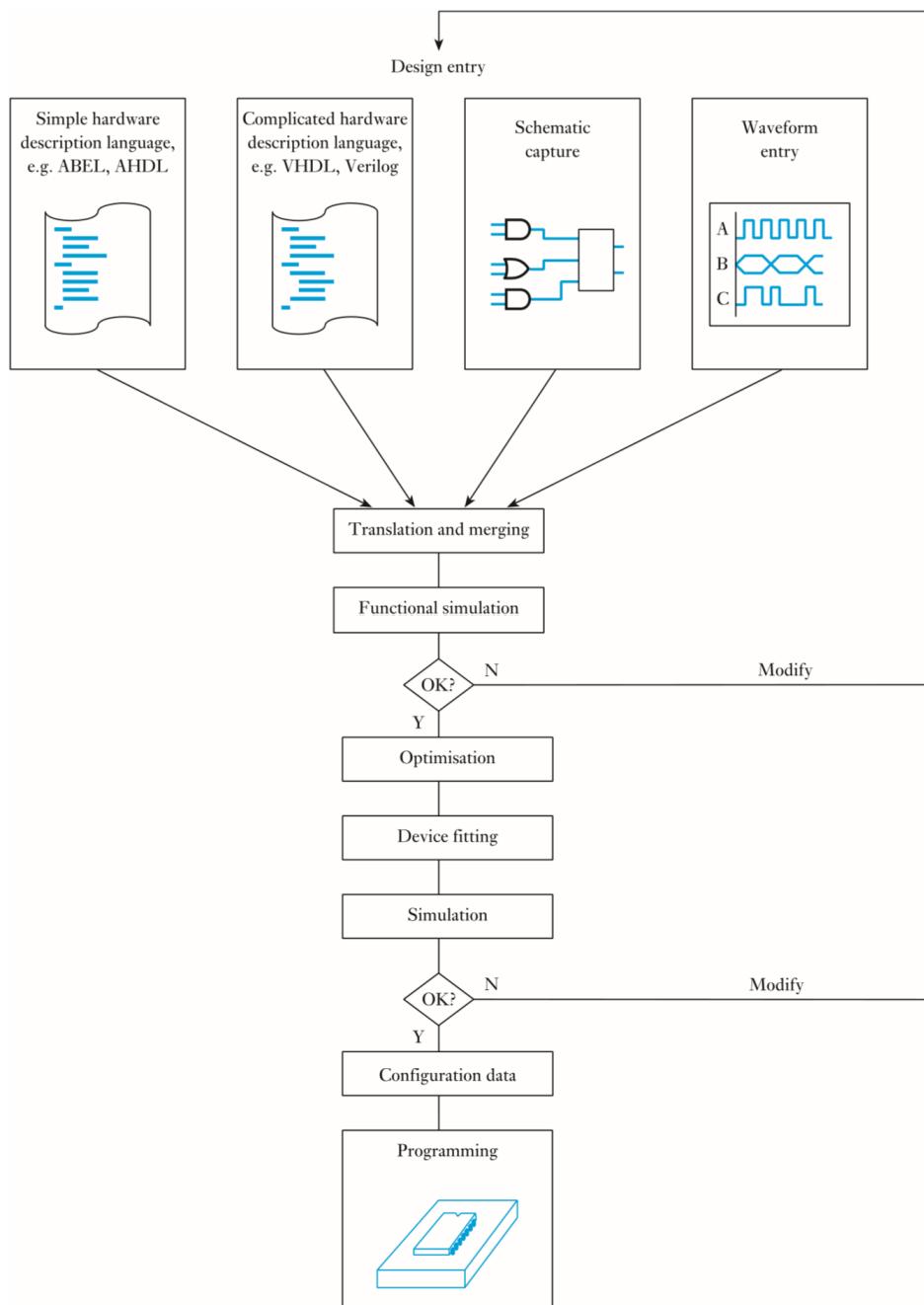


Figura 4.9: Sintesi hardware

## High-Level Synthesis (HLS)

*High-level synthesis* (HLS), in italiano Sintesi ad alto livello, è un processo automatizzato di compilazione che interpreta una descrizione algoritmica di un desiderato comportamento e crea l'hardware digitale che implementa tale comportamento [14].

La descrizione algoritmica del comportamento è espressa con linguaggi di alto livello come C, infatti è talvolta chiamata *C-Synthesis*. Nella pratica, gli strumenti di compilazione HLS convertono codice C o C-like in un linguaggio di descrizione dell'hardware (HDL), come VHDL o Verilog.

La scrittura di codice di alto livello permette al progettista di concentrarsi solo sull'aspetto funzionale dell'architettura. Quindi, ad un livello di astrazione più alto, sono necessari meno dettagli per la descrizione di un comportamento. Ad esempio, non si ha bisogno di preoccuparsi dei dettagli implementativi come gerarchie, processi, temporizzazioni, ecc.. come avviene per i linguaggi di descrizione dell'hardware. Questo rende la descrizione molto più facile da scrivere, riducendo notevolmente il rischio di errori e il tempo di testing.

La sintesi HLS è costituita da una serie di attività e diversi tool HLS eseguono queste attività in ordine diverso utilizzando algoritmi differenti. Altri tool HLS combinano alcune di queste attività e le eseguono iterativamente al fine di convergere verso la soluzione desiderata [31].

Di seguito vengono elencati i passi principali del processo di sintesi di alto livello:

1. Algorithm optimization: Nella prima fase, vengono eseguite ottimizzazioni del codice comunemente usate nei compilatori dei linguaggi ad alto livello come ad esempio: *Common Subexpression Elimination* (CSE), *Constant Propagation*, ecc.
2. DataFlow Graph analysis: Vengono analizzate le operazioni aritmetico-logiche e le dipendenze tra i dati. Il risultato di questa analisi si traduce nella costruzione di un grafo chiamato *DataFlow Graph* (DFG). Il grafo rappresenta le dipendenze dei dati e indica l'ordine di esecuzione delle operazioni.
3. Resource allocation: Dopo che il DFG è stato creato, ogni operazione aritmetico-logica viene attribuita ad una risorsa hardware. La risorsa hardware corrisponde ad un'implementazione fisica dell'operatore aritmetico-logico. Ogni operatore può avere più implementazioni hardware, ciascuna con differenti caratteristiche di area/ritardo/latenza.

Queste risorse sono selezionate da una libreria tecnologica che contiene tutte le implementazioni disponibili.

4. Scheduling: Questa fase introduce il concetto di tempo e di parallelismo. Lo scheduling prende le operazioni descritte nel DFG e decide quando (in quale ciclo di clock) saranno eseguite [18].
5. Module binding: Il module binding è la fase che assegna le operazioni aritmetico-logico a specifiche istanze di risorse hardware. Questa fase si basa sulle tipologie e le quantità di risorse hardware scelte nella fase di resource allocation.
6. Register binding: I registri di memoria sono necessari quando i valori prodotti in un ciclo di clock vengono utilizzati in un ciclo di clock differente. La fase del Register Binding consiste nell'allocare i registri necessari a conservare questi valori. In questa fase viene effettuata l'analisi del ciclo di vita di ogni valore allo scopo di poter utilizzare lo stesso registro fisico per memorizzare valori diversi in istanti di tempo diversi.
7. Output processing: I risultati prodotti dai passi precedenti vengono tradotti in codice HDL.

Per i motivi esposti all'inizio del paragrafo, abbiamo scelto di utilizzare un'ambiente di sviluppo che permettesse di sfruttare i benefici della sintesi ad alto livello. L'ambiente di sviluppo utilizzato per la programmazione della scheda FPGA è descritto nel paragrafo successivo.

## LabVIEW FPGA

Per lo sviluppo dell'architettura FPGA di questo lavoro di Tesi è stato utilizzato il modulo software LabVIEW FPGA. NI LabVIEW *FPGA Module* è un componente aggiuntivo di LabVIEW utilizzato per sviluppare applicazioni per FPGA.

Questo ambiente di sviluppo compila ed esegue il codice LabVIEW sul dispositivo FPGA. Il processo di compilazione automatica è suddiviso in tre fasi:

1. High-Level Synthesis: Il codice di alto livello LabVIEW viene tradotto in codice VHDL
2. Hardware Synthesis: Il codice VHDL, prodotto nel passo precedente, viene tradotto dal compilatore *Xilinx ISE*. *Xilinx ISE Compiler* è uno

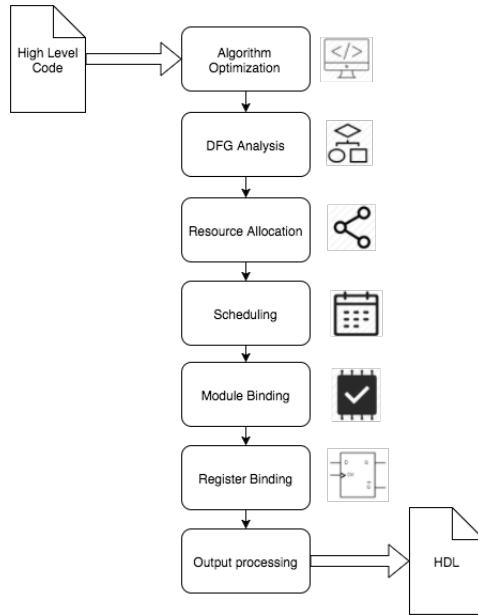


Figura 4.10: High Level Synthesis

strumento software prodotto da *Xilinx* che effettua la sintesi hardware di codice HDL. Il risultato finale di questa fase è l'FPGA *bitfile*.

3. Bitfile download: Il *bitfile* generato dal compilatore di *Xilinx* viene caricato ed eseguito sull'FPGA.

Al contrario del microprocessore, non esiste alcun sistema operativo sul chip FPGA, ma LabVIEW FPGA offre comunque la possibilità di controllare ingressi e uscite attraverso un applicazione host.

### 4.3 Analisi teorica degli algoritmi implementati

In questo paragrafo verranno introdotti i concetti fondamentali alla base dei principali algoritmi utilizzati nell'ambito dello sviluppo del firmware dell'interferometro.

Gli algoritmi utilizzati sono:

- Fast Fourier Transform (FFT)
- Interpolated Fast Fourier Transform (IFFT)
- Calcolo della distanza assoluta

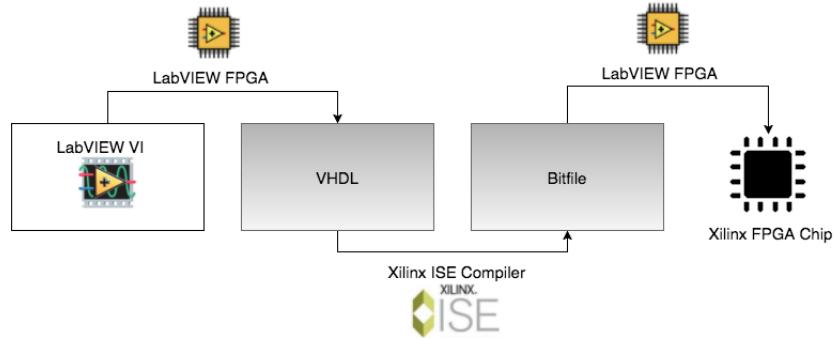


Figura 4.11: Processo di compilazione LabVIEW FPGA

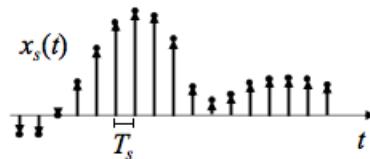


Figura 4.12: Segnale campionato con treno di impulsi

#### 4.3.1 Fast Fourier Transform (FFT)

Prima di illustrare il funzionamento dell'algoritmo di FFT, è bene richiamare alcuni concetti sull'analisi in frequenza dei segnali.

I principali metodi di analisi dei segnali di misura sono l'analisi nel dominio del tempo e nel dominio della frequenza. I due approcci sono tra loro intercambiabili, ovvero, sotto opportune condizioni, nessun informazione viene persa nel passare da un dominio all'altro.

Lo strumento matematico che consente di trasferire lo studio dei segnali dal dominio del tempo al dominio della frequenza è la trasformata di *Fourier*.

Il campionamento di un segnale analogico  $x(t)$  consiste nel prenderne solo i valori  $x(iT_s)$  in corrispondenza di istanti ben precisi  $iT_s$  detti istanti di campionamento. Il campionamento ideale consiste nel moltiplicare il segnale  $x(t)$  per il treno di impulsi  $s(t)$ :

$$x_s(t) = x(t)s(t) = \sum_{n=-\infty}^{+\infty} x(nT_s)(\delta(t - nT_s)) \quad (4.1)$$

Lo spettro in frequenza del segnale campionato è dato dalla Trasformata

di Fourier Tempo Discreta (DTFT), espressa dall'equazione:

$$X(e^{j\omega}) = \sum_{n=-\infty}^{+\infty} x(n)e^{-j\omega n} \quad (4.2)$$

dove  $x(n)$  indica la sequenza infinita di campioni in ingresso e  $\omega$  la pulsazione continua espressa in radianti.

Considerando, invece, una sequenza finita  $x[n]$  di lunghezza  $N$ , a cui corrisponde la DTFT  $X(e^{j\omega})$ , è possibile definire la Trasformata di Fourier Discreta (DFT) come la sequenza di  $N$  campioni:

$$X(k) = X(e^{j\omega})|_{\omega=\frac{2\pi k}{N}} = \sum_{n=0}^{N-1} x(n)e^{-j\frac{2\pi}{N}kn} \quad (4.3)$$

La DFT è costituita da un campionamento delle pulsazioni della DTFT con passo di quantizzazione  $\frac{2\pi}{N}$  dove ogni pulsazione quantizzata:

$$\omega_k = \frac{2\pi k}{N} \quad (4.4)$$

con:

$$0 \leq k \leq N - 1$$

viene chiamata *bin*. Si nota che lo spettro è calcolato unicamente per le frequenze di bin.

Per ottenere la sequenza  $x[k]$  a partire dai campioni  $x[n]$  è necessario uno sforzo computazionalmente considerevole (complessità quadratica  $O(N^2)$ ). Esso è dovuto all'elevato numero di operazioni di moltiplicazione e addizione presenti nel calcolo della DFT. Per tale motivo, sono stati realizzati algoritmi che riducono notevolmente il numero di operazioni necessarie rendendo il costo computazionale meno cospicuo.

Questi algoritmi hanno complessità  $O(N \log N)$  e prendono il nome di *Fast Fourier Transform (FFT) algorithm*.

L'algoritmo FFT più diffuso è il *Cooley-Tukey algorithm* che si basa sul principio di *divide et impera*. L'algoritmo decomponete ricorsivamente ad ogni passo la DFT in dimensioni più piccole.

Ai fini della trattazione dell'algoritmo di *Cooley-Tukey* si introduce il *twiddle factor*  $W_N = e^{-j\frac{2\pi}{N}}$  riscrivendo l'equazione 4.3 come:

$$X(k) = \sum_{n=0}^{N-1} x(n)W_N^{kn} \quad (4.5)$$

L'uso più conosciuto dell'algoritmo è di dividere ricorsivamente la DFT in due parti da  $\frac{N}{2}$  ad ogni passo. Esso è quindi ottimizzato solo per dimensioni

che siano potenze di due, ma in generale può essere utilizzato con qualsiasi fattorizzazione.

Se consideriamo come numero di campioni  $N$  della DFT una potenza di due possiamo riscrivere l'equazione 4.5 separando i termini di indice pari e di indice dispari ottenendo così l'equazione:

$$\begin{aligned} X(k) &= \sum_{n \text{ pari}} x(n) W_N^{kn} + \sum_{n \text{ dispari}} x(n) W_N^{kn} \\ &= \sum_{n=0}^{\frac{N}{2}-1} x(2n) W_N^{k2n} + \sum_{n=0}^{\frac{N}{2}-1} x(2n+1) W_N^{k(2n+1)} \end{aligned} \quad (4.6)$$

Essendo valida la relazione  $W_N^2 = W_{\frac{N}{2}}$ , possiamo riscrivere la precedente equazione come:

$$\begin{aligned} X(k) &= \sum_{n=0}^{\frac{N}{2}-1} x(2n) W_{\frac{N}{2}}^{kn} + W_N^k \sum_{n=0}^{\frac{N}{2}-1} x(2n+1) W_{\frac{N}{2}}^{kn} \\ &= X_1(k) + W_N^k X_2(k) \end{aligned} \quad (4.7)$$

Siccome  $X_1(k)$  e  $X_2(k)$  sono periodiche di periodo  $\frac{N}{2}$  ed è valida la relazione  $W_N^{k+\frac{N}{2}} = -W_N^k$ , possiamo riscrivere  $X(k)$  come:

$$X\left(k + \frac{N}{2}\right) = X_1(k) - W_N^k X_2(k) \quad (4.8)$$

Dalle precedenti equazioni si osserva che il calcolo di  $X_1(k)$  e  $X_2(k)$  richiede  $(\frac{N}{2})^2$  moltiplicazioni mentre  $W_N^k X_2(k)$  ne richiede solo  $\frac{N}{2}$ , ottenendo così  $2(\frac{N}{2})^2 + \frac{N}{2}$  moltiplicazioni per il calcolo di  $X(k)$ .

In confronto all'algoritmo di DFT vi è una riduzione di un fattore due del tempo di computazione  $\left(2(\frac{N}{2})^2 + \frac{N}{2}\right) \approx (\frac{N}{2})^2 < N^2$ .

Procedendo ricorsivamente e applicando la stessa tecnica di calcolo sulla DFT decomposta si ottengono  $\log N$  iterazioni della procedura se  $N$  è potenza di 2. Pertanto, il calcolo della trasformata di un vettore con  $N$  componenti richiama in maniera ricorsiva il calcolo della trasformata a due vettori con  $\frac{N}{2}$  componenti in cui sono presenti  $O(N)$  operazioni di somma e prodotto aggiuntive.

Possiamo quindi definire, per il calcolo della complessità dell'algoritmo di FFT, il numero totale  $T(N)$  delle operazioni necessarie per il calcolo della trasformata di Fourier di un vettore con  $N$  componenti:

$$T(N) = \begin{cases} 0, & N = 1 \\ 2T\left(\frac{N}{2}\right), & N > 1 \end{cases} \quad (4.9)$$

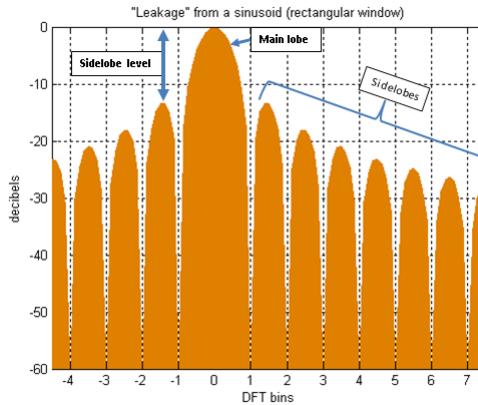


Figura 4.13: Spectral leakage

la cui soluzione è  $T(N) = O(N \log N)$ .

Alla luce dei risultati appena esposti, possiamo concludere che l'algoritmo di FFT riduce notevolmente il tempo di computazione portando il numero di operazioni a  $(\frac{N}{2}) \log N$ .

Nel listato 1 è mostrato lo pseudocodice dell'algoritmo di FFT.

Il calcolo della FFT è caratterizzato però da un problema che riguarda la valutazione finale dello spettro di frequenza: il calcolo della FFT si può considerare corretto solo se il segnale è periodico e l'analisi viene effettuata su un segnale campionato coerentemente, ovvero se si considerano un numero intero di periodi del segnale.

---

**Algorithm 1** Algoritmo di Coley-Tukey FFT

---

```

1: procedure  $X_{0,\dots,N-1} \leftarrow \text{FFT}(x, N, s)$ 
2:   if  $N = 1$  then
3:      $X_0 \leftarrow x_0$ 
4:   else
5:      $X_{0,\dots,\frac{N}{2}-1} \leftarrow \text{FFT}(x, \frac{N}{2}, 2s)$ 
6:      $X_{\frac{N}{2},\dots,N-1} \leftarrow \text{FFT}(x + s, \frac{N}{2}, 2s)$ 
7:     for  $K = 0$  to  $\frac{N}{2} - 1$  do
8:        $t \leftarrow X_k$ 
9:        $X_k \leftarrow t + e^{-2\pi i \frac{k}{N}} X_{k+\frac{N}{2}}$ 
10:       $X_{k+\frac{N}{2}} \leftarrow t - e^{-2\pi i \frac{k}{N}} X_{k+\frac{N}{2}}$ 
11:     end for
12:   end if

```

---

Il calcolo del FFT su una finestra di osservazione non contenente un numero intero di periodi produce un allargamento e uno spostamento delle righe dello spettro di frequenza; questo fenomeno è chiamato *spectral leakage* (dispersione spettrale, Figura 4.13).

Quindi, oltre a dover rispettare il teorema del campionamento di *Nyquist*, bisogna prestare attenzione alla dispersione spettrale che causa *aliasing* del segnale.

Applicare l'algoritmo di FFT direttamente alla sequenza campionata equivale ad usare una funzione di finestratura rettangolare, che pesa uniformemente i campioni.

Una soluzione che limita il fenomeno di *spectral leakage* è l'utilizzo di una finestra rettangolare di troncamento contenente un numero intero di periodi. Tale soluzione non è sempre realizzabile perciò, nella pratica, si utilizza un'altra soluzione che consiste nel sostituire la finestra rettangolare con finestre che presentano una transizione graduale alle estremità.

Pertanto, un importante aspetto da tenere in considerazione quando si esegue il calcolo della FFT di un segnale periodico campionato e l'operazione di finestratura [10].

Le finestre con transizione graduale alle estremità, chiamate anche *smoothing window*, risolvono i problemi dovuti ad un campionamento non coerente. I problemi del campionamento non coerente si manifestano principalmente agli estremi della finestra di osservazione, dove si osservano discontinuità, quindi è ragionevole ipotizzare che se si potessero trascurare gli estremi e si potesse concentrare l'analisi sulla parte centrale della finestra di osservazione si otterrebbe uno spettro in frequenza più corretto.

Pertanto, come già accennato, le *smoothing window* pesano differentemente i vari campioni assumendo valore basso agli estremi e valore elevato nelle porzioni centrali della finestra.

Le *smoothing window* sono divise tra finestre cosinusoidali e non cosinusoidali. Una delle più utilizzate è quella cosinusoidale di *Hanning*.

La funzione peso della finestratura di *Hanning* è definita dalla seguente relazione:

$$w(n) = \frac{1}{2} \left[ 1 - \cos \left( \frac{2\pi n}{N-1} \right) \right] \quad (4.10)$$

con:

$$0 \leq n \leq N-1$$

ed è mostrata in figura 4.14. La forma di questa finestra consente di eliminare la discontinuità del segnale agli estremi in caso di campionamento non coerente.

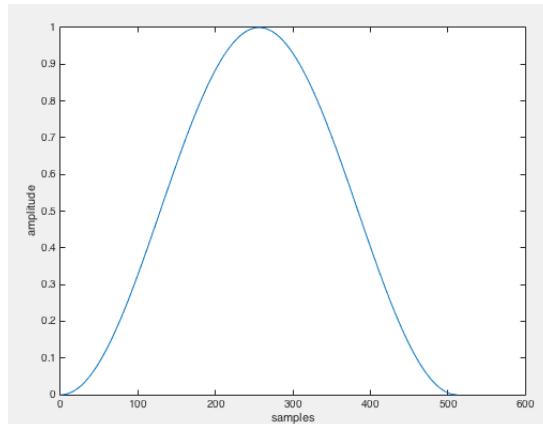


Figura 4.14: Finestra di Hanning

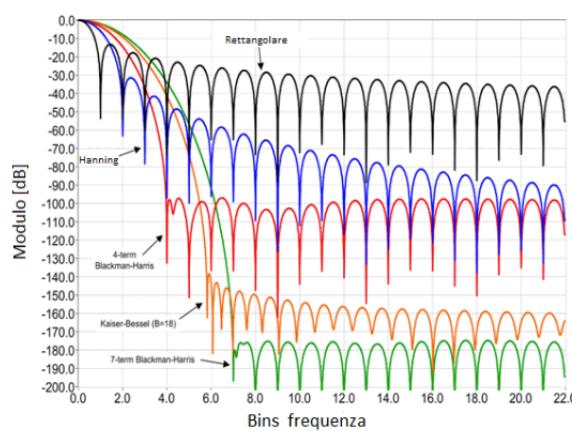


Figura 4.15: Spectral leakage per diversi tipi di finestratura

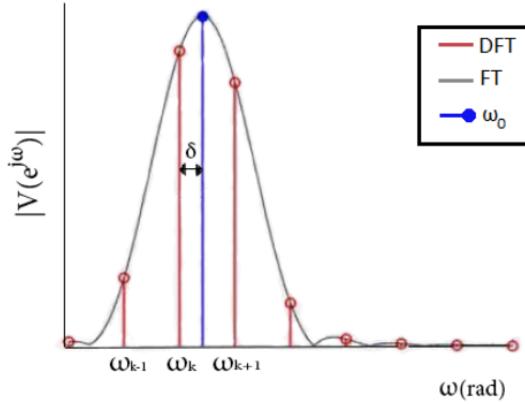


Figura 4.16: Confronto tra DFT e FT di un segnale a frequenza  $\omega_0$

Prima di eseguire l'algoritmo di FFT, quindi, ciascun campione della sequenza  $x(n)$  viene moltiplicato per un coefficiente della funzione peso  $w(n)$  della finestra. In figura 4.15 è mostrato il confronto dello *spectral leakage* per diversi tipi di finestratura. Emerge in modo chiaro dalla figura che l'utilizzo di *smoothing window* riduce notevolmente il fenomeno di dispersione spettrale.

Concludendo, il parallelismo hardware intrinseco che si ottiene con l'FPGA è l'ideale per l'elaborazione numerica dei segnali in parallelo. Per tale motivo si è scelto di eseguire l'algoritmo di FFT su FPGA.

#### 4.3.2 Interpolated Fast Fourier Transform (IFFT)

Come ampiamente descritto nel paragrafo precedente, per mezzo dell'algoritmo di FFT è possibile ricavare lo spettro di frequenza di un segnale. Lo spettro di frequenza ricavato è però discreto, pertanto le frequenze delle singoli componenti spettrali possono essere valutate dalla loro posizione nella spettro discreto con una risoluzione che dipende dal numero dei campioni [10].

Se si campiona il segnale di ingresso con una frequenza  $f_s$ , si ottiene una risoluzione pari a:

$$\Delta f = \frac{f_s}{N} \quad (4.11)$$

dove  $N$  è il numero dei campioni e  $\Delta f$  è la distanza tra 2 *bin* consecutivi.

Per migliorare la risoluzione del tono principale del segnale esistono algoritmi che utilizzano metodi di interpolazione.

Questi algoritmi, chiamati *Interpolated FFT (IFTT) algorithm*, permettono di trovare la correzione di frequenza delta intorno alla frequenza del tono fondamentale allo scopo di ottenere la frequenza del segnale  $\omega_0$  in modo più preciso:

$$\omega_0 = (k \pm \delta) \frac{2\pi f_s}{N} \quad (4.12)$$

con:

$$0 < \delta \leq 0.5$$

Gli algoritmi di interpolazione si dividono tra metodi a 2 punti e a 3 punti. Si è scelto di utilizzare algoritmi a due punti perchè essi limitano il contributo di rumore rispetto al caso a 3 punti.

Per migliorare ulteriormente l'accuratezza dell'estrazione del tono è stato necessario utilizzare un algoritmo di interpolazione dei moduli specifico per segnali a cui è stata applicata la finestra di *Hanning* [9]. Pertanto, si mostra ora il procedimento per il calcolo del fattore di correzione  $\delta$  nel caso di finestra di *Hanning*.

Si consideri un segnale sinusoidale di frequenza  $f_0$  campionato con periodo  $T_s = \frac{1}{f_s}$ :

$$x(nT_s) = A_0 \sin(2\pi f_0 nT_s + \phi) \quad n = 0, 1, \dots, N - 1 \quad (4.13)$$

Considerando la risoluzione  $\Delta f$  dell'equazione 4.11, possiamo riscrivere il tono fondamentale del segnale in funzione della dimensione del singolo *bin*:

$$f_0 = (L + \delta)\Delta f = \gamma\Delta f \quad (4.14)$$

dove  $L$  indica la parte intera di  $\gamma$  e  $\delta$  la correzione frazionaria di frequenza. Perciò, l'equazione 4.13 si può riscrivere come:

$$x[n] = A_0 \sin(2\pi \gamma \frac{n}{N} + \phi) \quad n = 0, 1, \dots, N - 1 \quad (4.15)$$

La DFT dell'equazione precedente alla linea spettrale  $k$  è espressa dalla relazione:

$$x[k] = \frac{1}{2} \left\{ A_0 e^{j\phi} W[(\gamma - k)\Delta f] + A_0 e^{-j\phi} W[(\gamma + k)\Delta f] \right\} \quad (4.16)$$

dove  $W(f)$  è lo spettro di frequenza della funzione di finestratura utilizzata.

Come anticipato, la finestratura utilizzata è quella di *Hanning* definita nel tempo come:

$$w_{n,H} \begin{cases} 0.5 - 0.5 \cos\left(\frac{2\pi}{N}n\right) & 0 \leq n \leq N \\ 0 & 0 > n \geq N \end{cases} \quad (4.17)$$

Lo spettro della finestra di *Hanning* è espresso dalla relazione:

$$W_H(f) = \frac{1}{2} \left\{ W_R(f) - \frac{1}{2} [W_R(f + f_0) + W_R(f - f_0)] \right\} \quad (4.18)$$

dove  $W_R(f)$  è lo spettro di frequenza della finestra rettangolare:

$$W_R(f) = \frac{\sin\left(\pi \frac{f}{\Delta f}\right)}{\sin\left(\pi \frac{f}{N \Delta f}\right)} e^{j\pi \frac{(N-1)f}{N \Delta f}} \quad (4.19)$$

Considerando ora il calcolo della FFT sul segnale a cui è stata applicata la finestra di *Hanning*: il risultato è la sequenza  $X_h(n)$ .

Determinato lo spettro  $X_h(n)$ , l'elaborazione continua calcolandone il modulo. È sufficiente calcolare il modulo di solo una metà della sequenza  $X_h(n)$ , poiché l'altra parte non aggiunge informazioni ulteriori.

Successivamente, si esegue la ricerca del valore massimo presente nel vettore  $|X_h(n)|$  dei moduli calcolato al passo precedente: una volta individuato, si memorizza tale valore e la relativa posizione all'interno del vettore.

Se ipotizziamo che il bin di ampiezza maggiore sia  $|X_h(k)| = |V_k|$  con  $k$  posizione relativa al valore massimo presente nel vettore, mentre  $|X_h(k+1)| = |V_{k+1}|$  sia quello con la seconda ampiezza più grande, possiamo stimare la correzione frazionaria con la formula:

$$\delta = \frac{2|V_{k+1}| - |V_k|}{|V_{k+1}| + |V_k|} \quad (4.20)$$

Per i passi che hanno portato alla stima della correzione  $\delta$  si rimanda alla letteratura [9].

Le finestre utilizzate nel lavoro di tesi sono quella di *Hanning* e quella rettangolare. Per tale motivo di seguito viene mostrato il fattore di correzione nel caso di finestra rettangolare.

Nel tempo la funzione che rappresenta la finestra rettangolare è definita con la relazione:

$$w_{n,R} \begin{cases} 1 & 0 \leq n \leq N \\ 0 & 0 > n \geq N \end{cases} \quad (4.21)$$

e il fattore correttivo di frequenza  $\delta$  è pari a:

$$\delta = \frac{|V_{k+1}|}{|V_{k+1}| + |V_k|} \quad (4.22)$$

Infine, per calcolare la frequenza del tono fondamentale è sufficiente usare la relazione:

$$f_0 = (k + \delta) \Delta f \quad (4.23)$$

In questo modo, il valore del tono estratto viene scorrelato dalla larghezza del singolo *bin*, riducendo notevolmente la risoluzione in frequenza.

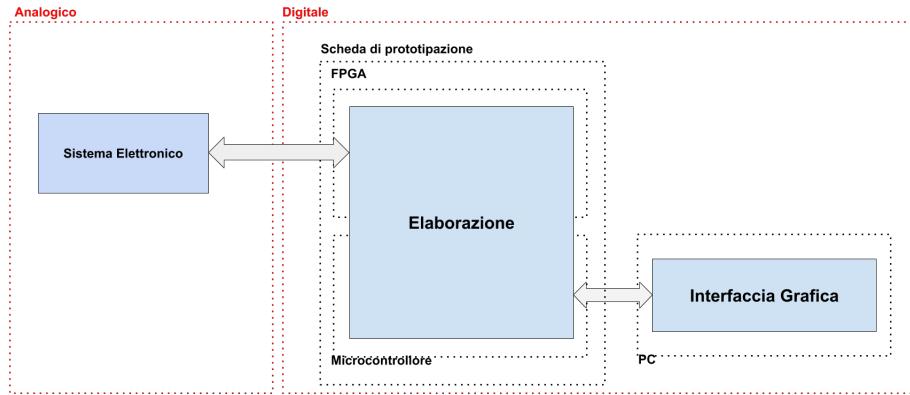


Figura 4.17: Schema a blocchi ad alto livello dell'architettura

#### 4.3.3 Calcolo della distanza assoluta

Come descritto in precedenza nel Capitolo 2, la distanza assoluta  $s$  del bersaglio si ricava con la relazione:

$$s = \frac{f_{rise} + f_{fall}}{2} \left[ \frac{\lambda^2}{2 \left( \frac{\Delta I}{\Delta T} \frac{\Delta \lambda}{\Delta I} \right)} \right] \quad (4.24)$$

dove  $f_{rise}$  e  $f_{fall}$  sono rispettivamente il tono fondamentale del segnale interferometrico nella rampa di salita e discesa dell'onda triangolare.

### 4.4 Architettura Software

L'obiettivo principale del lavoro di tesi è lo sviluppo di un firmware, per la scheda di prototipazione, che sia in grado di computare e mostrare all'utente finale le misure di distanza assoluta prodotte da un laser mediante la tecnica di interferometria a retroiniezione.

La computazione è completamente svolta sulla scheda di prototipazione, in parte su FPGA ed in parte su microprocessore. Mentre, è presente un computer connesso alla scheda che ha come unico scopo quello di mostrare i risultati dell'elaborazione mediante un interfaccia grafica. In figura 4.17 è mostrato uno schema a blocchi ad alto livello dell'architettura appena descritta.

In particolare, l'elaborazione consiste, inizialmente, nell'acquisizione del segnale interferometrico in arrivo dal sistema elettronico del laser. Una volta acquisito, il segnale interferometrico viene condizionato e se ne estraе il tono

fondamentale sfruttando l'algoritmo di FFT. Infine, utilizzando l'informazione del tono fondamentale del segnale, si ricava la misura di distanza assoluta che viene poi mostrata all'utente finale mediante l'interfaccia grafica.

Nei paragrafi successivi verranno descritte dettagliatamente le funzionalità svolte dall'FPGA e dal microcontrollore e come esse sono implementate via software.

#### 4.4.1 FPGA

Le funzionalità svolte in hardware dal FPGA sono:

- Generazione dei segnali di clock per la scheda di conversione
- Generazione del segnale di modulazione
- Campionamento del segnale interferometrico
- Condizionamento del segnale interferometrico
- Calcolo della Fast Fourier Transform (FFT)
- Estrazione del massimo bin di frequenza

**Generazione dei segnali di clock per la scheda di conversione** La scheda di conversione *SCO Board* collegata ai pin digitali della board di prototipazione possiede due canali di conversione: un canale di conversione Analogico/Digitale collegato ai pin digitali d'ingresso e uno di conversione Digitale/Analogico collegato ai pin digitali d'uscita della *sbRIO*. Entrambi i canali di conversione per funzionare correttamente vengono politati, dall'FPGA, con un segnale di clock a una frequenza di  $30MHz$ . Per garantire la sincronizzazione dei due canali di conversione viene utilizzato lo stesso segnale di clock per entrambi i circuiti di conversione.

**Generazione del segnale di modulazione** Il segnale di modulazione del laser viene generato, dai 12 pin digitali d'uscita, sul fronte di salita del segnale di clock. Il segnale in uscita, generato punto per punto ad ogni ciclo di clock, è composto da 1250 punti. Ogni punto è un dato binario da 12 bit con una dinamica da 0 a 4095 livelli di tensione.

Il segnale di modulazione viene riprodotto completamente in uscita ogni 1250 punti; questo fa sì che il laser venga pilotato da un segnale di modulazione con frequenza pari a:

$$f_{mod} = \frac{30MHz}{1250} = 24KHz \quad (4.25)$$

**Campionamento del segnale interferometrico** Il segnale interferometrico in arrivo dal laser viene campionato, dai pin digitali d'ingresso, con una frequenza pari alla frequenza di clock ( $30MHz$ ) e una risoluzione di 12 bit.

Il segnale d'ingresso è anch'esso composto da 1250 punti ognuno con una dinamica da 0 a 4095 livelli di tensione. Pertanto un segnale interferometrico completo equivalente a un periodo di modulazione del laser è disponibile ogni  $41.6us$  ( $24KHz$ ).

**Condizionamento del segnale interferometrico** Il segnale interferometrico, acquisito dal circuito di conversione Analogico/Digitale, viene condizionato prima di passato alla fase di elaborazione vera e propria.

Le operazioni svolte nella fase di condizionamento sono:

- Sottrazione del residuo
- Estrazione di 512 campioni
- Finestratura

**Sottrazione del residuo** La sottrazione del residuo è un'operazione che consiste nel sottrarre punto per punto al segnale interferometrico completo un segnale di 1250 punti chiamato *residuo*. Il *residuo* è un segnale che viene calcolato e conservato nel FPGA nelle fasi preliminari di avviamento dello strumento di misura, il significato di tale segnale verrà illustrato nel Capitolo 5.

**Estrazione di 512 campioni** Il segnale interferometrico, dopo essere stato privato del suo residuo, viene suddiviso in 2 semiperiodi da 625 campioni. Per ciascuno dei due semiperiodi vengono estratti solamente 512 campioni contigui. I restanti 113 campioni vengono scartati.

Il motivo che ha spinto all'estrazione di soli 512 campioni è dovuto all'algoritmo di FFT. Come spiegato in precedenza, tale algoritmo raggiunge ottime prestazioni con segnali che hanno lunghezza pari a una potenza di 2.

**Finestratura** Ciascuno dei due sottoinsiemi da 512 campioni viene poi moltiplicato punto per punto per una funzione finestra da 512 punti a scelta tra quella rettangolare e quella di Hanning. La funzione finestra da utilizzare viene configurata in fase di inizializzazione dello strumento.

**Calcolo della Fast Fourier Transform (FFT)** La penultima funzionalità eseguita dal FPGA è il calcolo della Trasformata Discreta di Fourier (DFT) mediante l'algoritmo di FFT.

Il calcolo dell'FFT viene eseguito su ciascuno dei 2 sottoinsiemi da 512 punti producendo in uscita, per ciascuno dei due sottoinsiemi, lo spettro di frequenza. In particolare, il risultato dell'algoritmo di FFT è composto da due array di 512 punti contenenti rispettivamente la parte reale e immaginaria dello spettro di potenza del segnale nel dominio del tempo.

Dalla parte reale e immaginaria si ricava solamente, per ragioni dipendenti dalle performance del FPGA, il modulo quadrato dello spettro di frequenza.

Dato che i campioni in ingresso all'algoritmo di FFT sono campionati a  $30MHz$ , la risoluzione di frequenza di ciascun *bin* è pari a:

$$f_{bin} = \frac{f_{sample}}{n_{sample}} = \frac{30MHz}{512} = 58.59375KHz \quad (4.26)$$

Ciò significa che ciascun *bin* di frequenza rappresenta la quantità totale di energia al quadrato posseduta dal segnale a quella particolare frequenza. Il primo punto corrisponde a  $0Hz$  (componente continua), il secondo punto corrisponde a  $58.59375KHz$ , il terzo punto corrisponde a  $117.1875KHz$  e così via.

Sappiamo dal teorema di *Nyquist* che il campionamento a  $30MS/s$  sarà in grado di misurare frequenze fino a un massimo di  $15MHz$ . Perciò, tutti i bin maggiori al  $257^{\circ}$  (pari alla frequenza  $256 * 58.59375KHz = 15MHz$ ) rappresentano frequenze negative: il  $258^{\circ}$  corrisponde alla frequenza  $-14.9414062MHz$ , il  $259^{\circ}$  corrisponde a  $-14.8828124MHz$  e così via.

**Estrazione del massimo bin di frequenza** L'ultima fase dell'elaborazione numerica del segnale interferometrico su FPGA è l'estrazione del bin di frequenza con massima ampiezza.

L'algoritmo per l'estrazione del massimo bin viene applicato solo su una parte dei 512 bin di frequenza, questo perché il segnale interferometrico è un segnale puramente reale che non ha componenti immaginarie, e pertanto produce un FFT che è simmetrica rispetto alla componente continua  $0Hz$ . Ciò significa che i valori a frequenze negative sono esattamente gli stessi delle loro controparti positive, e di conseguenza questi punti vengono considerati ridondanti e quindi scartati.

Inoltre, eliminando i primi campioni dello spettro si filtra la potenza della continua, poiché altrimenti risulterebbe il tono più alto.

A questo punto, l'algoritmo di estrazione del massimo bin valuta il campione con ampiezza maggiore, corrispondente al tono fondamentale, ed estrae la sua posizione e la sua ampiezza, l'ampiezza del tono successivo e quella del tono precedente.

L'algoritmo effettua l'estrazione del massimo bin dopo ogni semiperiodo del segnale di modulazione. Ciò indica che la frequenza con cui viene prodotto un risultato è  $48\text{KHz}$ .

Infine, i risultati dell'elaborazione vengono inviati al microprocessore.

### Implementazione software

Il software eseguito dall'FPGA è implementato su un unico VI, come richiesto da LabVIEW; infatti, a differenza di PC e microcontrollori, una FPGA che esegue codice LabVIEW può avere un solo VI in esecuzione simultanea.

Il VI sviluppato si compone di una *Sequence Structure*, composta da 2 frame. La *Sequence Structure* è un costrutto di LabVIEW utilizzato per impostare vincoli di precedenza temporale nell'esecuzione del codice. Questa scelta è stata fatta per garantire il corretto ordine di esecuzione delle istruzioni del codice, con il primo *frame* che si occupa dell'inizializzazione dello strumento e il secondo e ultimo *frame* che esegue le funzionalità precedentemente descritte.

Il *frame* di inizializzazione si occupa di abilitare e inizializzare i pin digitali (DIO pin, *Digital Input Output pin*) della scheda e di predisporre le memorie dell'FPGA per l'esecuzione. In particolare vengono inizializzate le memorie che conservano il segnale di modulazione, della funzione finestra e del residuo.

Il *frame* finale è stato diviso in blocchi funzionali, ognuno contenente una porzione di funzionalità che è necessario garantire. I blocchi funzionali realizzati sono descritti di seguito:

- Generazione del clock, del segnale di modulazione e acquisizione del segnale interferometrico: In questo blocco funzionale è implementata una macchina a stati finiti, composta da due stati: il primo attiva alto il segnale di clock per i circuiti di conversione DAC e ADC ed invia ai pin digitali di uscita il valore del punto del segnale di modulazione da generare. Nel secondo stato, invece, il segnale di clock per i circuiti di conversione viene portato basso e si legge il valore digitale presente ai pin d'ingresso della scheda che corrisponde al valore letto dal circuito elettronico del laser in quell'istante. In questo modo viene garantita una sincronizzazione tra il valore di modulazione e il valore di risposta del laser.

- Lettura dalla memoria del segnale di modulazione: Questo blocco funzionale provvede a leggere dalla memoria, precedentemente inizializzata, il punto successivo del segnale di modulazione usato per pilotare il laser. Ogni punto letto viene inviato al blocco di generazione del segnale di modulazione.
- Condizionamento del segnale: Questo blocco funzionale riceve il segnale dal blocco di acquisizione e esegue le operazioni di condizionamento. Infine, invia il segnale condizionato, punto per punto, al blocco di computazione dell'FFT.
- Computazione FFT: In questo blocco si esegue l'algoritmo di FFT sulla porzione di segnale precedentemente condizionata. L'algoritmo di FFT viene eseguito in pipeline con una latenza di 1316 cicli di clock. Il risultato dell'algoritmo viene inviato, punto per punto, al blocco finale.
- Calcolo del massimo bin dell'FFT: Per ogni risultato generato dal blocco di computazione dell'FFT si verifica se quest'ultimo è quello di ampiezza massima finora rilevato e si provvede a salvare in dei registri il numero del bin con la relativa ampiezza, oltre all'ampiezza precedente e successiva. Questi valori vengono poi inviati al microprocessore per poter eseguire l'FFT interpolata e ricavare il tono fondamentale dal segnale.

Ogni blocco funzionale è rappresentato graficamente da un ciclo. I cicli vengono eseguiti fisicamente in parallelo, per via delle caratteristiche intrinseche dell'esecuzione su FPGA, e la comunicazione inter-processo avviene utilizzando delle code dati chiamate *FIFO*.

In Figura 4.18 è mostrata una versione semplificata del diagramma a blocchi utilizzato per implementare le funzionalità FPGA soprascritte. Il primo ciclo possiede una frequenza di  $60MHz$  per garantire la corretta generazione del clock a  $30MHz$  per i circuiti di conversione A/D e D/A, al contrario dei restanti cicli che hanno frequenza  $30MHz$ .

**Design pattern utilizzati** Nello sviluppo del codice eseguito in hardware dall'FPGA ci si è serviti di due *design pattern* tipicamente utilizzati durante lo sviluppo di codice LabVIEW: il *Producer-Consumer* e il *4-Wire Handshake*, che verranno spiegati in dettaglio nel seguito del paragrafo.

**Producer-consumer** Il *Producer-Consumer design pattern* è un pattern utilizzato per la sincronizzazione tra processi basato sul Master/Slave

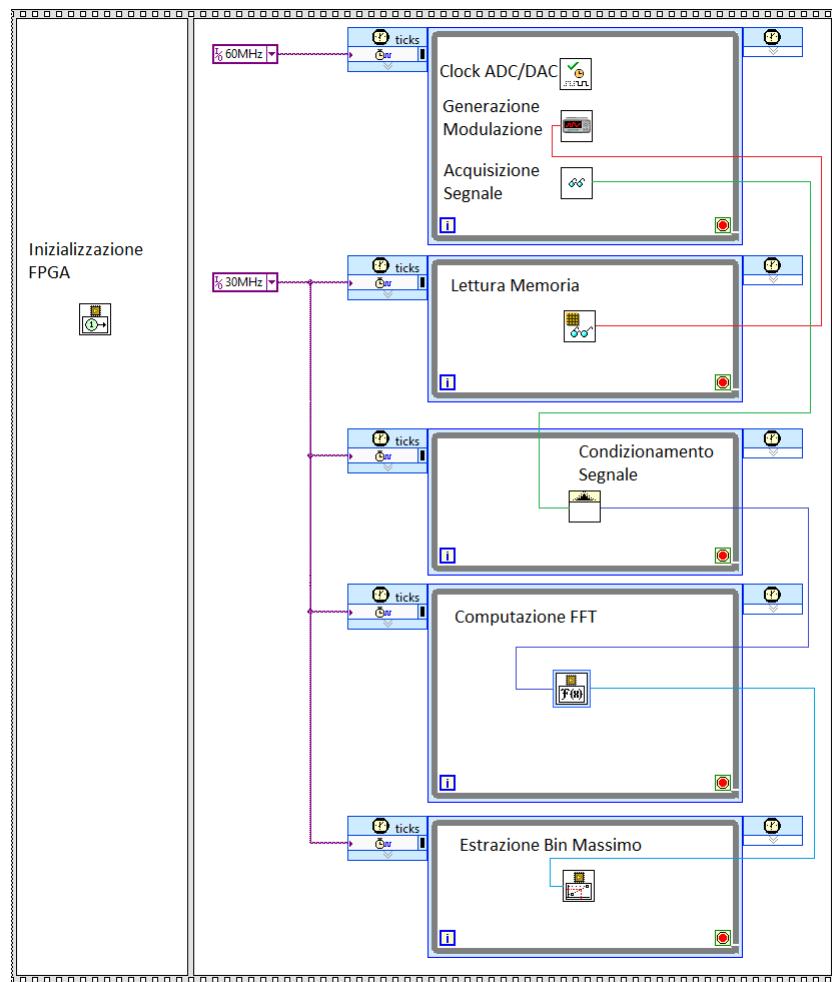


Figura 4.18: Block Diagram semplificato del VI eseguito su FPGA

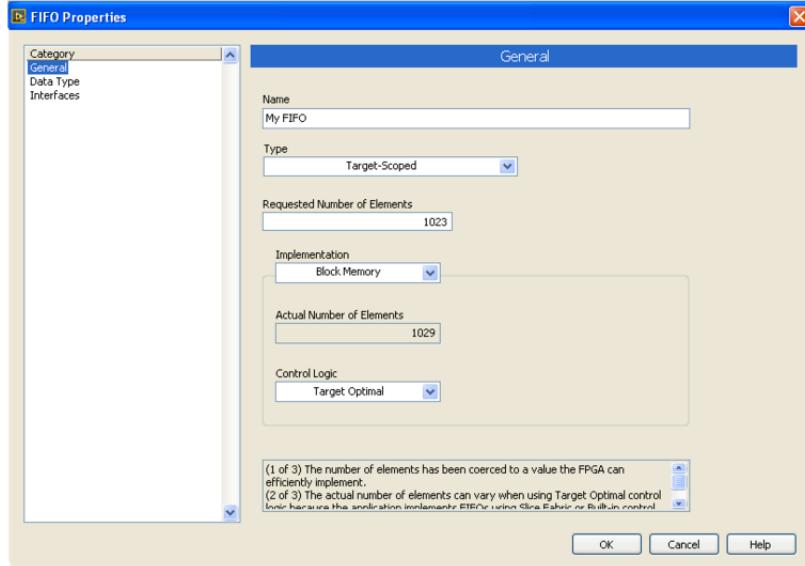


Figura 4.19: Pannello di configurazione per la creazione di una FIFO

pattern. Il problema descrive due processi, uno produttore (*Producer, Master*) ed uno consumatore (*Consumer, Slave*), che condividono una memoria comune, chiamata buffer, di dimensione fissata.

Il compito del produttore è generare dati e depositarli nel buffer. Contemporaneamente, il consumatore utilizzerà i dati prodotti, rimuovendoli di volta in volta dal buffer. Il problema è assicurare che il produttore non elabori nuovi dati se il buffer è pieno, e che il consumatore non cerchi dati se il buffer è vuoto.

In LabVIEW i processi producer e consumer sono rappresentati rispettivamente da due cicli (*While* o *Timed-Loop*) eseguiti in parallelo, mentre il buffer è rappresentato da una memoria *FIFO*.

La *FIFO* è una struttura dati che contiene gli elementi nell'ordine in cui sono ricevuti e fornisce l'accesso a quest'ultimi usando un criterio *First-In First-Out*. Quando si configura una FIFO, è necessario specificarne la dimensione e la tipologia di dato (Figura 4.19).

La Figura 4.20 seguente mostra un esempio in LabVIEW di Producer/Consumer pattern utilizzato per trasferire dati tra due cicli Single Timed Loop sulla stessa FPGA. I due processi risiedono su due diversi domini di frequenza: il producer utilizza un clock a  $60MHz$  mentre il consumer utilizza un clock a  $30MHz$ .

Nel processo producer, le letture relative al numero dell'iterazione del

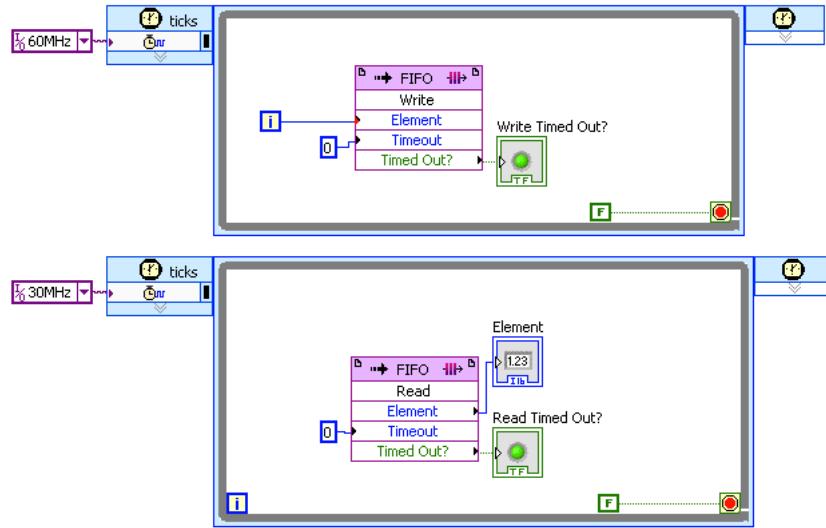


Figura 4.20: Esempio di Producer/Consumer implementato su LabVIEW FPGA

ciclo vengono scritte nel buffer di memoria FIFO attraverso il nodo *FIFO Method Node Write*. Se il dato numerico in ingresso al metodo non è disponibile o la coda è piena, il metodo *Write* non scrive alcun dato nella coda e asserisce l'uscita *Timed Out?* a vero segnalando che l'elemento non è stato depositato nella coda. Il metodo *FIFO Method Node Read*, nel processo consumer, legge i dati dalla coda. Se i dati non sono disponibili per la lettura asserisce l'uscita *Timed Out?* a vero segnalando che la coda è vuota.

**4-Wire Handshake** L'implementazione *pipelined* dell'*express VI* FFT ha richiesto l'uso del pattern del *4-wire handshake* [1].

Il *4-wire handshake* è un pattern tipicamente usato nell'implementazione di sub-VI eseguiti in pipeline nell'ambito di LabVIEW FPGA.

Nei nodi che consentono l'utilizzo di questo pattern tipicamente si hanno due ingressi e due uscite aggiuntive, che consentono l'implementazione del protocollo:

- Ingressi

- *Input Valid*: specifica che il prossimo dato da processare è pronto in ingresso
- *Ready for output*: Specifica se i nodi a valle sono pronti per ricevere nuovi dati in ingresso

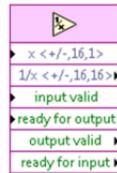


Figura 4.21: Ingressi ed uscite di un express VI con esecuzione pipelined

- Uscite

- *Output valid*: specifica ai nodi a valle che il dato corrente in uscita è valido
- *Ready for input*: specifica se il subVI può ricevere nuovi dati in ingresso al prossimo ciclo

L'esecuzione del protocollo *4-Wire Handshake* inizia con il nodo pipelined che comunica alla coda in ingresso che è pronto per accettare nuovi dati in ingresso. Tipicamente è utilizzato un feedback node per comunicare il valore logico della variabile *ready for input* ai nodi a valle del subVI.

Quando il nodo a valle del subVI fornisce il dato assegna il valore logico Vero alla variabile *Input Valid* indicando così al subVI che il dato al suo ingresso è pronto per essere processato. Se il nodo pipelined ha ricevuto tutti i dati di cui necessitava il valore logico della variabile *Ready for input* viene portato a Falso.

Al termine della computazione il subVI attende che il valore della variabile *ready for output* sia Vero e modifica il valore logico della variabile *output valid* a Vero, per comunicare ai nodi a monte che il valore in output è valido. Contemporaneamente il valore della variabile *ready for input* viene portato a Vero, per poter accettare un nuovo valore in ingresso e ricominciare la computazione.

Nel caso del subVI dell'FFT il valore di *input ready* è dato dal *Time out?* della coda FIFO *Waveform*; infatti, se la coda contiene almeno un dato al suo interno, il dato che fornisce è valido. Al termine della computazione dell'FFT il valore *output valid* consente al calcolo del modulo al quadrato dell'FFT di essere eseguito, e, come specificato dal pattern, il valore logico di *ready for input* viene portato a Vero.

Poiché il calcolo del modulo quadrato dell'FFT non è stato sviluppato sfruttando la pipeline, esso è sempre pronto a ricevere un input, rendendo il valore di *ready for output* sempre Vero.

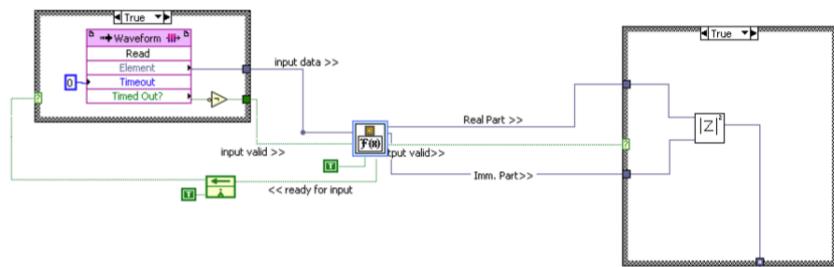


Figura 4.22: Computazione FFT

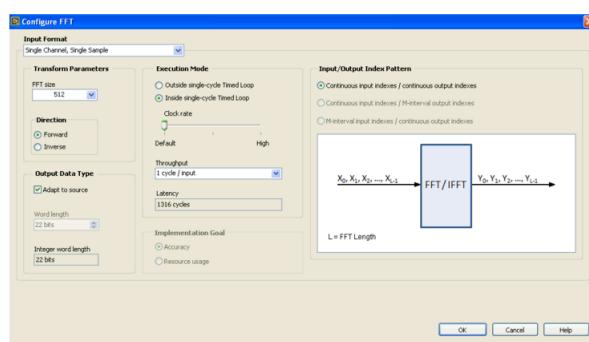


Figura 4.23: Pannello di configurazione del FFT Express VI

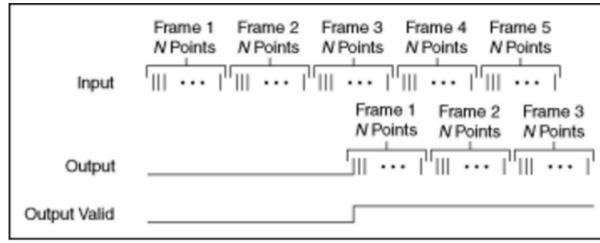


Figura 4.24: FFT Express VI Timing Diagram

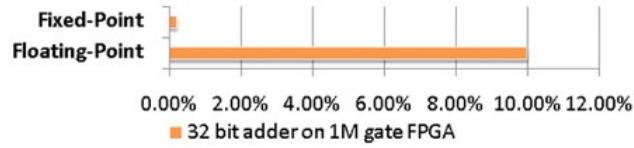


Figura 4.25: Confronto dello spazio occupato da un sommatore su FPGA con Fixed Point e Floating Point

Il SubVI *FFT Express VI*, che effettua la computazione del FFT, è inserito all'interno di un *Single Cycle Timed Loop*. Esso effettua il calcolo dell'FFT di 512 campioni con una latenza di 1316 cicli di clock ( $\frac{1316}{30MHz} = 43.86us$ ) e un *throughput* di ingresso pari a 1 *input/ciclo*. In Figura 4.23 è mostrato il pannello di configurazione dell'*FFT Express VI*.

Il *Timing diagram* in Figura 4.24 mostra le relazioni temporali tra gli ingressi e le uscite dell'FFT Express VI quando esso si trova all'interno di un single-cycle Timed Loop con Throughput pari 1 input/ciclo.

**Fixed Point** Per motivi di prestazioni e di area, è stato ampiamente utilizzato il tipo di dato *fixed point* nello sviluppo del codice FPGA. Per i precedenti motivi, inoltre, le funzioni di libreria presenti in LabVIEW FPGA accettano come ingressi numeri rappresentati in *fixed point*.

Come specificato dalla documentazione di LabVIEW [4], l'utilizzo di algebra *fixed point* permette di avere un risparmio notevole in termini di area utilizzata su FPGA rispetto all'uso del *floating point*, come è visibile in Figura 4.25.

Il tipo di dato *fixed point* è caratterizzato da tre parametri: la lunghezza in numero di bit (*word length*), la lunghezza della parte intera (*integer word length*) e la presenza del segno (*signed/unsigned*). Inoltre è possibile scegliere se implementare un ulteriore bit in grado di segnalare la presenza di overflow e di scegliere il comportamento in caso di overflow.

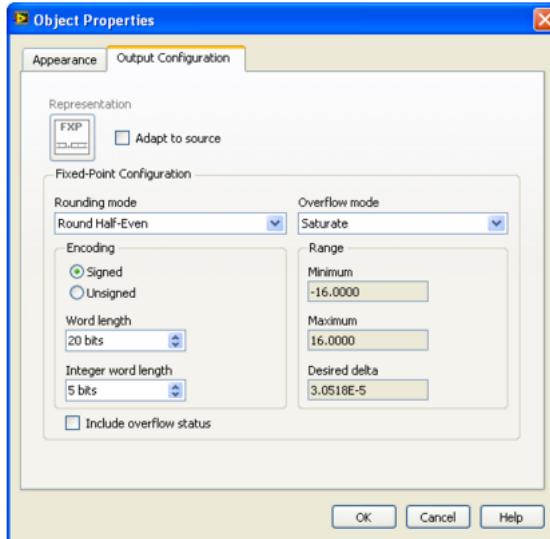


Figura 4.26: Fixed point in LabVIEW

La lunghezza della parte intera può essere maggiore o minore rispetto alla lunghezza della parola. Nel primo caso non è possibile rappresentare numeri con una parte decimale, mentre nel secondo caso i numeri avranno una parte decimale con una lunghezza in bit pari alla differenza tra la lunghezza della parola e della parte intera.

È anche possibile rappresentare numeri con parte intera negativa. In questo caso non è possibile rappresentare numeri interi, ed in particolare si potranno rappresentare soltanto numeri compresi tra 0 e 1 in caso di *fixed point unsigned*, oppure tra  $-1$  e 1 in caso di *fixed point signed*.

Per lo sviluppo del progetto sono stati utilizzati fixed point signed a 12-bit con parola intera di 12 bit per i segnali generati ed acquisiti. Questa scelta è stata fatta tenendo conto della risoluzione dei convertitori utilizzati, che è appunto di 12-bit.

Per quanto riguarda le restanti configurazioni del tipo di dato all'interno del codice si è fatto affidamento al sistema di auto-adattamento delle uscite in base alla lunghezza dei risultati integrato in LabVIEW. Questo sistema calcola il massimo valore rappresentabile in uscita da un blocco funzionale e adatta il tipo di dato di conseguenza.

#### 4.4.2 Microcontrollore

Il software sviluppato per l'esecuzione su microcontrollore svolge le funzionalità che, per via della struttura intrinseca e dei limiti di spazio dell'FPGA,

non è stato possibile implementare direttamente in hardware. In particolare il software eseguito sul microcontrollore fornisce le seguenti funzionalità:

- Algoritmo per il controllo del rumore
- Calcolo della Fast Fourier Transform interpolata (IFFT)
- Calcolo delle frequenze di frangia
- Calcolo della distanza assoluta

**Algoritmo per il controllo del rumore** A causa di possibili disturbi esterni, non tutte le misure sono da ritenere valide. È stato quindi implementato un algoritmo di controllo che scartasse le ampiezze dei bin qualora fossero palesemente errate.

Per fare ciò si è usato un semplice criterio di selezione: se le ampiezze dei bin massimo e laterali sono al di sotto di una determinata soglia, chiamata "soglia di rumore", vengono scartate altrimenti accettate. Le ampiezze vengono accettate se e solo se tutte le ampiezze di un periodo di modulazione superano positivamente il criterio di selezione.

La "soglia di rumore" è la massima ampiezza raggiungibile da un bin di frequenza in assenza di ostacolo da misurare. Essa viene calcolata rimuovendo l'ostacolo e non considerando le ampiezze dei primi bin.

**Calcolo della Fast Fourier Transform interpolata (IFFT)** La funzionalità più importante eseguita dal microprocessore è il calcolo dell'FFT interpolata (IFFT).

I dati, ricevuti del FPGA, relativi alle ampiezze dei bin massimo e laterali sono espressi come modulo al quadrato. Per permettere la corretta esecuzione dell'algoritmo di IFFT è necessario, però, che le ampiezze siano espresse in modulo, per questo motivo viene applicata l'operazione di radice quadrata per convertirli. Ultimata l'operazione di radice quadrata, viene calcolato, per ogni FFT computata dal FPGA, il fattore di correzione della frequenza.

Se la funzione finestra utilizzata è la rettangolare viene applicata l'equazione:

$$\delta = \frac{|V_x|}{|V_k| + |V_x|} \quad (4.27)$$

dove:

- $k$  è il bin di frequenza di ampiezza massima
- $|V_k|$  è il modulo del bin di ampiezza massima

- $|V_{k-1}|$  è il modulo del bin precedente al massimo
- $|V_{k+1}|$  è il modulo del bin successivo al massimo
- $|V_x|$  è il massimo tra  $|V_{k-1}|$  e  $|V_{k+1}|$

Altrimenti, se la funzione finestra utilizzata è *Hanning* viene applicata l'equazione:

$$\delta = \frac{2|V_x| - |V_k|}{|V_k| + |V_x|} \quad (4.28)$$

Infine, la correzione di frequenza viene aggiunta al massimo bin  $k$  calcolato dal FPGA ottenendo così il bin interpolato:

$$k' = k + \delta \quad (4.29)$$

con  $0 < \delta \leq 0.5$ .

**Calcolo delle frequenze di frangia** La frequenza del segnale, in funzione del bin interpolato, si calcola con la relazione:

$$f_0 = \frac{f_{sample}}{N} k' = \frac{30MHz}{512} k' \quad (4.30)$$

Per preparare il sistema ad eseguire la misura di distanza assoluta è necessario analizzare separatamente i dati corrispondenti al semiperiodo di discesa e al semiperiodo di salita del segnale triangolare di modulazione, in modo da estrarre separatamente i toni fondamentali  $f_{fall}$  e  $f_{rise}$ .

**Calcolo della distanza assoluta** L'ultima operazione compiuta dal microcontrollore è il calcolo della distanza assoluta mediante la relazione:

$$s = \frac{f_{rise} + f_{fall}}{2} \left[ \frac{\lambda^2}{2 \left( \frac{\Delta T}{\Delta T} \frac{\Delta \lambda}{\Delta \lambda} \right)} \right] = \frac{f_{rise} + f_{fall}}{2} \Psi \quad (4.31)$$

Il parametro  $\Psi = \frac{\lambda^2}{2 \left( \frac{\Delta T}{\Delta T} \frac{\Delta \lambda}{\Delta \lambda} \right)}$  non è definito, questo perché  $\frac{\Delta \lambda}{\Delta T}$  non è noto a priori, ma si può ritenere in prima approssimazione costante, e si può ricavare variando la distanza e mantenendo tutti gli altri parametri costanti. Dopo aver mediato i risultati si ottiene  $\Psi = 261.5nm/Hz$ .

Inoltre, è possibile mediare  $M$  misure di distanza ottenendo così una misura più accurata mediante la relazione:

$$\bar{s} = \frac{\sum_{i=1}^M s_i}{M} \quad (4.32)$$

L'introduzione del calcolo della media di  $M$  distanze assolute riduce il *throughput* di misura di un fattore  $M$ .

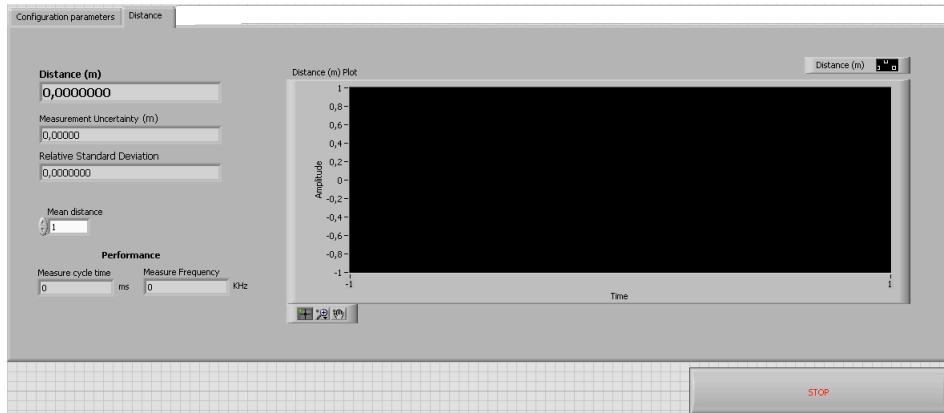


Figura 4.27: Interfaccia grafica dello strumento di misura

### Implementazione software

Il software sviluppato per essere eseguito su microcontrollore è contenuto in un unico VI, nonostante questo non sia un obbligo imposto dal linguaggio. La scelta di utilizzare un singolo VI è stata fatta per semplificare il processo di sviluppo; tuttavia sono stati sviluppati più VI, utilizzati poi come subVI nell'unico VI eseguito dal microcontrollore, per rendere il codice più leggibile e lo sviluppo più modulare.

Il VI principale eseguito dal microcontrollore si compone di una parte di codice sequenziale che svolge l'inizializzazione ed in seguito di due cicli paralleli, uno per leggere i dati inviati dall'FPGA ed uno per elaborare i dati ricevuti. La comunicazione tra i due cicli è fatta utilizzando il già descritto pattern del Producer-Consumer.

La parte di codice iniziale si occupa della configurazione dello strumento, inviando all'FPGA il *bitfile* da eseguire, i parametri di esecuzione ed il contenuto delle memorie. Il ciclo di elaborazione, invece, esegue tutte le funzionalità appena descritte.

L'interfaccia grafica, che corrisponde al front panel del VI implementato su microcontrollore, viene eseguita su un PC collegato con cavo *Ethernet* alla scheda di prototipazione. Il microcontrollore invia, attraverso il protocollo *Ethernet*, i dati da visualizzare graficamente sugli indicatori del front panel.

#### 4.4.3 Comunicazione tra FPGA e Microcontrollore

Lo comunicazione tra il microprocessore e il chip FPGA avviene attraverso l'utilizzo di un componente hardware dedicato chiamato DMA.

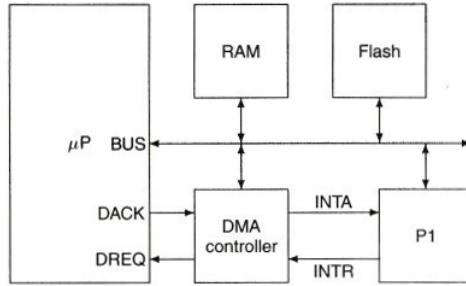


Figura 4.28: Schema architettonale per DMA

Il DMA, acronimo di *Direct Memory Access*, di una CPU è quel meccanismo che permette ad altri sottosistemi, quali ad esempio le periferiche, di accedere direttamente alla memoria interna per scambiare dati, in lettura e/o scrittura, senza coinvolgere l'unità di controllo per ogni byte trasferito tramite l'usuale meccanismo dell'interrupt e la successiva richiesta dell'operazione desiderata, ma generando un singolo interrupt per blocco trasferito [43]. Nella nostra architettura il sottosistema in questione è il chip FPGA.

L'utilizzo del DMA permette al chip FPGA di trasferire grosse quantità di dati sulla memoria del Real-Time host senza l'intervento del microprocessore. Questo meccanismo offre un notevole incremento delle prestazioni del sistema poiché il microprocessore, non essendo impegnato nel trasferimento dei dati, può svolgere altre operazioni.

Di seguito è descritto il funzionamento generale di una DMA. Lo schema generale di funzionamento è mostrato in Figura 4.28.

Il flusso degli eventi che caratterizzano un trasferimento mediante DMA inizia con l'asserzione di un segnale di interrupt (INTR). Il DMA controller, dispositivo incaricato di interfacciarsi con microprocessore e periferiche, rileva la richiesta e richiede al microprocessore, con un segnale di interrupt (DREQ) il controllo del bus dati. Quando il microprocessore è pronto per rilasciare il bus dati lo segnala al DMA controller con un segnale di acknowledgement (DACK). Successivamente, il DMA controller, comunica alla periferica, mediante il segnale di INTA, che la trasmissione dei dati può avere inizio. Il DMA controller trasferisce i dati in arrivo dalla periferica sulla memoria senza l'intervento del microprocessore. Al termine della trasmissione, il DMA controller de-asserisce i segnali di DREQ e INTA e la periferica il segnale di INTER. A sua volta il microprocessore abbassa il segnale di DACK.

In LabVIEW FPGA questo flusso di eventi è nascosto al programma-

tore. L’ambiente di sviluppo mette a disposizione dei nodi *Read DMA* e *Write DMA* che svolgono al loro interno il protocollo di trasmissione appena descritto.



## Capitolo 5

# Misure effettuate e dati sperimentali



# Conclusioni e sviluppi futuri

Il lavoro di questa tesi ha riguardato la progettazione del firmware di un misuratore laser che sfrutta l'interferometria a *self-mixing* per fornire l'informazione di distanza assoluta di un bersaglio fisso.

Questo obiettivo è stato raggiunto in modo soddisfacente, come mostrato dai risultati presentati nel Capitolo 5. Si sono ottenute incertezze relative di misura nell'ordine di  $10^{-4}$ , che, per le distanze considerate, si traducono in errori assoluti dell'ordine dei micrometri. Inoltre le misure effettuate per valutare la linearità dello strumento hanno mostrato che lo strumento finale è lineare nello spostamento all'interno dell'incertezza della misura.

Tuttavia in futuro sono possibili alcuni miglioramenti.

In particolare, per quanto riguarda la frequenza di elaborazione di una singola misura di distanza, non è stato possibile ottenere singoli campioni di distanza a frequenze superiori ai  $50Hz$ , prestazione molto al di sotto delle potenzialità delle tecnologie utilizzate.

Inoltre la tecnica di interferometria a *self-mixing* utilizzata è in grado di fornire una misura della velocità del bersaglio. Questa funzionalità, a causa di vincoli di tempo nello sviluppo del progetto, non è stata adeguatamente testata e si è quindi scelto di escluderla dal lavoro presentato.

Infine si sono verificati alcune anomalie durante le misure effettuate, riconducibili a problemi termici.

Per ottenere prestazioni di misura migliori è necessario utilizzare una piattaforma di prototipazione più moderna, più ampia e con capacità di calcolo maggiore. In particolare, a causa dei limiti di spazio imposti dall'FPGA utilizzata in questo lavoro non è stato possibile effettuare tutta la computazione su di essa. Pertanto è stato necessario implementare su microcontrollore parte del codice necessario al calcolo della misura di distanza.

L'utilizzo di una FPGA più versatile dal punto di vista della capacità di elaborazione permetterebbe di effettuare anche la parte di computazione assegnata al microcontrollore direttamente in hardware, in modo da dimi-

nuire il tempo totale di elaborazione e garantire quindi una frequenza di elaborazione della singola misura più elevata.

Per mantenere immutata la maggior parte del firmware è necessario scegliere un prodotto ufficialmente supportato da NI LabVIEW. Una scelta adeguata potrebbe essere il *System On Module* (SOM), codice prodotto *sbRIO-9651*, progettato dalla stessa *National Instruments*. Questa scheda di sviluppo, oltre a garantire capacità doppie in termini di area integra al suo interno un microcontrollore più veloce ed è appositamente sviluppata per essere utilizzata per la produzione in volumi, garantendo così anche la possibilità di rendere lo strumento commercializzabile senza ulteriori modifiche.

Per quanto riguarda i problemi di deriva termica dello strumento la soluzione migliore sarebbe quella di implementare un controllo di temperatura in grado di mantenere lo strumento in condizioni termiche stabili. Per questa implementazione è necessario sviluppare un controllore di temperatura all'interno del firmware dello strumento, oppure alternativamente è possibile utilizzare un controllore hardware dedicato. La seconda soluzione è preferibile, poiché richiede un minor sforzo in termini di tempo di sviluppo, ed inoltre riduce il tempo di testing della soluzione implementata.

Si potrà inoltre utilizzare lo stesso strumento anche per misure di velocità e vibrometria, con l'implementazione di algoritmi che sfruttano a loro volta l'estrazione dei toni fondamentali del segnale interferometrico. Per fare ciò è necessario, come per la misura di distanza, effettuare una fase approfondita di taratura e implementare algoritmi di controllo dell'affidabilità della misura.

## Appendice A

### Documentazione software



# Bibliografia

- [1] <https://decibel.ni.com/content/docs/doc-12682>.
- [2] <http://sine.ni.com/nips/cds/view/p/lang/it/nid/210421>.
- [3] <http://windriver.com/products/vxworks/>.
- [4] <http://www.ni.com/newsletter/50303/en/>.
- [5] <http://www.ti.com/lit/ds/symlink/ads807.pdf>.
- [6] <http://www.ti.com/lit/ds/symlink/dac902.pdf>.
- [7] <http://www.xilinx.com/products/silicon-devices/fpga/spartan-6/lx.html>.
- [8] Palludo A. and Silvano A. Sistema ottico a retroiniezione per la misura di distanza assoluta. Master's thesis, Politecnico di Milano, 2015.
- [9] D. Agrez. Frequency estimation of the non-stationary signals using interpolated dft. In *Instrumentation and Measurement Technology Conference, 2002. IMTC/2002. Proceedings of the 19th IEEE*, volume 2, pages 925–930 vol.2, 2002.
- [10] G. Andria, M. Savino, and Amerigo Trotta. Windows and interpolation algorithms to improve electrical measurement accuracy. *Instrumentation and Measurement, IEEE Transactions on*, 38(4):856–863, Aug 1989.
- [11] J.M. Birkner and H.T. Chua. Programmable array logic circuit, November 7 1978. US Patent 4,124,899.
- [12] Actel Corporation. Axcelerator family fpga, May 2005.
- [13] Actel Corporation. Proasic3 flash family fpgas, October 2005.

- [14] Philippe Coussy and Adam Morawiec. *High-Level Synthesis: From Algorithm to Digital Circuit*. Springer Publishing Company, Incorporated, 1st edition, 2008.
- [15] S. Disegna. Struttura hardware per interferometria a self-mixing. Master's thesis, Politecnico di Milano, 2015.
- [16] S. Donati. *Electro-Optical Instrumentation: Sensing and Measuring with Lasers*. Prentice Hall, 2003.
- [17] S. Donati, G. Giuliani, and S. Merlo. Laser diode feedback interferometer for measurement of displacements without ambiguity. *Quantum Electronics, IEEE Journal of*, 31(1):113–119, Jan 1995.
- [18] M. Fingeroff. *High-Level Synthesis Blue Book*. Xlibris Corporation, 2010.
- [19] W. Fornaciari and C. Brandoles. *Sistemi embedded. Sviluppo hardware e software per sistemi dedicati*. Pearson Italia, 2007.
- [20] Guido Giuliani, Michele Norgia, Silvano Donati, and Thierry Bosch. Laser diode self-mixing technique for sensing applications. *Journal of Optics A: Pure and Applied Optics*, 4(6):S283, 2002.
- [21] J. Greene, E. Hamdy, and S. Beal. Antifuse field programmable gate arrays. *Proceedings of the IEEE*, 81(7):1042–1056, Jul 1993.
- [22] E. Hamdy, J. McCollum, S.-O. Chen, S. Chiang, S. Eltoukhy, J. Chang, T. Speers, and A. Mohsen. Dielectric based antifuse for logic and memory ics. In *Electron Devices Meeting, 1988. IEDM '88. Technical Digest., International*, pages 786–789, Dec 1988.
- [23] A.N.S. Institute. American national standard for safe use of lasers: Ansi z 136.1. Technical report, Laser Institute of America, 2007.
- [24] Safa O Kasap. *Optoelectronics and Photonics: Principles and Practices*. Pearson Higher Ed, 2012.
- [25] H. Kopetz. *Real-time Systems: Design Principles for Distributed Embedded Applications*. Kluwer Academic Publishers, 1997.
- [26] Ian Kuon and J. Rose. Measuring the gap between fpgas and asics. *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on*, 26(2):203–215, Feb 2007.

- [27] Ian Kuon, Russell Tessier, and Jonathan Rose. Fpga architecture: Survey and challenges. *Found. Trends Electron. Des. Autom.*, 2(2):135–253, February 2008.
- [28] R. Lang and K. Kobayashi. External optical feedback effects on semiconductor injection laser properties. *Quantum Electronics, IEEE Journal of*, 16(3):347–355, Mar 1980.
- [29] R. Lipp, R. Freeman, and T. Saxe. A high density flash memory ppga family. In *Custom Integrated Circuits Conference, 1996., Proceedings of the IEEE 1996*, page 239, May 1996.
- [30] D. Melchionni. Vibrometro laser retroazionato in digitale. Master’s thesis, Politecnico di Milano, 2012.
- [31] Michael Meredith. A look inside behavioral synthesis. *EETimes*, 2004.
- [32] Robert C. Minnick. A survey of microcellular research. *J. ACM*, 14(2):203–241, April 1967.
- [33] P. Pavan, R. Bez, P. Olivo, and E. Zanoni. Flash memory cells—an overview. *Proceedings of the IEEE*, 85(8):1248–1271, Aug 1997.
- [34] E. M. Randone. *Fenomeni di retro-iniezione ottica in laser a semiconduttore e applicazioni*. PhD thesis, Università Degli Studi di Pavia, 2007.
- [35] J. Rose S. D. Brown, R. Francis and Z. Vranesic. *Field-Programmable Gate Arrays*. Kluwer Academic Publishers, 1992.
- [36] Freescale semiconductor. *e300 Power Architecture Core Family Reference Manual*.
- [37] Chih-Ching Shih, R. Lambertson, F. Hawley, F. Issaw, J. McCollum, E. Hamdy, H. Sakurai, H. Yuasa, H. Honda, T. Yamaoka, T. Wada, and Chenming Hu. Characterization and modeling of a highly reliable metal-to-metal antifuse for high-performance and high-density field-programmable gate arrays. In *Reliability Physics Symposium, 1997. 35th Annual Proceedings., IEEE International*, pages 25–33, Apr 1997.
- [38] K.G. Shin and P. Ramanathan. Real-time computing: a new discipline of computer science and engineering. *Proceedings of the IEEE*, 82(1):6–24, Jan 1994.

- [39] D.J. Smith. Vhdl and verilog compared and contrasted-plus modeled example written in vhdl, verilog and c. In *Design Automation Conference Proceedings 1996, 33rd*, pages 771–776, Jun 1996.
- [40] N. Storey. *Electronics: A Systems Approach*. Pearson Education, fourth edition, 2009.
- [41] D. Storti. Misuratore di distanza assoluto basato sulla interferometria a retro-iniezione. Master's thesis, Politecnico di Milano, 2014.
- [42] O. Svelto. *Principles of Lasers*. Plenum Press, fourth edition, 1998.
- [43] A.S. Tanenbaum. *Architettura dei calcolatori. Un approccio strutturale*. Pearson Education, 2006.
- [44] R. H. Freeman H. Hsieh J. Y. Ja J. E. Mahoney L. T. Ngo W. Carter, K. Duong and S. L. Size. A user programmable reconfiguration gate array. in *Proceedings of the IEEE Custom Integrated Circuits Conference*, pages 233–235, May 1986.
- [45] S. E. Wahlstrom. *Programmable logic arrays - cheaper by the millions*. Electronics, December 1967.
- [46] Xilinx. Xc9500 in-system programmable cpld family ds063(v5.4), April 2006.