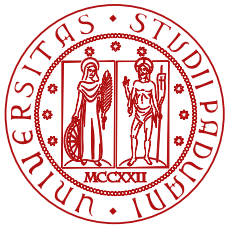

Università degli Studi di Padova

Dipartimento di Ingegneria dell'Informazione

Laurea Triennale in Ingegneria Elettronica

Sviluppo e caratterizzazione di una catena di ritardo elettronico su tecnologia FPGA per un sistema di Quantum Key Distribution

Candidato

Gabriele Orlandi

Matricola 1222371

Relatore

Dr. Andrea Stanco

Università di Padova

Correlatore

Dr. Matías R. Bolaños W.

Università di Padova

Anno Accademico
2022/2023

Sommario

L'obiettivo di questa tesi è la caratterizzazione e ottimizzazione di una delay line elettronica implementata su Field Programmable Gate Array (FPGA) per il modulatore elettro-ottico dello schema iPognac, utilizzato per modulare la polarizzazione di impulsi ottici, impiegato nell' ambito della Quantum Key Distribution. In questa tesi vengono presentati lo studio, la caratterizzazione e la linearizzazione dei tempi di ritardo di una catena di delay sviluppata al fine di rendere facilmente sincronizzabile l'allineamento tra impulso ottico ed elettrico all'interno del modulatore elettro-ottico, con anche l'analisi della loro variazione al variare della temperatura di lavoro. Vengono inoltre presentate le misure condotte in funzione della temperatura del chip FPGA al fine di analizzare e determinare le variazioni in termini di linearità, ritardo e jitter.

Indice

1 Introduzione

Se pensare è equivalente a lavorare, io ho dedicato ad esso quasi tutte le mie ore di veglia. - Nikola Tesla

1.1 Obiettivo della tesi

Alla base di questa tesi vi è la necessità di creare un sistema di ritardo che permetta una calibrazione più efficiente di una sorgente in Quantum Key Disribution (QKD) implementata con il metodo iPognac utilizzato per modulare ad alta velocità la polarizzazione di impulsi ottici. Un cattivo allineamento infatti porterebbe ad un Qubit Error Rate (QBER) maggiore e quindi ad un sistema meno performante.

L'obiettivo di questa tesi è la creazione, caratterizzazione e ottimizzazione di una delay line elettronica implementata su Field Programmable Gate Array (FPGA) per il modulatore elettro-ottico dello schema iPognac, che si integri con quelle già esistenti.

Inizialmente erano presenti due tipi di ritardo nel design FPGA:

- Ritardo di tipo Coarse: È un semplice contatore che permette di ritardare l'impulso elettrico con precisione di 5 ns, pari ad un ciclo di clock della scheda utilizzata.
- Ritardo di tipo Superfine: Permette di ritardare l'impulso con una precisione di 62.5 ps ed è definito da un modulo esterno fornito dalla Xilinx.

Dato che i segnali che si andranno ad utilizzare hanno una larghezza di 5 ns, è fondamentale poter coprire più range: il Coarse ha una risoluzione troppo

grande (serve una risoluzione inferiore ai 5 ns), mentre il Superfine ha un' ottima risoluzione ma ha un range massimo di 2ns. Si rende quindi necessario aggiungere una nuova delay chain, definita "Fine" che permetta di aumentare il range dei ritardi, mantenendoli più lineari possibile. L'introduzione di questa nuova delay chain renderà molto più agevole la sincronizzazione tra l'impulso ottico e il sistema elettronico del modulatore elettro-ottico. In particolare verrà posta molta attenzione sulle differenti modalità che si possono utilizzare per creare la delay chain, partendo dalla più semplice fino alla più complessa (in termini di connessione di blocchi logici) per poi utilizzare quella che permette di avere una linearizzazione dei tempi di ritardo migliore. Ci si è poi concentrati sull'analisi degli effetti che la temperatura ambientale può avere sui tempi di ritardo e sui jitter della delay line, essa infatti si potrà utilizzare anche in condizioni molto diverse da quelle a temperatura ambiente.

La tesi si articola in 5 capitoli: nel primo capitolo viene fornita un'introduzione del sistema che si vuole creare, e le motivazioni per cui esso è utile.

Nel secondo capitolo ci si occupa di spiegare la teoria necessaria alla comprensione della tesi.

Il terzo capitolo si concentra sulla spiegazione degli strumenti utilizzati per permettere l'eventuale ripetizione degli esperimenti svolti.

Nel quarto capitolo si procede alla spiegazione degli esperimenti e alla discussione dei risultati ottenuti, anche mediante l'uso di grafici e schemi delle delay line implementate.

Nel quinto capitolo, infine, si procede a commentare i risultati ottenuti dallanalisi dellelaborazione dei dati, esponendo gli elementi più rilevanti ottenuti dagli esperimenti svolti.

2 Teoria

2.1 Quantum Key Distribution (QKD)

La Quantum Key Distribution (o, in italiano, distribuzione quantistica di chiavi) è un metodo per trasmettere chiavi segrete tra due dispositivi posizionati in un punto A e B (tipicamente chiamati Alice e Bob) su un canale insicuro in modo sicuro [3]. Essa garantisce la sicurezza della comunicazione sfruttando le leggi della meccanica quantistica. Infatti qualsiasi tentativo di un soggetto esterno di intercettare la comunicazione introdurrebbe, secondo le leggi della meccanica quantistica, inevitabilmente disturbi che possono essere rilevati. Alla fine della trasmissione di un dato Alice e Bob possono valutare la quantità di informazione in possesso di un attaccante esterno e rimuoverla dalla loro chiave finale, tramite algoritmi di post-processing, nel caso in cui questa quantità non sia troppo grande[1]. Un metodo che può essere utilizzato per la definizione della chiave di comunicazione è il protocollo BB84

2.1.1 Protocollo BB84

Il primo protocollo per la comunicazione quantistica è stato elaborato nel 1984 da Charles H. Bennett e da Gilles Brassard, ecco il motivo del nome BB84 nome con il quale questo protocollo è conosciuto [2]. Questo protocollo utilizza 4 stati

[1]Nel caso in cui invece la quantitá di informazione in possesso dell'attaccante esterno sia troppo grande allora il protocollo viene interrotto e fatto ricominciare dall'inizio.

quantistici, o qubit, che possono essere sintetizzati sfruttando la polarizzazione di singoli fotoni, appartenenti a 2 basi:

- Verticale-Orizzontale
- Diagonale-Antidiagonale

Convenzionalmente viene attribuito alla base orizzontale e diagonale il valore 0 mentre alla base verticale e antidiagonale il valore 1.

Alice crea una "chiave grezza" composta da bit a cui è stato associato casualmente una delle due basi, Bob misura la polarizzazione degli stati inviati da Alice e associa ad ogni bit, anch'esso in modo casuale, una delle due basi. Alla fine della comunicazione viene eseguito un data-processing che permette di verificare quali bit sono stati inviati da Alice e ricevuti da Bob usando la stessa base. Alice quindi dice solo se lo stato in cui ha codificato quel bit è compatibile oppure no con la base annunciata da Bob. Se lo stato è compatibile il bit viene mantenuto, altrimenti viene ignorato. Questa chiave più corta, ottenuta dalla conciliazione delle basi, viene chiamata "chiave estratta". Il funzionamento del protocollo si può osservare anche in figura 2.1.

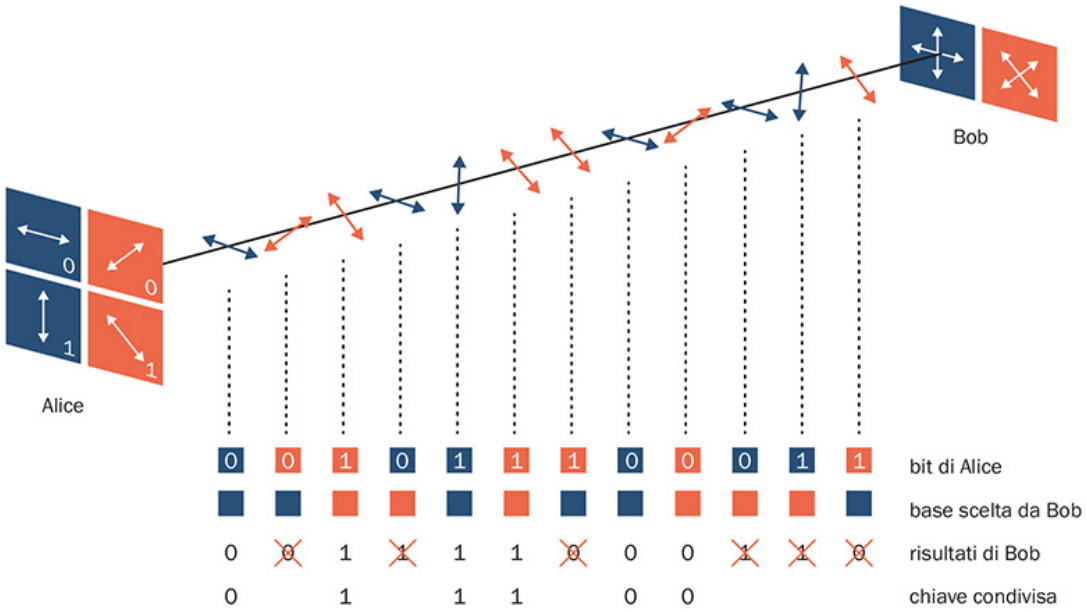

Figura 2.1: In questa immagine si può osservare il funzionamento del protocollo BB84: Alice e Bob confrontano le basi con cui il bit è stato inviato e letto e vengono rimossi quelli che sono stati letti con base diversa da quella con cui sono stati inviati. Alla fine della trasmissione i bit rimanenti formano la chiave. [6]

2.2 Introduzione all' iPognac

L' iPOGNAC, visibile in figura 2.2, è un encoder di polarizzazione che genera stati di polarizzazione con sistema di riferimento fisso nello spazio libero, che non richiede una calibrazione né al ricevitore né al trasmettitore, caratterizzato da un'alta stabilità temporale e con un basso QBit Error Rate (QBER) intrinseco [1].

L'iPOGNAC è una soluzione promettente per la comunicazione quantistica tra satelliti o oggetti in movimento almeno per due ragioni:

- La sua abilità di generare degli stati di polarizzazione fissi rispetto al sistema di riferimento del trasmettitore elimina la necessita di avere una calibrazione del trasmettitore
- Dato che il suo output è nello spazio libero si presta ad essere interfacciato con un telescopio.

Inoltre la sorgente presentata può essere miniaturizzata attraverso della tecnologia micro-ottica, potendo quindi essere sfruttata nelle comunicazioni QKD in fibra ottica con il requisito di avere una calibrazione al ricevitore.

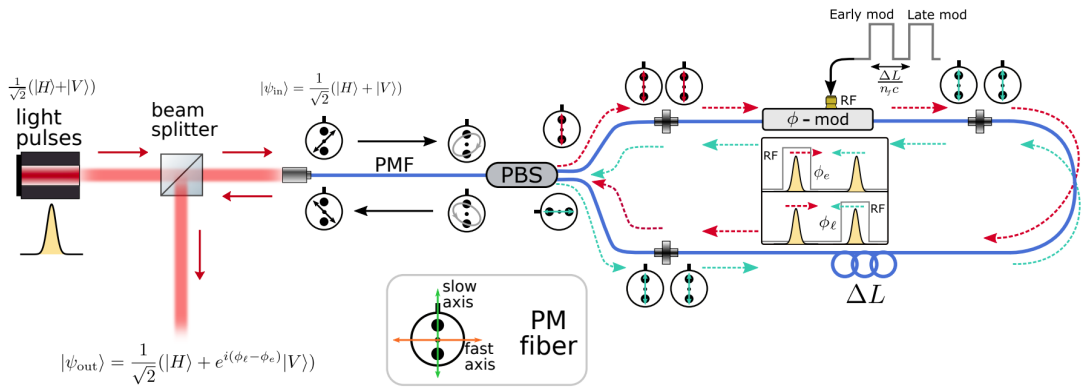

Figura 2.2: In questa immagine si può osservare com'è composto l'iPOGNAC. La delay chain che si è creata agisce su Early mod e Late mod, visibili in alto a destra dell'immagine [1]

3

Materiali e metodi utilizzati

3.1 Cos'è un FPGA

Un FPGA (o Field Programmable Gate Array) è un dispositivo elettronico costituito da componenti logici e connessioni programmabili. In particolare i circuiti interni possono essere riprogrammati, tramite una opportuna stringa di bit (o bitstream) di configurazione, in modo da implementare i principali dispositivi logici noti, sia combinatori che sequenziali. Anche le connessioni tra i vari blocchi possono essere programmate, in modo da ottenere il percorso desiderato per i segnali.

La struttura interna di un FPGA è tipicamente composta da una matrice di blocchi riconfigurabili, o CLB (Configurable Logic Blocks), dei quali quelli sul contorno della matrice si occupano anche di gestire i segnali di Input/Output. I CLB costituiscono l'unita base configurabile e sono collegati tra loro tramite una rete anch'essa configurabile a seconda delle necessità dell'utente. Ogni CLB può essere programmato in modo da implementare qualsiasi tra i circuiti supportati dall' FPGA. Internamente ogni CLB e formato da (tipicamente 4) slice e ogni slice puo contenere a sua volta diversi componenti logici a seconda del modello di FPGA considerato. Tipicamente all' interno delle slice sono presenti delle LUT (Look Up Tables). Una LUT opera come una funzione logica che, avendo N bit in ingresso, permette di assegnare un valore all' unica uscita binaria per ciascuna delle 2^N possibili configurazioni degli ingressi. Una rappresentazione fedele di una LUT e una tabella con 2^N righe e $N + 1$ colonne.

La scheda utilizzata per condurre gli esperimenti necessari allo sviluppo della tesi è la ZedBoard della Avnet ed é stata programmata con l'ambiente di sviluppo Vivado per la parte FPGA/HW e VITIS per la parte software.

3.1.1 Scheda utilizzata

ZedBoard di Digilent è una scheda di sviluppo a basso costo per un All Programmable SoC (AP SoC) Xilinx Zynq-7000. Questa scheda contiene tutto il necessario per creare un progetto basato su Linux o altro sistema operativo/RTOS. Inoltre possiede numerosi connettori di espansione del sistema di elaborazione e I/O di logica programmabile per un facile accesso da parte dell'utente. La scheda è visibile in figura 3.1.

La scheda di sviluppo zedboard è dotata di una porta Ethernet 10/100/1000 Mbps, 512 MB di SDRAM DDR3 a 16 bit e 32 MB di memoria Flash QSPI. Il chip FPGA su questa scheda è lo Zynq XC7Z020-2CLG484I con Arm Cortex®-A-A9 MPCore dual core integrato con un 28nm Artix basato su logica programmabile, possiede fino a 6.6M di celle logiche[7].

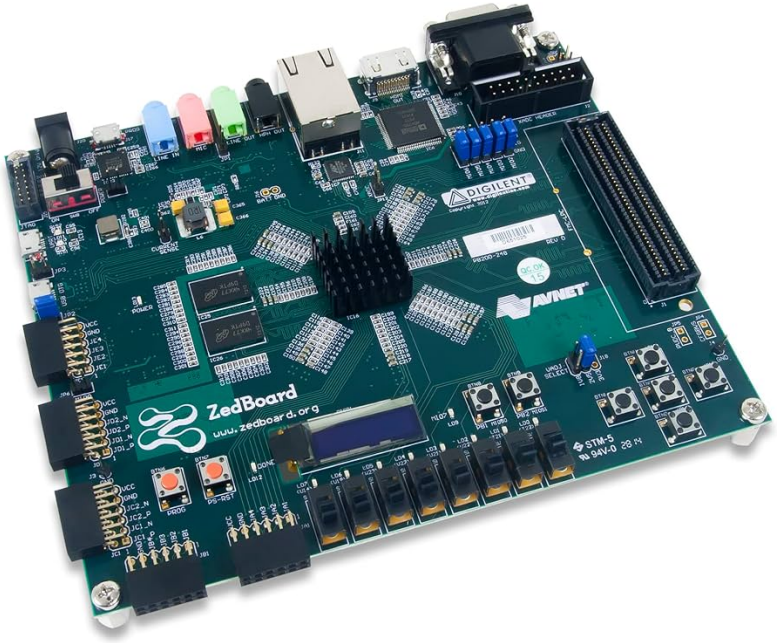

Figura 3.1: Zedboard Zynq-7000, scheda utilizzata durante tutte le analisi

3.1.2 Catena di ritardo

La catena di ritardo che si andrà a implementare sarà composta da una serie di elementi di ritardo che possono essere diversi a seconda del tipo di implementazione utilizzata. Viene quindi sfruttato il tempo impiegato dal segnale in ingresso per andare dall'elemento di delay N all' elemento N+1 viene definito "step" e corrisponde quindi alla sensibilità della delay line che dipenderà anch'esso dal metodo di utilizzato per la creazione di catena di ritardo. In figura 3.2 è possibile vedere un esempio generale della struttura. è possibile utilizzare un Multiplexer (MUX) o equivalente per collegare l'uscita principale ad uno qualsiasi degli step della catena e quindi ritardare a piacere il segnale di uscita rispetto a quello di ingresso.

3.2 Sistema di controllo della temperatura

Il sistema di controllo della temperatura utilizzato durante le misurazioni è composto da una camera climatica ed una cella introdotta tra il chip della scheda e il dissipatore. La combinazione di questi due strumenti permette di eseguire un'analisi con temperatura variabile tra i 5 °C e gli 80 °C

3.2.1 Cella di Peltier

La cella di Peltier è uno strumento elettronico che si basa su due effetti diversi:

- Effetto Seeback
- Effetto Peltier

Il primo effetto stabilisce che in presenza di una differenza di temperatura in un circuito realizzato con due metalli, si è in grado di produrre una forza elettromotrice.

Il secondo effetto invece stabilisce che se la corrente elettrica circola tra due parti metalliche diverse ma poste in contatto, si ha un trasferimento di calore.

Con una cella di Peltier si possono applicare entrambi i principi in modo da generare corrente elettrica da una fonte di calore o refrigerare un dispositivo elettronico troppo caldo. Questo è reso possibile perchè all'interno di una cella di Peltier sono presenti due semiconduttori uno di tipo P e uno di tipo N, collegati a ponte, creando quindi un generatore di corrente e contemporaneamente uno scambiatore di energia termica

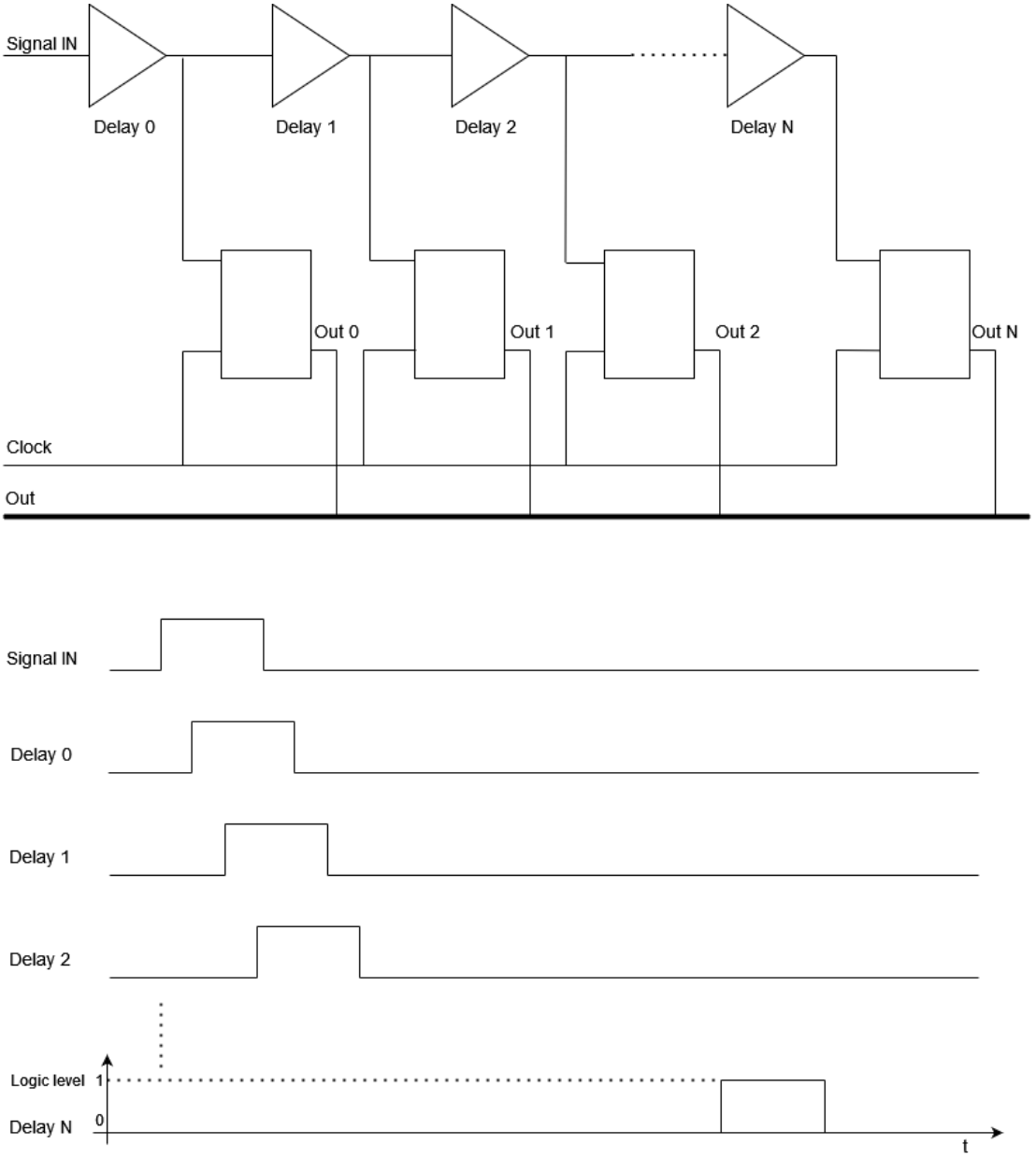

Figura 3.2: La delay chain che si implementerà permetterà di ritardare il segnale in ingresso da un minimo definito dal tempo necessario al segnale per entrare nel primo elemento di ritardo e uscire dal flip flop rispettivo, ad un massimo che corrisponde a N volte questo tempo di ritardo, dove N dipende da quanti elementi di ritardo si sono implementati nella delay chain. Si sottolinea il fatto che la linea corrispondente al segnale in uscita Out è più marcata perché corrisponde ad un bus mentre le linee di Signal IN e Clock corrispondono a singoli bit e sono quindi più sottili

3.2.2 Camera climatica

Una camera climatica è un'apparecchiatura da laboratorio, che permette di testare la reazione di campioni di diverso tipo a specifiche condizioni ambientali,

come ad esempio test di temperatura ed umidità. La camera climatica che verrà utilizzata durante l'esperienza è la ACS Compact Climatic Chamber (60-200L).

4

Analisi e Risultati

4.1 Situazione iniziale

La situazione iniziale si presenta con due contributi: ritardo Coarse realizzato in linguaggio VHDL e Superfine implementato mediante l'utilizzo del modulo IODelay fornito dalla Xilinx. Per la catena di ritardo "Fine" era già presente una prima implementazione su cui però non era stata fatta nessuna analisi sulla linearizzazione, inoltre tale catena presentava degli step molto irregolari.

Si sono quindi eseguiti i test sulla catena di ritardo Fine iniziale, l' andamento che si è riscontrato è descritto in figura 4.1. Si evidenzia un comportamento lineare della catena di ritardo ma con fluttuazioni dovute ai diversi tipi di percorso compiuti dal segnale all'interno della scheda, al variare del numero di ritardi voluti, decisi da Vivado in fase di Sintesi del programma. Il parametro che descrive il grado di linearità iniziale dei tempi di ritardo è il χ^2:

$$\chi^2 = \sum_i \frac{(X_i - M_i)^2}{M_i}, \tag{4.1}$$

dove X_i è il valore misurato mentre M_i è il valore che si dovrebbe riscontrare nella misura secondo la funzione lineare che meglio rappresenta i dati ottenuti. Questo parametro permette di valutare quanto simile è il dato che si è osservato rispetto alla funzione lineare che meglio rappresenta l'andamento dei dati che si sono ottenuti. L'andamento ottenuto è indicato in figura 4.1.

Si è quindi reso necessario applicare delle ottimizzazioni alla struttura im-

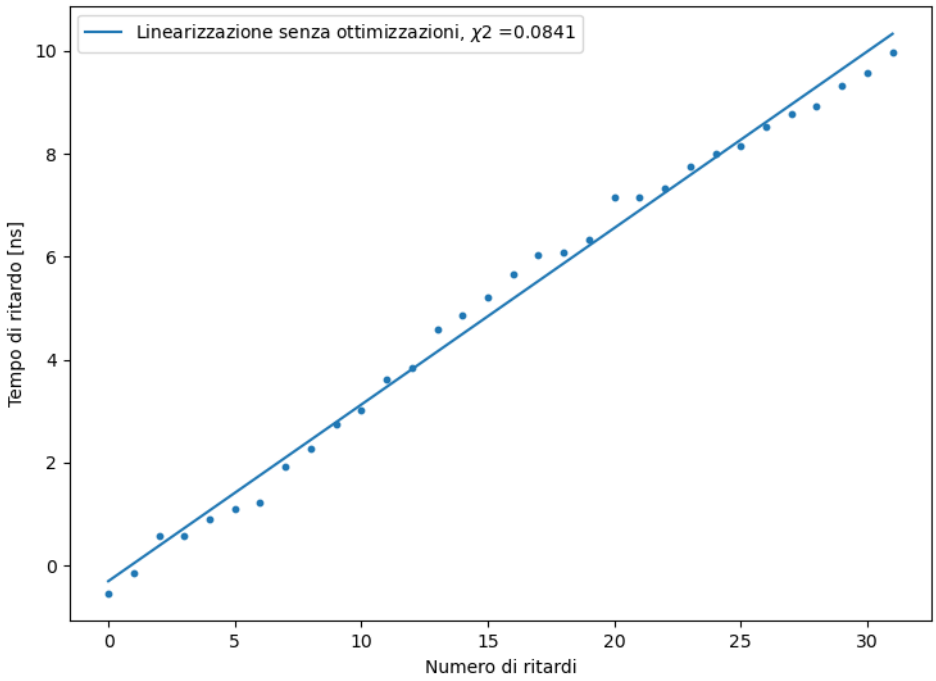

Figura 4.1: Andamento dei tempi di ritardo che si avevano inizialmente quando non era implementata nessuna ottimizzazione e i percorsi della della delay chain erano decisi direttamente dal programma Vivado in fase di sintesi

plementata in modo da ottenere un andamento più lineare.

4.2 Ottimizzazioni

4.2.1 Ottimizzazione 7x2

La prima ottimizzazione che si è implementata è stata mediante l'utilizzo della funzione pBlock resa disponibile dall'ambiente di lavoro Vivado. Si è creato un blocco costituito da 7 righe e 2 colonne di Configurable Logic Blocks (CLBs), ovvero le unità base che possono essere configurate nella struttura dell' FPGA, e posizionato in un punto della scheda in cui la struttura creata potesse essere più simmetrica possibile (evitando quindi zone già interessate da altri segnali necessari ad altri scopi del sistema). Con questa ottimizzazione quindi non sarà più l'ambiente di sviluppo Vivado a decidere la posizione dei componenti della delay chain (che possono anche essere posizionati molto distanti l'uno dall'altro,

in base a ciò che il software ritiene più efficiente) ma tutti i componenti della catena di ritardo saranno posizionati all'interno del blocco che si è creato. Questo ha messo in evidenza un leggero miglioramento in termini di grado di linearità dei tempi di ritardo, osservabile in Figura 4.3. Si nota inoltre che il χ^2 (χ^2 7x2) è diminuito rispetto al caso precedente di figura 4.1.

4.2.2 Ottimizzazione 1x14

La seconda ottimizzazione attuata è stata una modifica della prima: si è sostituito il blocco precedentemente creato con uno da creato da 1 riga e 14 colonne per verificare se i tempi di ritardo migliorassero modificandone la la geometria all'interno della scheda. In questo caso quindi tutti i componenti della delay chain sono stati posizionati all'interno di questo nuovo blocco. Questa modifica ha portato ad un dimezzamento del χ^2, ovvero ad un raddoppio del grado di linearità rispetto alla retta di regressione descritta dal χ^2 7x2. Il coefficiente di correlazione di questa retta viene mostrato in Figura 4.3 come χ^2(1x14).

4.2.3 Ottimizzazione mediante Look Up Tables

Il terzo tipo di ottimizzazione che si è provato ad utilizzare invece è un'ottimizzazione più forte, evitando quindi di usare la funzione pBlock di Vivado, per vedere se i risultati migliorassero ulteriormente. Si è quindi riprogrammato il blocco della catena dei ritardi mediante codice VHDL creando un LOC Constrainto, ovvero istanziando inizialmente una catena di porte AND, ma non è stato possibile in quanto la scheda utilizzata non la riconosceva come tipo di primitiva. Si è quindi creato un LOC Constraint di Look Up Table (LUT) [5] che, come spiegato in precedenza, sono presenti nella struttura dell'FPGA. Si è quindi modificato il programma in modo che creasse il circuito in figura 4.2.

Questo tipo di approccio ha portato inizialmente un peggioramento in termini di linearità dato che il χ^2 LUT 1, definito cosí poiché è stato il primo tentativo di implementazione della delay chain mediante delle LUT, è aumentato rispetto al χ^2 1x14, visibile in figura 4.3. Si è quindi provato a riservare una parte della scheda ancora interamente libera ad essa, in modo da vedere se la completa assenza di altri componenti e segnali necessari per la sintesi di altre funzioni potesse migliorarne le qualità. Si sono ripresi i dati e il risultato è stato dimezzamento del χ^2 ovvero ad un raddoppio del grado di linearità dei tempi di ritardo

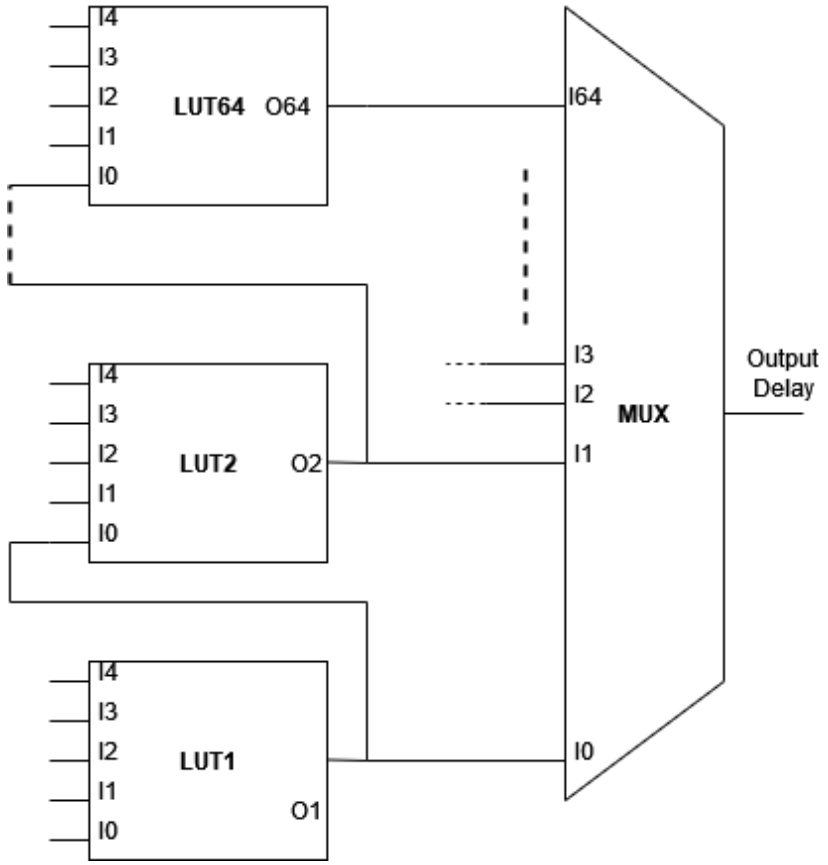

Figura 4.2: Per creare una delay chain con delle look up tables si è deciso di utilizzare solo un ingresso di ognuna di esse, collegate una in successione all' altra e contemporaneamente collegate ad un multiplexer (MUX) che permette di cambiare l'uscita dalla catena di ritardo, ovvero la grandezza del delay in termini temporali

anche se la pendenza della retta è aumentata. Quest'ultima retta è identificata dal χ^2 LUT 2 indicando che è il secondo tentativo di implementazione delle LUT.

L'aumento della pendenza della retta indica un aumento del tempo che impiega il segnale ad andare dalla LUT N alla LUT N+1. Dato che l' obiettivo che ci si era prefissato era di avere una delay line con tempi di ritardo più lineari possibile questo risultato può essere considerato trascurabile. Si avranno quindi degli step più grandi ma più simili tra loro.

4.2.4 Ottimizzazione mediante Carry4

Si è infine implementato la delay chain con un altro tipo di LOC Constraint, questa volta utilizzando dei Carry4, [4] poiché la lunghezza del percorso fatto

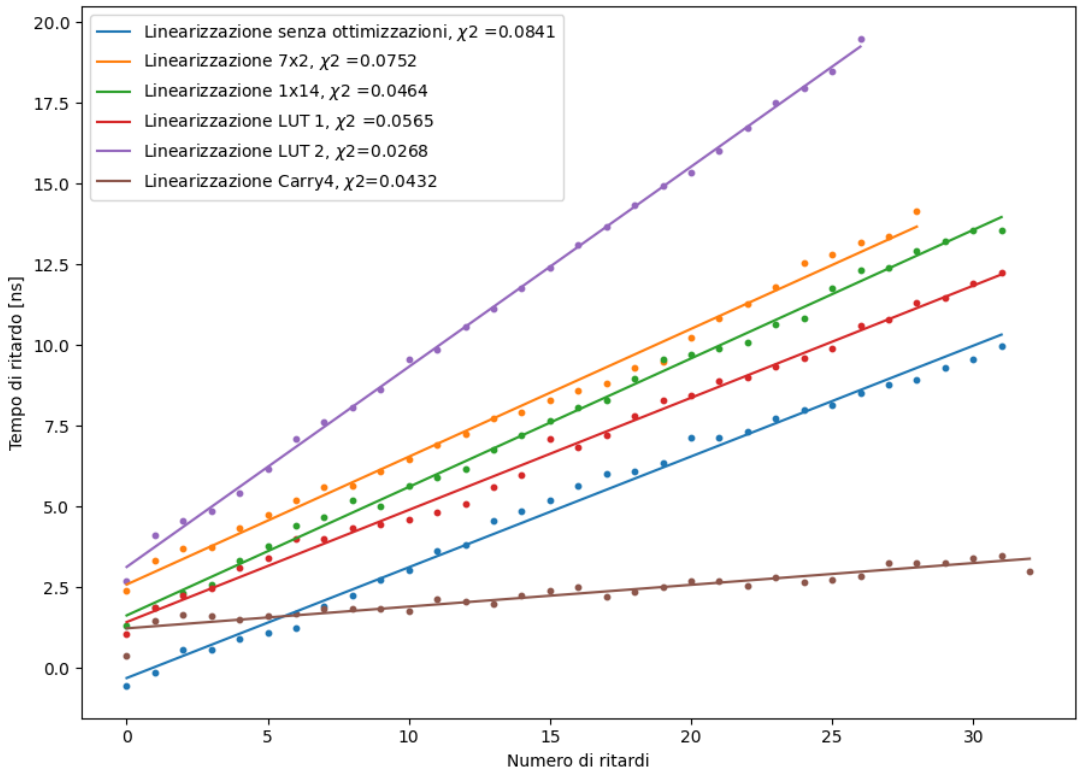

Figura 4.3: Risultati con ottimizzazione 7x2 e 1x14 mediante l'utilizzo della funzione pBlock

dal segnale per passare da un Carry4 al successivo può essere considerata costante, dato che la scheda è stata ottimizzata proprio per svolgere questo tipo di operazione, cosa che invece non accade nelle look up tables, che hanno dei percorsi molto irregolari. Si sono quindi sostituite, nello stesso punto della scheda, tutte le LUT con dei Carry4. Questo approccio ha sottolineato una diminuzione della pendenza della retta caratteristica, indicando quindi una diminuzione dei tempi di ritardo, come spiegato in precedenza. Si evidenzia anche però che il grado di linearità χ^2 Carry4 risulta essere più alto di quello ottenuto durante la seconda implementazione delle LUT (χ^2 LUT 2) come è possibile osservare in figura 4.3. Questo approccio non è stato utilizzato, si è preferito utilizzare la seconda implementazione delle LUT poiché fornivano un risultato migliore.

4.3 Caratterizzazione al variare della temperatura

Si vuole ora osservare il comportamento dei tempi di ritardo della scheda variare al variare della temperatura, per una migliore comprensione del loro

andamento in condizioni diverse dalla temperatura ambiente. Si è quindi creato un sistema che possa variare la condizione termica del processore della scheda ZedBoard in modo da poter osservare i possibili cambiamenti nei tempi di ritardo all'aumentare o diminuire della temperatura di lavoro Durante queste misurazione si è usata solo la cella di Peltier (senza l'utilizzo della camera climatica) che permette, alimentandola con corrente positiva, di raffreddare il processore o, alimentandola con corrente negativa, di riscaldarlo. Data l'assenza della camera climatica, le misure sono state realizzate tra i 25°C e gli 80°C. Utilizzando il sensore che monitora la temperatura del processore, integrato nella scheda, si è potuto misurare i tempi di ritardo nelle diverse condizioni di lavoro. Per una migliore comprensione dell'andamento al variare della temperatura si sono osservate le variazioni dei coefficienti angolari delle rette al variare della temperatura. Questo ha sottolineato una variazione minima dei delay, rendendone trascurabile il risultato. Nel grafico di Figura 4.4 é possibile osservare la variazione dei coefficienti angolari.

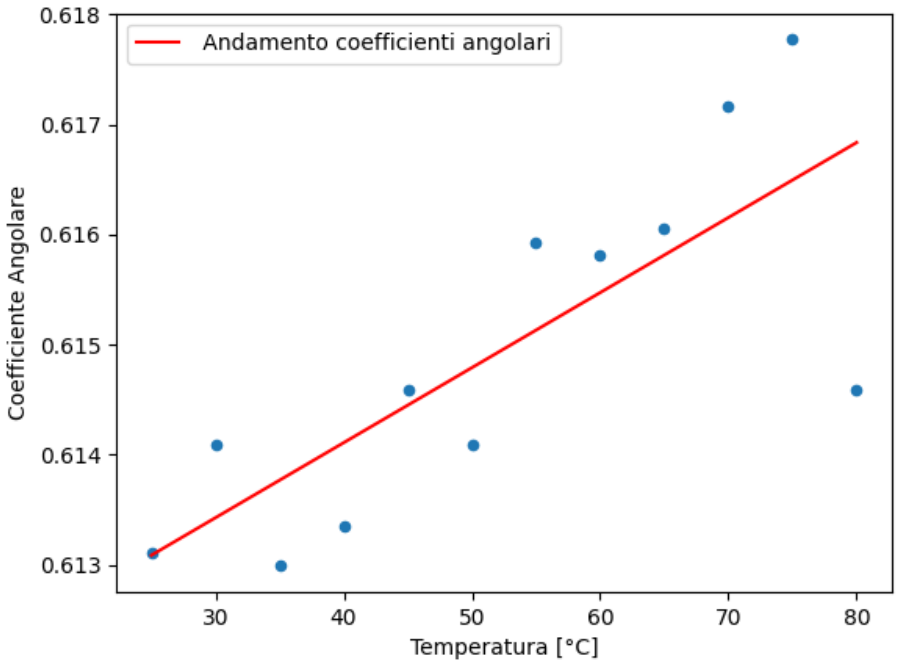

Figura 4.4: è possibile osservare che la variazione dei coefficienti angolari sull'asse delle ordinate è molto piccola al variare della temperatura, questo giustifica il fatto di ritenere il suo effetto sui tempi di ritardo trascurabile

4.4 Analisi dei jitter al variare della temperatura

In ultima analisi si sono voluti verificare i cambiamenti dei jitter della delay chain, ovvero i disturbi elettronici intrinseci del sistema, al variare della temperatura di lavoro.

Per questo scopo si è utilizzata sia la camera climatica che la cella di peltier, in modo tale da riuscire ad avere un'analisi di essi con una temperatura che varia tra i 5 °C e gli 80 °C, e un programma implementato su un'ulteriore scheda FPGA che permettesse di misurare il tempo di arrivo dei segnali in ingresso rispetto al clock, permettendo quindi di misurare i jitter al variare della temperatura all'inizio, ovvero con delay nullo (pari ad un fine delay di 0) e alla fine della delay line (pari allo step 32 del delay fine). In figura 4.5 è possibile osservare il diagramma del setup utilizzato mentre in figura 4.6 e 4.7 è possibile osservare il setup utilizzato per l'acquisizione delle misure.

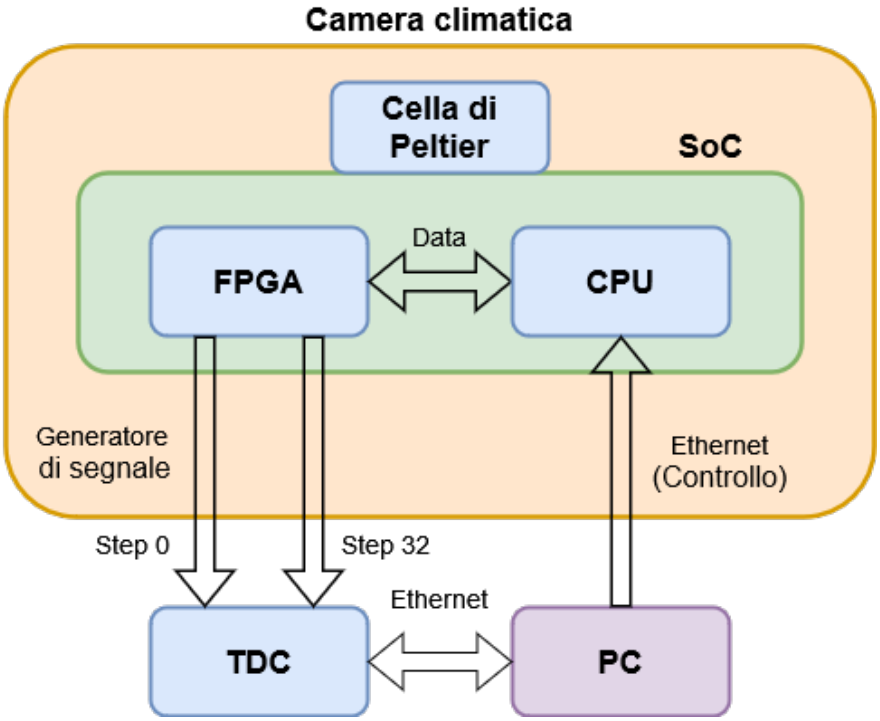

Figura 4.5: Diagramma del setup utilizzato utilizzato durante le misurazioni della variazione dei jitter al variare della temperatura

Per analizzare la variazione dei jitter si è misurato il valore all'inizio della catena (0) e al 32esimo elemento di ritardo, in questo modo sottraendo il 32 esimo a quello iniziale si ottiene il valore dei jitter relativo solo alla delay chain.

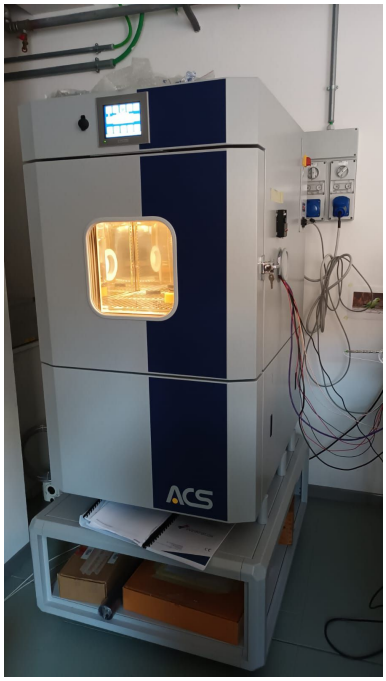

Figura 4.6: Setup esterno della camera climatica

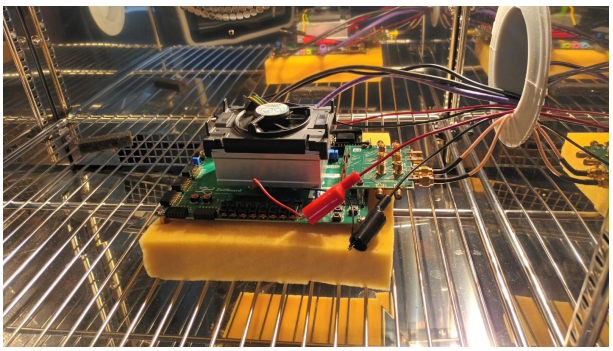

Figura 4.7: Setup e collegamenti alla scheda all'interno della camera climatica

Si è fatto questo perché il valore all'inizio della catena di ritardo è soggetto anche all'influenza degli elementi presenti prima di essa, questo metodo ha quindi permesso di eliminare con buona approssimazione questo contributo indesiderato. Il risultato si può osservare graficamente in figura 4.8

4.5 Test Ottico

Per un'analisi più completa si è poi implementata la struttura realizzata sul reparto elettronico del sistema iPognac e si sono raccolti i dati riguardanti la variazione della polarizzazione dell'impulso ottico, mediante l'uso di un polarimetro, variando il delay della catena. Si è poi realizzato un programma in python che permettesse di visualizzare la variazione della polarizzazione all'aumentare degli step di ritardo per constatare la correttezza dell'analisi eseguita e una sfera di Bloch che mostrasse la posizione dei punti misurati, visibili nelle figure 4.9 e 4.10.

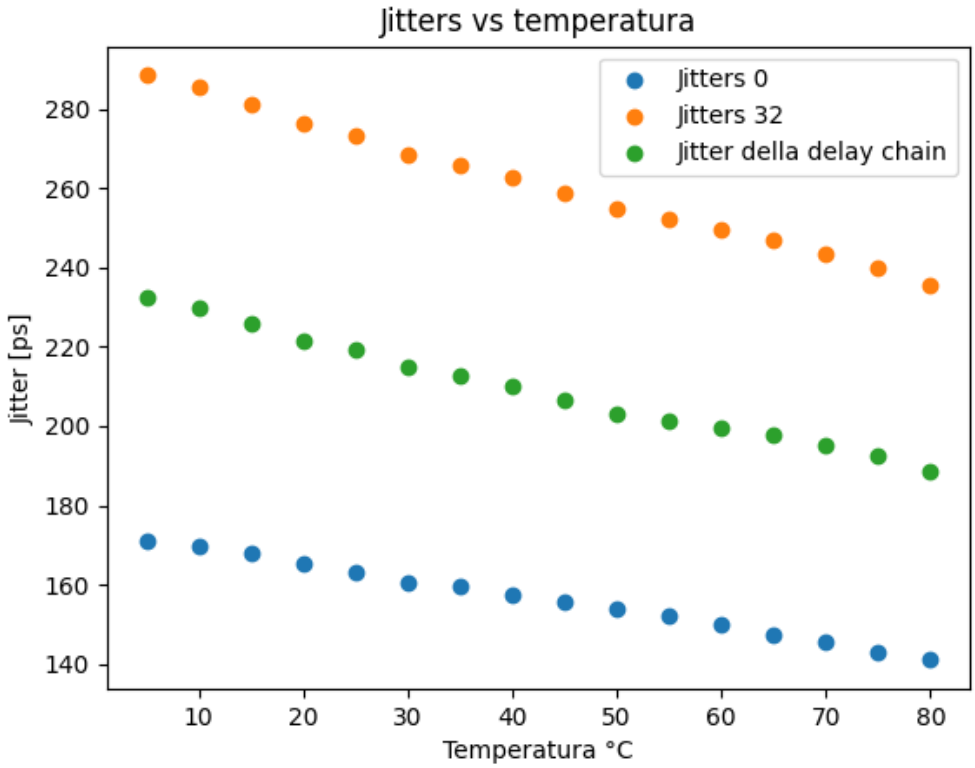

Figura 4.8: Si possono osservare in blu i jitter relativi all'inizio della catena di ritardo mentre in arancione i jitter relativi al 32esimo elemento di ritardo. Inoltre in verde si può osservare l'andamento effettivo dei jitter nella delay chain calcolato come spiegato nel paragrafo precedente

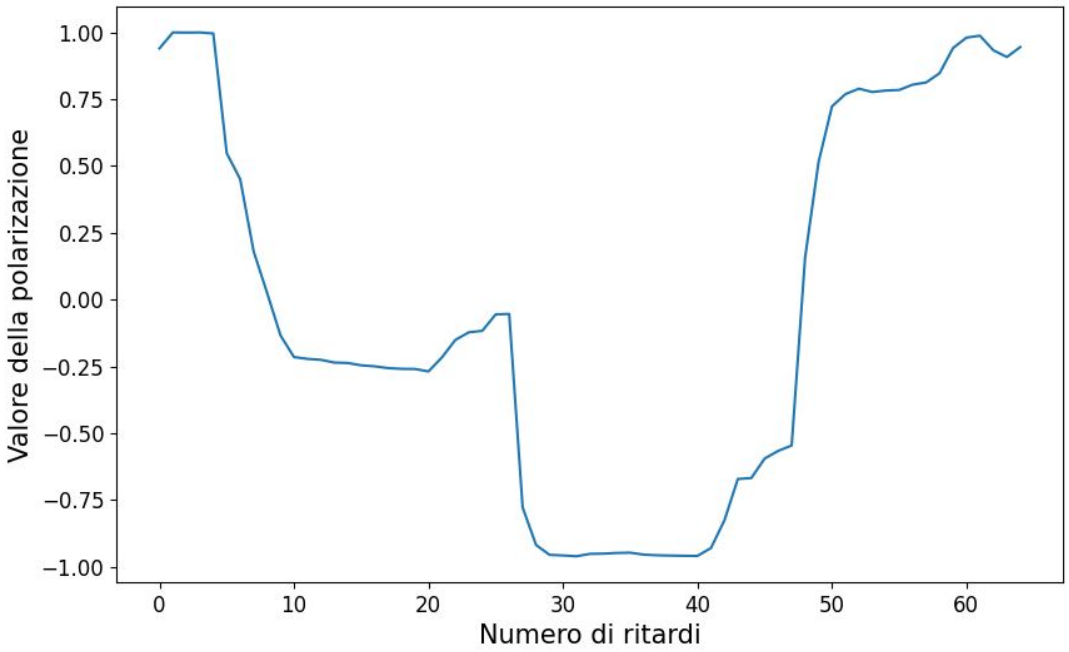

Figura 4.9: Nel grafico è possibile osservare la variazione della polarizzazione al variare degli step di delay utilizzati: La polarizzazione comincia da D (che ha valore 1), scende fino circa ad H (che ha valore 0), scende ulteriormente fino a V (che coincide con il valore -1) ed infine ritorna al valore di D. Si può notare un andamento non privo di rumore e con alcuni punti che si distanziano dal valore atteso. Ciò è dovuto ad una non perfetta calibrazione del setup esterno all'FPGA (elettronica di driving e setup ottico) e a disturbi e riflessioni causate dalla misura eseguita in spazio libero.

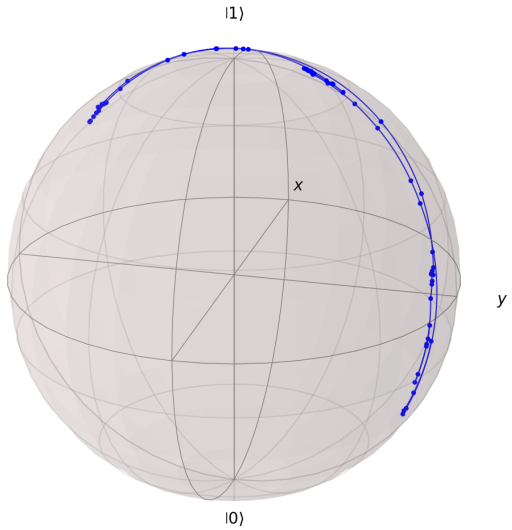

Figura 4.10: Per completezza si osserva anche la variazione degli stati della polarizzazione sulla sfera di Bloch

5

Conclusioni

In questa tesi si è dunque creata ed ottimizzata una delay chain che si andrà ad integrare con le due già esistenti (Coarse e Superfine) che permette di aumentare il range dei ritardi, mantenendoli più lineari possibile ed esaminandone il comportamento anche al variare della temperatura ambientale. In conclusione si è osservato che le ottimizzazioni implementate hanno portato graduali miglioramenti del sistema: inizialmente si aveva un χ^2= 0.0841 che è poi diminuito sempre di più passando da ottimizzazioni più semplici (ovvero quelle implementate tramite pBlock) alle ultime ottimizzazioni realizzate con il constraint di LUT e Carry4. Si è quindi arrivati alla conclusione che l'implementazione migliore sia quella utilizzata durante il secondo tentativo di utilizzo delle Look Up Tables, che permette di ottenere un grado di linearizzazione con un χ^2=0.0268, portando quindi aumento del grado di linearità circa 3 volte superiore a quello iniziale. Si vuole ricordare che l'aumento della pendenza della retta indica un aumento del tempo che impiega il segnale ad andare dalla LUT N alla LUT N+1. Dato che l' obiettivo che ci si era prefissato era di avere una delay line con tempi di ritardo più lineari possibile questo risultato può essere considerato accettabile. Si avranno quindi degli step più grandi ma più simili tra loro.

Si è visto che al variare della temperatura del chip e ambientale, i delay subiscono solo un offset, che non è difficile da correggere. I tempi di ritardo quindi non variano in modo significativo al variare della temperatura, rendendo quindi il suo effetto trascurabile.

L'analisi dei jitter ha invece fatto emergere un comportamento anomalo di questi ultimi, in quanto il loro valore medio tende a diminuire all'aumentare

della temperatura, evidenziando quindi un andamento inversamente proporzionale. Questo comportamento non rappresenta un problema poiché, dal punto di vista dell'applicazione, è sufficiente conoscere il comportamento in funzione della temperatura e non è necessario conoscerne le ragioni fisiche. Questa investigazione sarà comunque oggetto di ulteriori misure in futuro.

Infine l'analisi del test ottico ha portato alla conferma del comportamento che ci si aspettava di osservare, si è verificata e constatata la presenza dei tre stati V, H e D e il tempo in cui la polarizzazione rimane in tali stati è coerente con le aspettative (5 ns in H, 5 ns in V e 10 ns in D).

Referenze

[1] Marco Avesani et al. «Stable, low-error, and calibration-free polarization encoder for free-space quantum communication». In: *Optics Letters* 45.17 (2020), pp. 4706–4709.

[2] Charles H Bennett e Gilles Brassard. «Quantum cryptography: Public key distribution and coin tossing». In: *arXiv preprint arXiv:2003.06557* (2020).

[3] Nicolas Gisin et al. «Quantum cryptography». In: *Reviews of modern physics* 74.1 (2002), p. 145.

[4] «Implementazione carry4 utilizzata». In: URL: https://docs.xilinx.com/r/en-US/ug953-vivado-7series-libraries/CARRY4.

[5] «Implementazione LUT utilizzate». In: URL: https://docs.xilinx.com/r/en-US/ug953-vivado-7series-libraries/LUT5.

[6] Francesco Basso Basset e Rinaldo Trotta Michele Rota. «Comunicare nellera dei quanti. » In: (2022). DOI: 10.23801/asimmetrie.2022.33.7. URL: https://www.asimmetrie.it/comunicare-nell-era-dei-quanti.

[7] «Scheda di sviluppo SoC ZedBoard Zynq-7000 Arm/FPGA». In: URL: https://digilent.com/reference/programmable-logic/zedboard/start.