



**Politecnico
di Torino**

Laurea Magistrale in Ingegneria Elettronica
Anno Accademico 2021-2022

Relazione di laboratorio di sistemi digitali integrati

Progetto Eco

Docenti:

Prof. Massimo Ruvo Roch
Prof. Maurizio Zamboni

Studenti:

Marco Massetti matr. 301163
Pietro Fagnani matr. 303490

Indice

1.	Introduzione	1
2.	Convertitore	2
2.1.	Datapath.....	3
2.2.	Timing Diagram	4
2.3.	Diagramma degli stati	5
3.	Generatore di eco	6
3.1.	Datapath.....	7
3.2.	Timing Diagram	10
3.3.	Diagramma degli stati	13
4.	Analisi risultati.....	14
4.1.	Simulazione con Modelsim	14
4.2.	Sintesi con Quartus.....	17
4.3.	Funzionamento su FPGA	18

1. Introduzione

L'obiettivo di questo laboratorio è quello di progettare un circuito in grado di generare l'eco di un segnale che rispetti alcune caratteristiche (delay, decay, gain).

Questi parametri sono ricevuti tramite comunicazione seriale utilizzando l'interfaccia UART progettata precedentemente, e poi salvati in tre appositi registri.

Il delay indica il ritardo in numero di campioni tra il segnale e le sue copie generate.

Il decay specifica la quantità di segnale che verrà ripetuto, un valore di decay più alto significa un eco formato da più ripetizione del segnale

Il mix denota il volume dell'eco che sarà presente nel segnale d'uscita

Il generatore di eco lavora in modo quantizzato, quindi in assenza di segnale in ingresso il circuito non processa alcun dato e di conseguenza l'output resta costante. Questo permette all'utente che trasmette i dati di dettare i tempi di processamento ed al circuito non resta altro che effettuare le operazioni senza la preoccupazione di gestire la frequenza del segnale campionato.

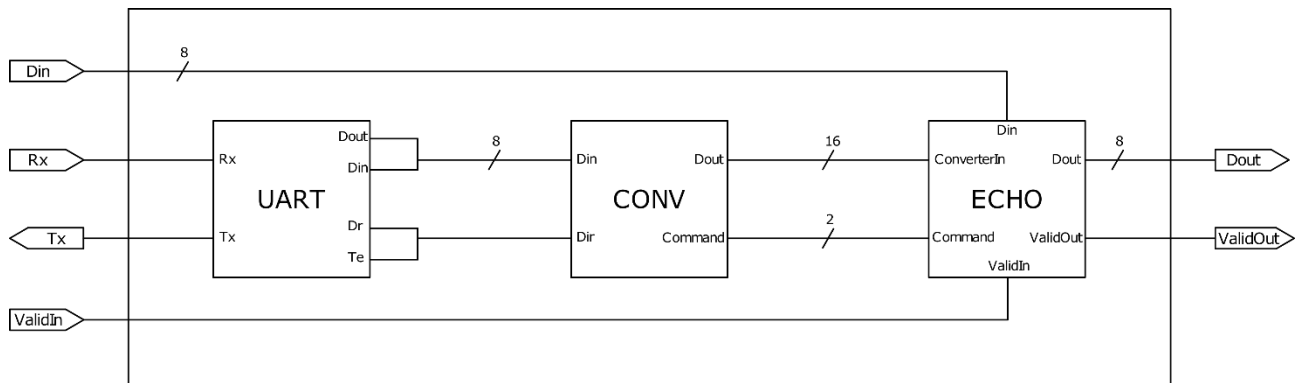


Figura 1 Schema a blocchi generatore di eco e interfaccia

Come si può notare dallo schema, l'architettura è divisa in tre parti:

- UART: ha il compito di ricevere i comandi dal microcontrollore.
- CONV: converte i comandi ricevuti in valori in esadecimali.
- ECHO: salva i comandi nei registri e processa i segnali in ingresso.

2. Convertitore

Il convertitore ha il compito di acquisire dati in ingresso su 8 bit con codifica ASCII dalla UART. Capire se si tratta di un comando o di un valore e nel secondo caso convertirlo in esadecimale.

Se una comunicazione avviene correttamente, il valore convertito deve essere poi trasmesso al generatore di eco per poter essere salvato nel rispettivo registro.

I comandi validi sono:

- TXXXX Delay
- *XX Mix
- -XX Decay

(Dove X indica un carattere ASCII)

I valori sono espressi in base esadecimale quindi i diversi parametri avranno dimensione 16 bit, nel caso del delay, e 8 bit per parametri mix e decay.

2.1. Datapath

Il registro “*INPUT_REGISTER*” campiona il dato in ingresso quando viene ricevuto il segnale “*data ready*” proveniente dall’interfaccia UART.

Il comparatore “*COMAPARATOR_T*” ha la funzione di controllare che il primo carattere ricevuto sia uno dei tre validi (T, *, -) e nel caso comunicare alla control unit tramite 2 bit “*Val*” di quale comando si tratti.

Il comparatore “*COMPARATOR_H*” invece, controlla e converte in binario il carattere arrivato in ASCII, se il carattere non è valido lo comunica alla control unit tramite il bit “*Valid*”. L’uscita su 4 bit è collegata in ingresso allo “*SHIFT_REGISTER*” di profondità 4, in cui il segnale in ingresso occupa sempre la parte meno significativa. In questo modo a fine comunicazione il primo dato ricevuto si troverà nei bit più significativi.

Tutti e 16 i bit sono poi trasmessi alla parte di generazione dell’eco, in caso il comando fosse di gain o decay saranno prelevati solamente gli 8 bit meno significativi dello shift register.

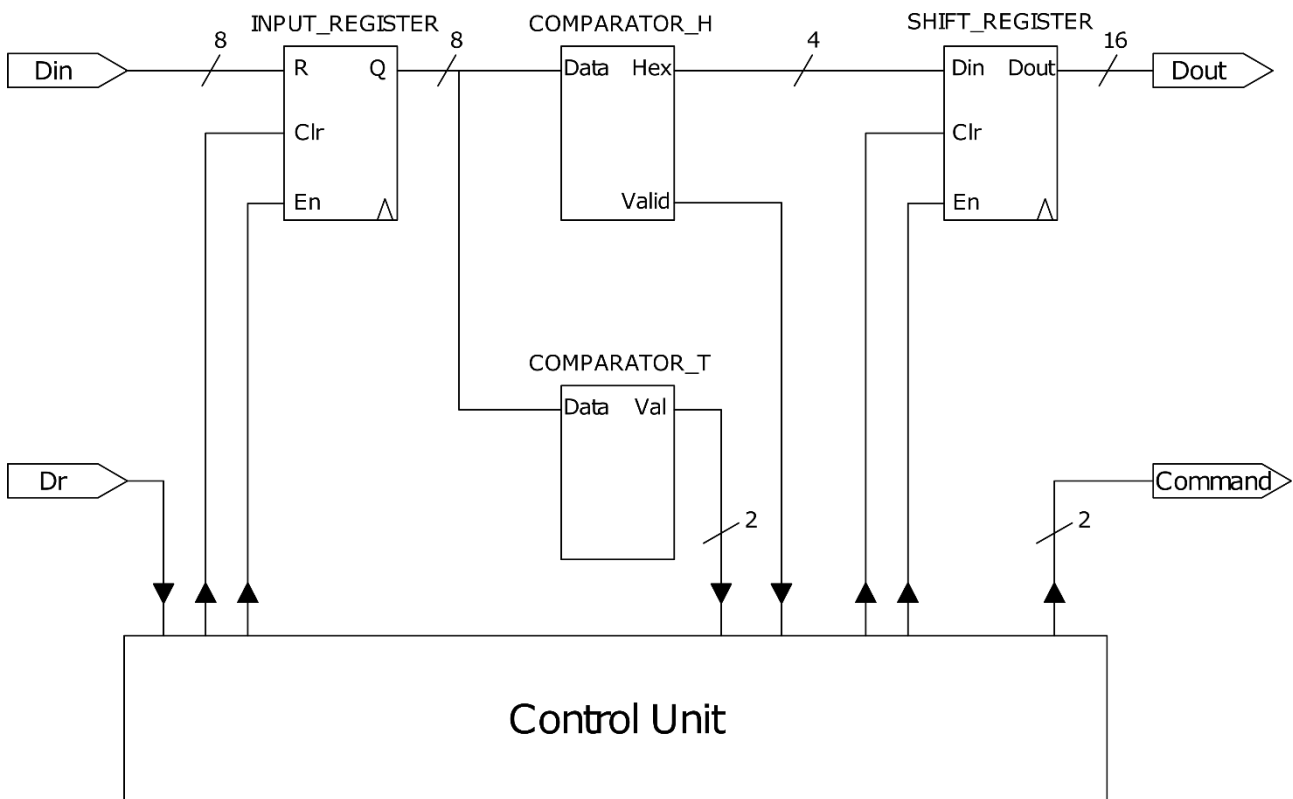


Figura 2 Datapath convertitore

2.2. Timing Diagram

Tramite la realizzazione di un timing diagram è stato possibile definire le tempistiche di lavoro del circuito e della control unit.

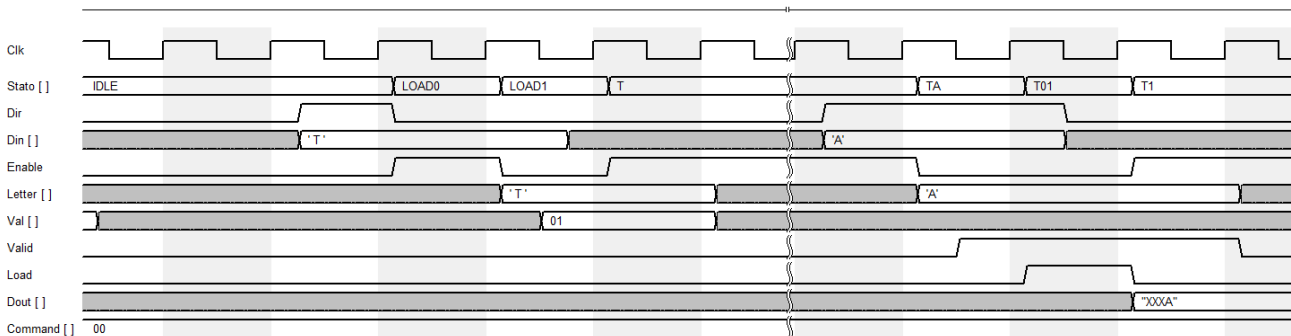


Figura 3 Timing del convertitore: campionamento dati

Dopo il reset la macchina si trova in uno stato di “IDLE”, quando viene ricevuto “Dir” uguale ad 1 il dato in ingresso è valido. Per come è stata realizzata l’UART nei laboratori precedenti il dato in uscita resta valido per diverso tempo dopo una comunicazione, è quindi possibile campionare il dato nel colpo di clock successivo, questo permette di non campionare in modo continuo l’ingresso risparmiando in termini di energia consumata.

Nello stato “LOAD0” quindi viene campionato l’ingresso “Letter” e successivamente in “LOAD1” il carattere in ingresso viene convertito in un comando: come nell’esempio (01 per il delay). Viene poi atteso il nuovo segnale di dato valido nello stato “T”, campionando continuamente l’ingresso (Enable = 1).

Quando “Dir” è nuovamente valido si effettua una conversione del carattere in esadecimale (stato “TA”), se la conversione è andata a buon fine “Valid” viene portato a 1 e nel colpo di clock successivo. Nello stato “T01” viene caricato il primo dato nello “SHIFT_REGISTER”.

Successivamente si torna nuovamente in attesa nello stato “T1”.

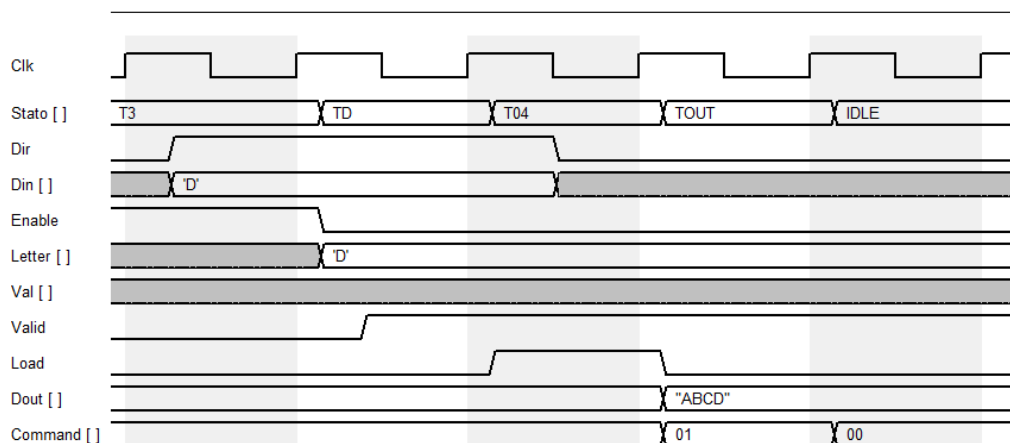


Figura 4 Timing del trasmettore: invio dei dati

Nel caso del delay (comando T) al caricamento del quarto valore “TOUT” viene comunicato al generatore di ego di salvare i dati presenti all’interno dello “SHIFT_REGISTER” nel registro del Delay tramite i bit di “Command” (01).

2.3. Diagramma degli stati

Per comprendere meglio gli stati della control unit può essere utile la creazione di un diagramma degli stati.

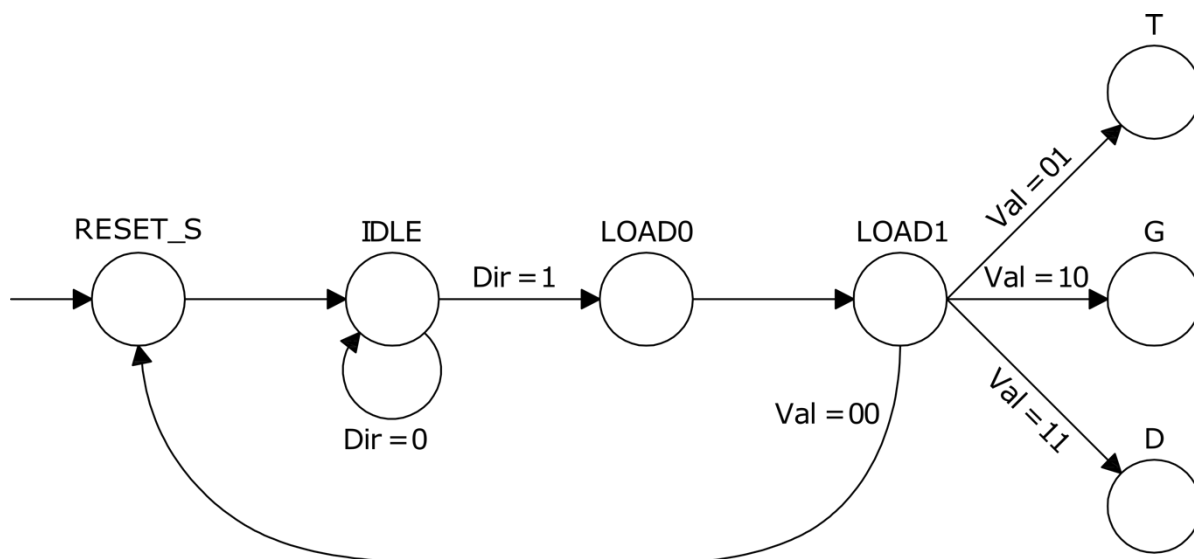


Figura 5 Diagramma degli stati convertitore: identificazione del comando

Come descritto già nel paragrafo precedente, la conversione del primo carattere avviene nello stato “LOAD1”. In base al valore di “Val” la CU intraprende 3 “percorsi” diversi, se Val = 00 allora la conversione non è andata a buon fine e si ritorna nello stato “RESET_S”.

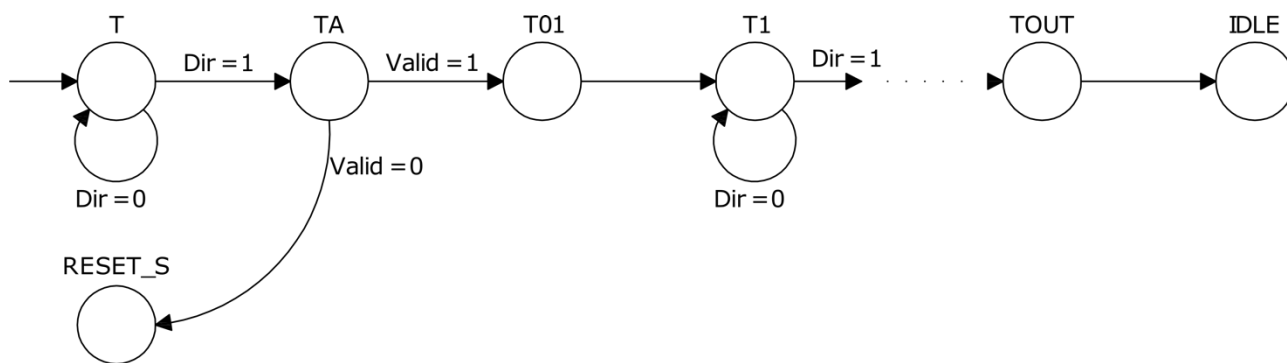


Figura 6 Diagramma degli stati convertitore: lettura del valore e trasmissione

Le operazioni effettuate per T (delay), G (gain) e D (decay) differiscono solamente per il numero di caratteri ricevuti (nel caso del delay sono 4 altrimenti solo 2) e per i due bit di Command trasmessi durante l’ultimo stato (TOUT, GOUT, DOUT) che abilitano l’enable del rispettivo registro.

Durante ogni stato di conversione (TA nell’esempio) se il carattere ricevuto non è ammesso allora si ritorna nello stato di “RESET_S”.

3. Generatore di eco

Il blocco responsabile della generazione dell'eco può essere riassunto come una memoria la cui uscita è riportata in ingresso dopo un ritardo (delay) e con una attenuazione (decay) impostabili. L'uscita del blocco è poi formata da una somma tra l'ingresso attuale e il campione di uscita dalla memoria attenuata a seconda del parametro mix.

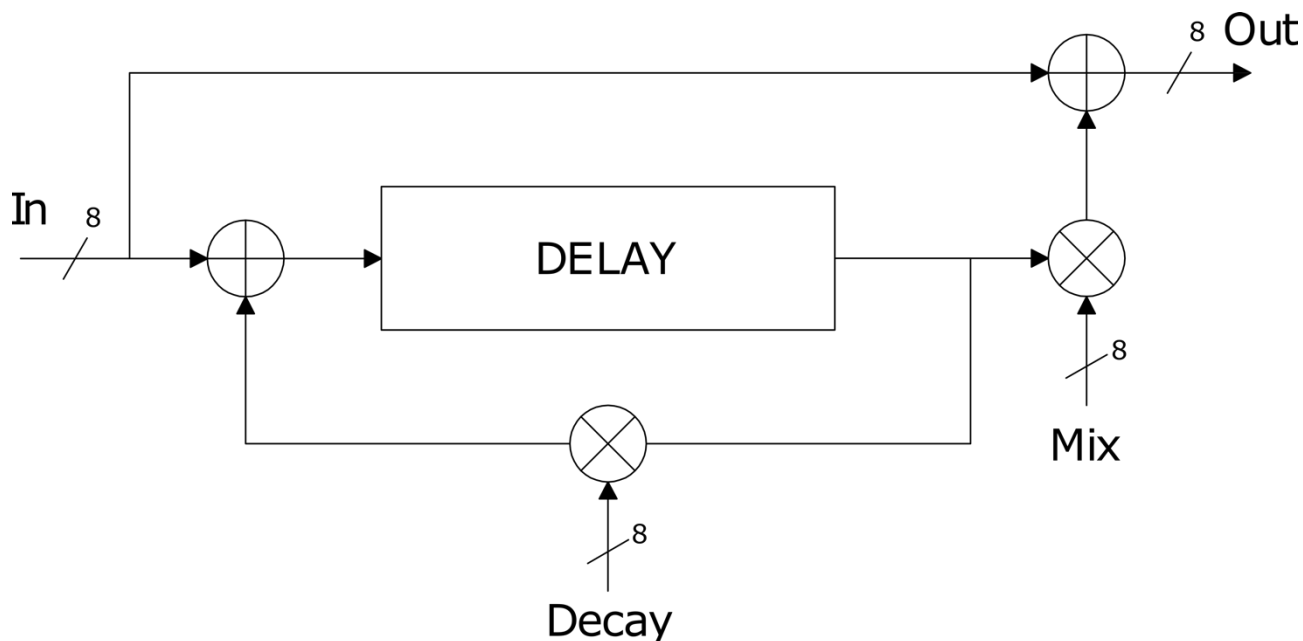


Figura 7 Schema a blocchi generatore di eco

Come descritto in (**Figura 7**) i campioni in ingresso vengono sommati con la retroazione dell'uscita della memoria e il risultato è mandato in ingresso ad essa.

La memoria causa un ritardo dei valori ricevuti, infatti, essa si comporta come una FIFO in cui il valore di profondità è variabile ed è pari al delay.

L'attenuazione della retroazione è ottenuta moltiplicando l'uscita della memoria per il parametro decay, che avendo valore minore di uno, genera sicuramente un valore di uscita inferiore a quello in ingresso.

Come descritto in precedenza l'uscita del componente è poi formata da una somma tra l'ingresso e il campione di uscita dalla memoria, quest'ultimo è prima attenuato venendo moltiplicato per il parametro mix. Anch'esso, come per il decay, sarà inferiore all'unità e quindi provoca un'attenuazione del segnale.

3.1. Datapath

Il blocco di generazione dell'eco può essere suddiviso in quattro sotto blocchi per essere studiato con più semplicità.

- Blocco di salvataggio dei parametri
- Blocco di calcolo degli indirizzi di memoria
- Blocco di retroazione dei campioni
- Blocco di calcolo dell'uscita

- Blocco di salvataggio dei parametri:

Il primo elemento necessario per il funzionamento del dispositivo è quello di salvataggio dei parametri, esso è formato da tre registri, uno a sedici bit e due a otto, che servono rispettivamente per salvare i parametri delay, mix e decay. Il suddetto blocco è direttamente collegato all'esterno tramite due segnali, uno a sedici bit su cui è possibile fornire il dato "*ConverterIn*", e uno a due bit "*Command*" che permette di specificare se il valore sul segnale di dato è valido e che parametro rappresenta. Dato che i registri di decay e mix non necessitano di campionare l'intero valore su sedici bit essi sono collegati esclusivamente agli otto bit meno significativi.

- Blocco di calcolo degli indirizzi di memoria:

La memoria campioni è utilizzata come un buffer circolare di profondità pari al delay. Questo è ottenuto sfruttando una memoria con indirizzi di lettura e scrittura separati, per variare la profondità è quindi sufficiente variare l'offset tra i due indirizzi.

L'indirizzo di scrittura è ottenuto tramite il contatore "*ADDRESS_COUNTER*" senza dover fare nessun ulteriore calcolo. L'indirizzo per la lettura viene invece calcolato come:

$$\text{indirizzo di scrittura} - \text{delay}$$

Dato che è possibile che il valore di delay superi quello massimo accettabile di 44100 viene per prima cosa verificata la sua validità, in caso fosse superiore al massimo viene limitato il suo valore forzandolo al valore massimo tramite il multiplexer "*MUX_2*". Viene inoltre evitato che il suo valore sia nullo, questo perché in una condizione simile gli indirizzi di lettura e scrittura combacerebbero e dato che prima di leggere la cella di memoria essa viene scritta si avrebbe un delay effettivo pari al valore massimo.

Nel caso il risultato sia negativo, tramite il multiplexer "*MUX_1*", viene sommato 44100, che equivale alla profondità della memoria per riportarlo nella dinamica corretta.

- Blocco di retroazione dei campioni:

Per comprendere a pieno i calcoli che vengono svolti è prima necessario spiegare le valutazioni fatte sul parallelismo dei dati.

Per il parallelismo dei dati, dopo un'analisi della memoria disponibile sulla FPGA, si è deciso di dedicare 12 bit per ognuno.

I campioni ricevuti su otto bit vengono quindi estesi aggiungendo un bit a destra, per avere una migliore precisione quando si effettuano delle moltiplicazioni, e tre bit a sinistra per aumentare la

dinamica dei campioni in memoria. Infatti, se viene ricevuto un segnale in complemento a due su 8 bit con un valore equivalente di $\pm 0,5$, con l'aggiunta di 3 bit a sinistra si può considerare di avere una dinamica per i calcoli intermedi di ± 4 , questo può essere utile quando vengono sommati molti valori (condizione ottenibile con valori elevati di decay).

Se per esempio vengono sommando diversi valori si arriva ad avere un valore in memoria fuori dalla dinamica di uscita è possibile che moltiplicandolo per un valore di mix sufficientemente basso esso ritorni nella dinamica accettabile. Per evitare overflow nella retroazione della memoria e per riportare l'uscita nella dinamica corretta senza errori vengono utilizzati due stadi di saturazione separati.

La retroazione della memoria è ottenuta collegando la sua uscita al moltiplicatore di decay, il suo risultato viene successivamente sommato al campione in ingresso e prima di essere mandato in memoria quest'ultimo risultato, se necessario, può essere saturato sfruttando il multiplexer "*MUX_3*".

- Blocco di calcolo dell'uscita:

L'uscita della memoria è collegata anche al blocco di calcolo dell'output, esso ha la stessa struttura del blocco di retroazione. Ovvero un moltiplicatore, questa volta che moltiplica il parametro mix, un sommatore con il campione di ingresso e un blocco di saturazione. Qui viene verificato che il segnale sia all'interno della dinamica d'uscita ($\pm 0,5$), se non ci sono problemi di dinamica i bit corretti sono collegati direttamente all'uscita, altrimenti vengono forzati i valori di saturazione del segnale tramite il multiplexer "*MUX_4*".

Sono inoltre presenti due registri, uno in ingresso "*INPUT_REGISTER*" e uno in uscita "*OUTPUT_REGISTER*", per mantenere costanti i valori ricevuti e le uscite ottenute.

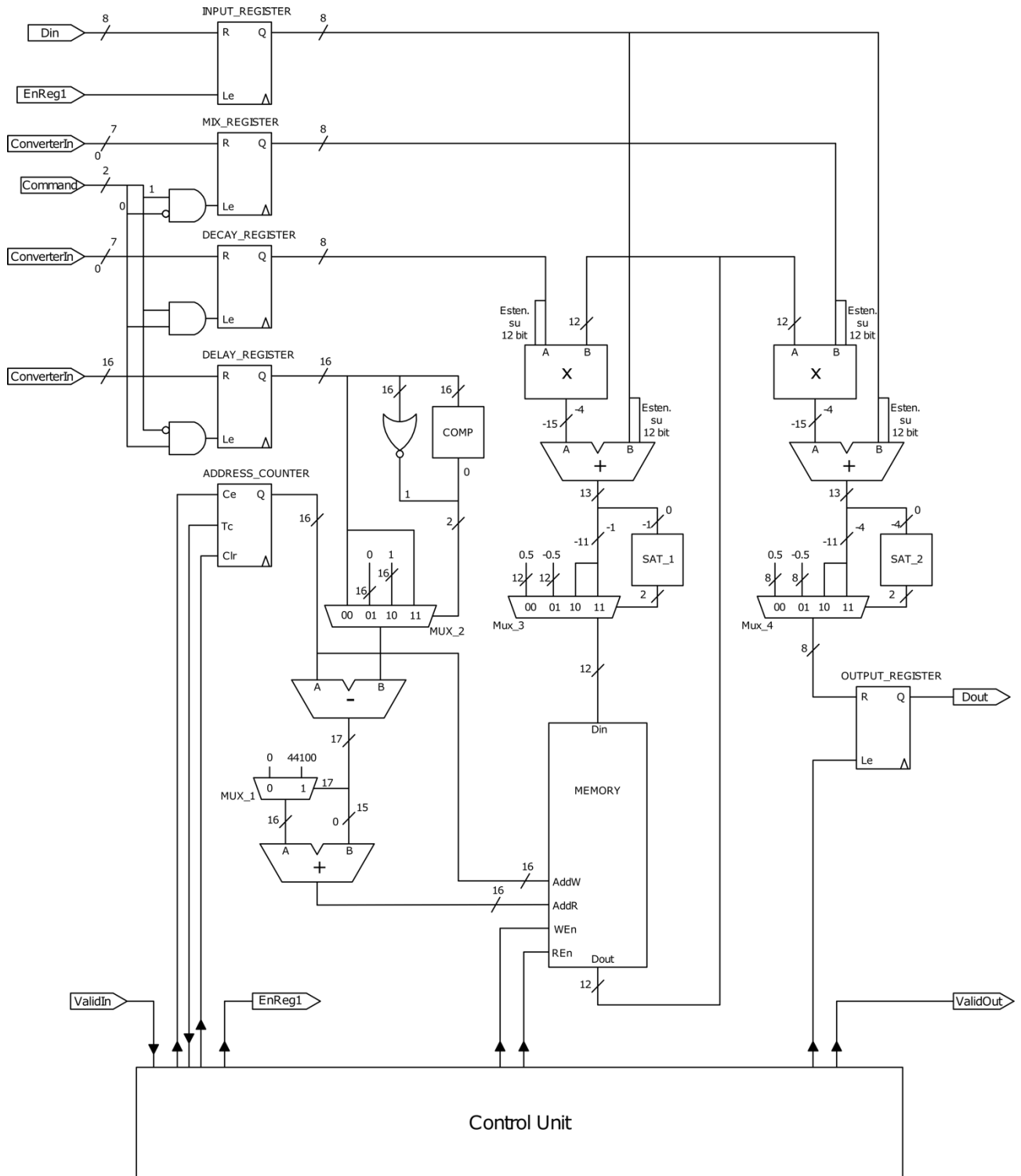


Figura 8 Datapath generatore di eco

3.2. Timing Diagram

Con l'utilizzo di un Timing diagram è possibile a definire il comportamento del circuito.

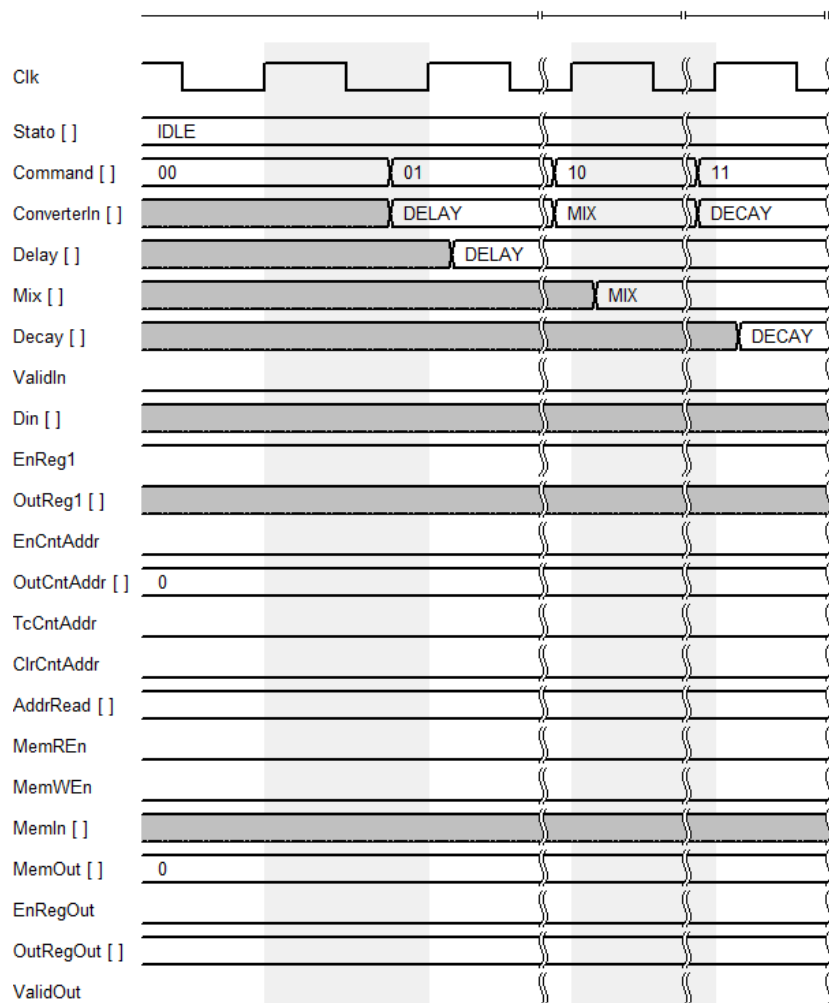


Figura 9 Timing generatore di eco: ricezione parametri

In questa prima parte è possibile osservare come per diversi valori di “*Command*” i valori ricevuti dal convertitore vengano salvati rispettivi registri. Questa operazione viene fatta in modo autonomo dalla rete combinatoria che gestisce i segnali di enable dei registri senza nessun intervento dalla control unit.

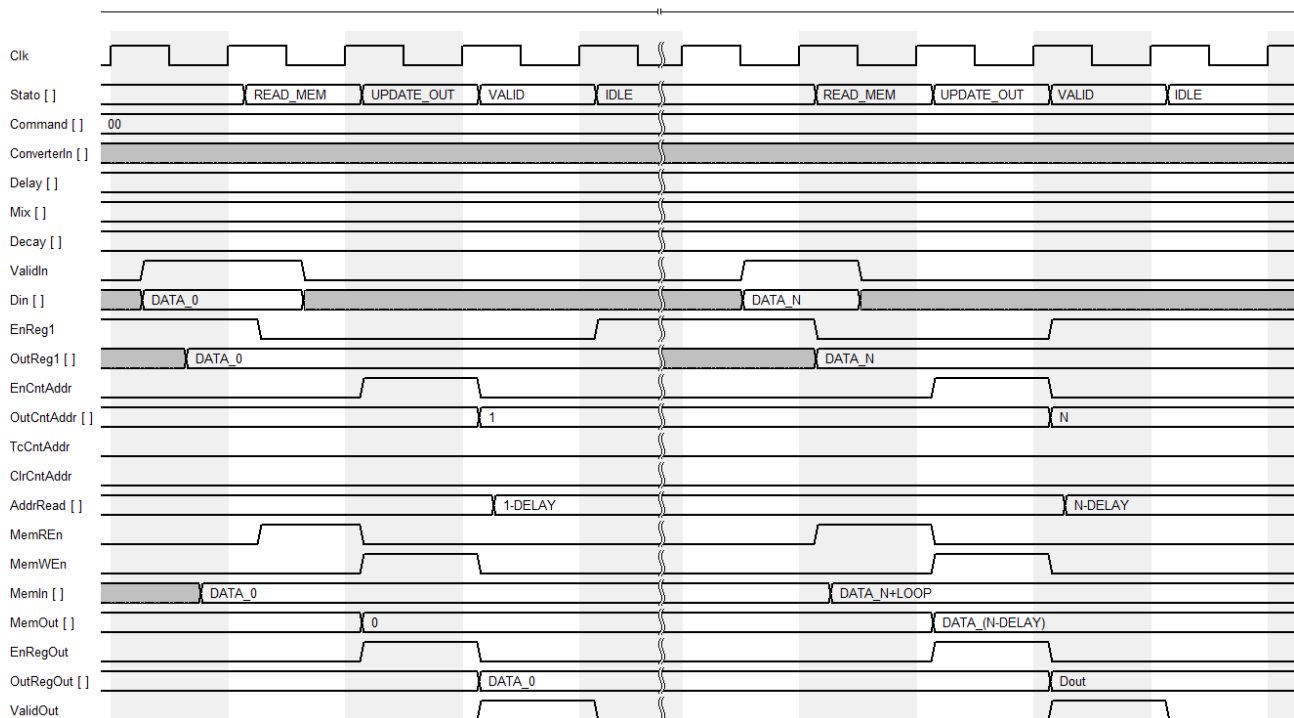


Figura 10 Timing generatore di eco: elaborazione dei campioni

La macchina resta in stato di “IDLE” in cui viene campionato continuamente l’ingresso finché non viene rilevato il segnale “ValidIn”. Nel fronte di salita del clock successivo la control unit passa allo stato “READ_MEM” in cui viene interrotto il campionamento dell’ingresso e viene abilitata la lettura della memoria.

L’uscita della memoria sarà valida solo nel periodo di clock successivo, quindi, dopo l’attesa di un colpo di clock, avviene la transizione allo stato “UPDATE_OUT”.

In tale stato viene abilitato il campionamento del risultato “EnRegOut” e la scrittura in memoria “MemWEn”, in modo tale da poter campionare i risultati delle diverse operazioni al prossimo fronte di salita del segnale clock. Viene inoltre incrementato il contatore degli indirizzi in modo da avere a disposizione gli indirizzi corretti per il prossimo campione.

Lo stato che viene raggiunto immediatamente dopo è quello “VALID”. In questo stato viene segnalato all’esterno che l’output corrente è valido e può essere prelevato. In questo stato è disponibile il valore del contatore “ADDRESS_COUNTER”, dato che i valori assumibili dall’output del contatore sono superiori al più alto indirizzo di memoria (44099) viene usato un segnale di terminal count per segnalare la necessità di resettarlo.

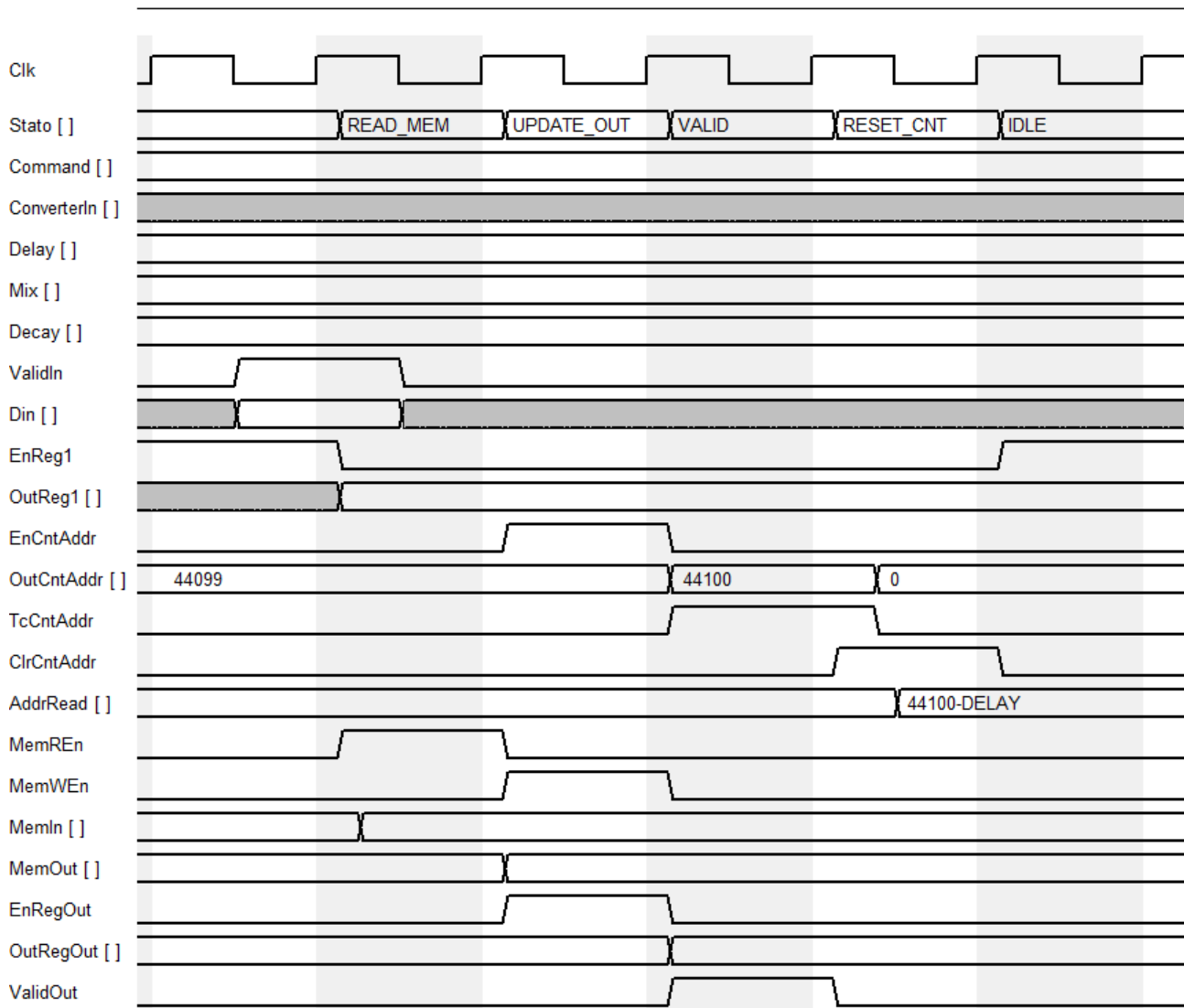


Figura 11 Timing generatore di eco: reset contatore indirizzi

In quest'ultima sequenza è mostrato come il contatore viene resettato nello stato "RESET_CNT" prima di ritornare nello stato di idle.

Tutte le operazioni di saturazione o verifica della validità dell'indirizzo non sono state incluse nei timig diagram per facilitarne la lettura e in quanto vengono eseguite in modo combinatorio e quindi non richiedono nessun intervento dalla control unit.

3.3. Diagramma degli stati

Per rendere più chiare le operazioni effettuate è possibile osservare il diagramma degli stati:

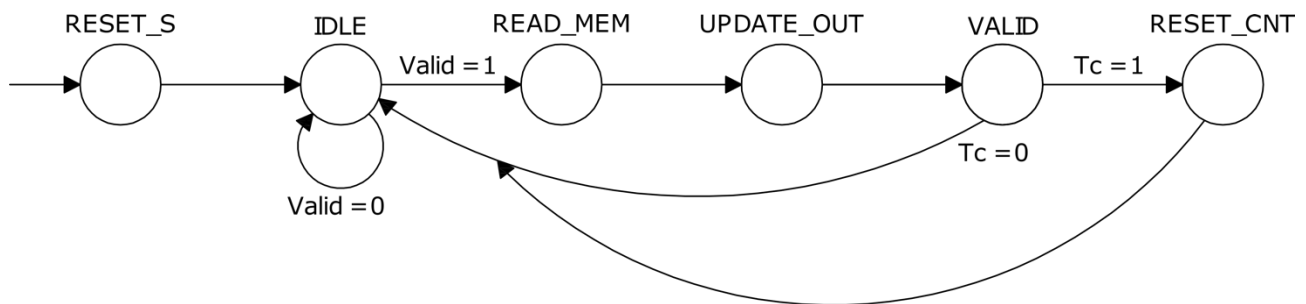


Figura 12 Diagramma degli stati generatore di eco

Dato che molte operazioni vengono eseguite in modo combinatorio la control unit risulta formata da un numero ridotto di stati.

Descrizione degli stati:

- **RESET_S**: stato raggiungibile in maniera asincrona da ogni altro stato a seguito della asserzione del segnale di reset. Viene eseguito l’azzeramento di tutti i registri ed il contatore del datapath.
- **IDLE**: stato in cui la macchina ritorna al termine di ogni calcolo del valore di uscita. Viene campionato continuamente l’ingresso fino a rilevare attivo il segnale di “ValidIn”.
- **READ_MEM**: viene comandata la lettura della memoria.
- **UPDATE_OUT**: vengono svolti tutti i calcoli necessari ad ottenere il nuovo valore da inserire in memoria e quello da mandare in uscita, vengono quindi abilitate la scrittura della memoria ed il campionamento del registro di uscita.
- **VALID**: all’uscita del registro di output viene reso disponibile il campione elaborato, la validità del campione viene segnalata con l’asserzione del segnale “ValidOut”. A seconda del valore del segnale di terminal count del contatore si passa allo stato di idle o di reset del contatore.
- **RESET_CNT**: viene eseguito il reset del contatore degli indirizzi.

4. Analisi risultati

4.1. Simulazione con Modelsim

Prima di caricare il circuito realizzato su una FPGA fisica è necessario eseguire dei test su simulatore e verificare che i risultati rispettino le aspettative. Per fare questo è stato sfruttato il software Modelsim, è stato realizzato un testbench che genera gli stimoli da dare in input al circuito e fornisce gli output che esso genera. Per facilitare la verifica sono stati generati gli stessi segnali mostrati nei timing diagram realizzati in fase di progetto.

Vengono di seguito mostrati i risultati delle simulazioni fatte con Modelsim.

- Convertitore:

Simulazione della ricezione del valore di delay, stessa sezione del timing in **(Figura 3)**

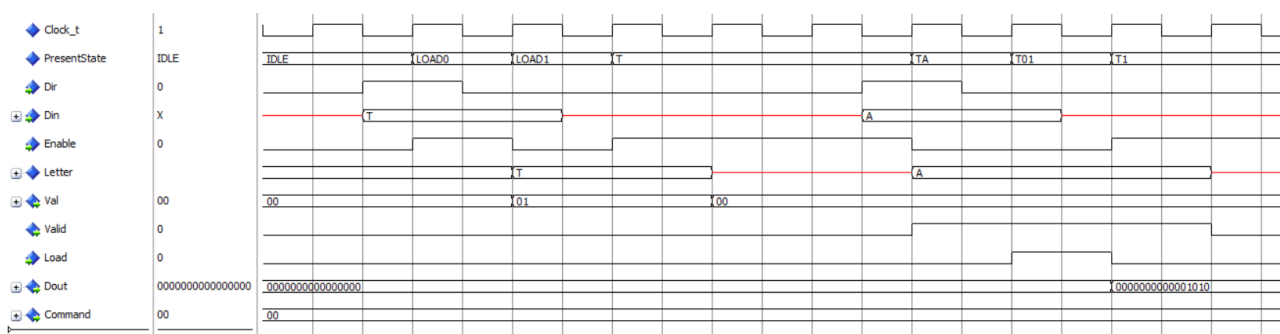


Figura 13 Simulazione convertitore: campionamento dati

Simulazione dell'invio del valore di delay ricevuto, stessa sezione del timing in **(Figura 4)**

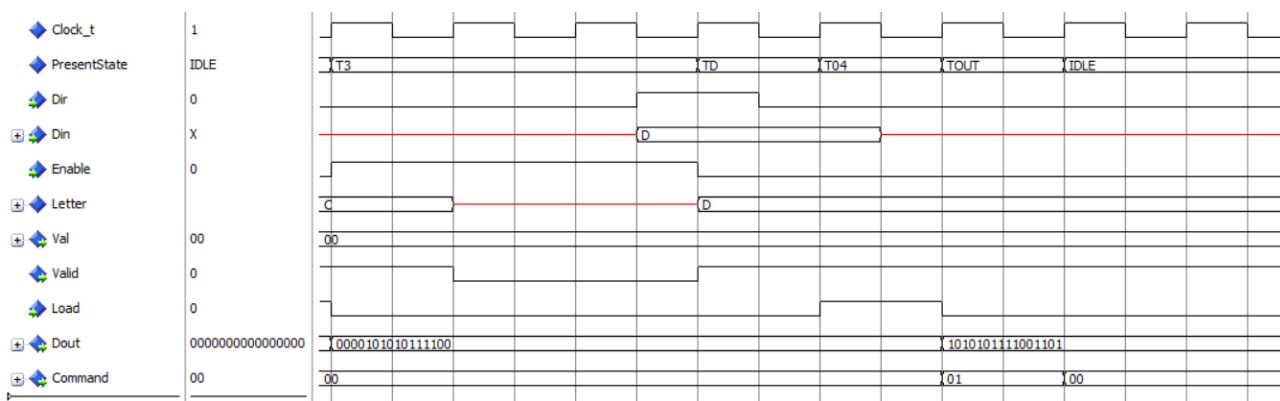


Figura 14 Simulazione convertitore: invio dei dati

- Generatore di eco:

Simulazione della ricezione dei parametri dell'eco, stessa sezione del timing in (*Figura 9*)

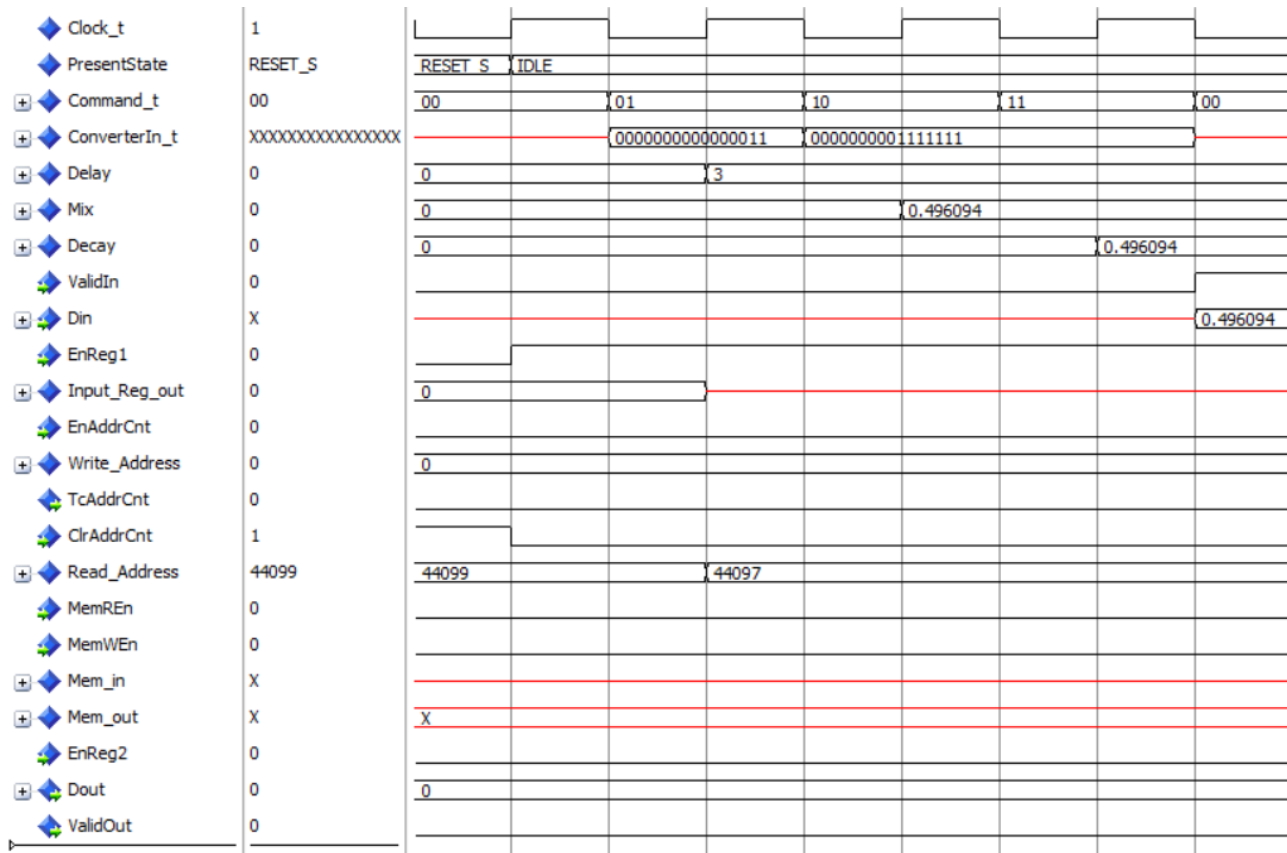


Figura 15 Simulazione generatore di eco: ricezione parametri

Simulazione dell'elaborazione e output dei campioni, stessa sezione del timing in **(Figura 10)**

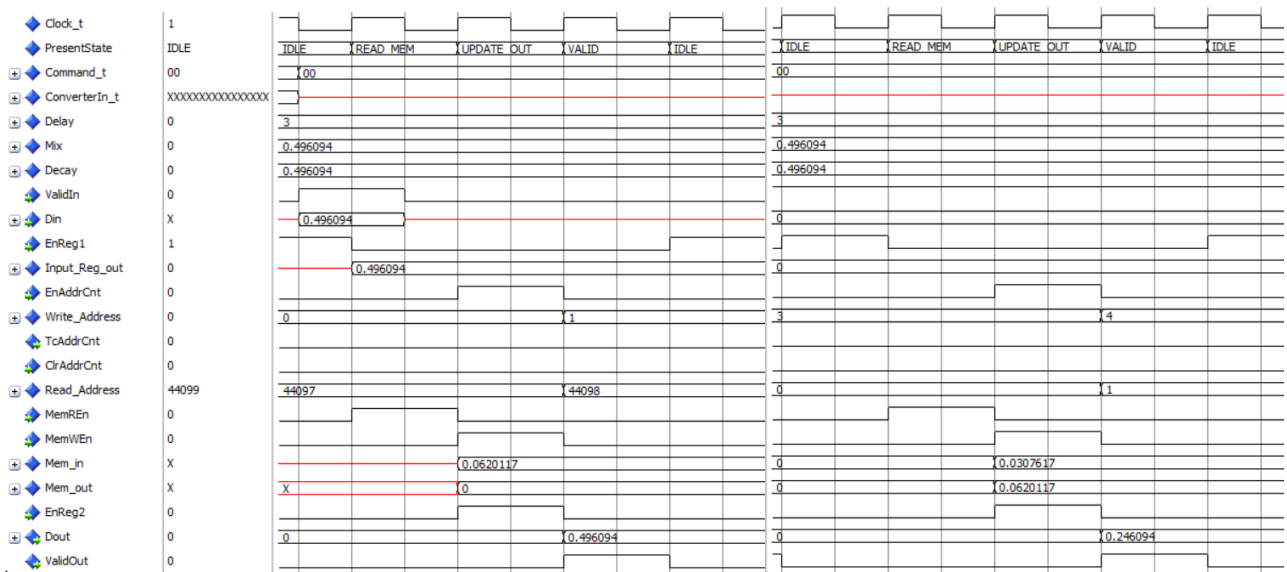


Figura 16 Simulazione generatore di eco: elaborazione dedi campioni

Simulazione del reset del contatore di indirizzi, stessa sezione del timing in *(Figura 11)*

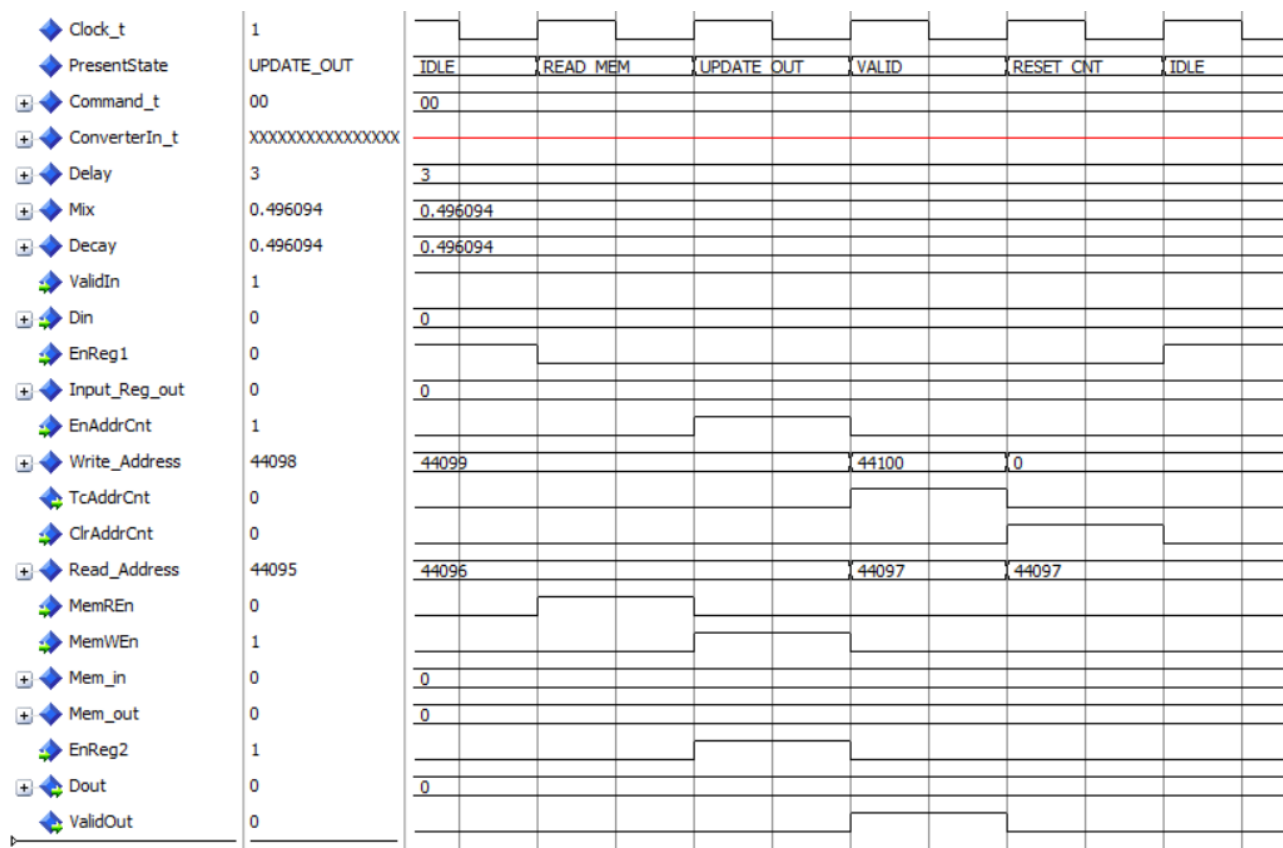


Figura 17 Simulazione generatore di eco: reset contatore indirizzi

Come è possibile notare dalle simulazioni mostrate e dai relativi timing diagram il dispositivo realizzato mostra il comportamento atteso.

4.2. Sintesi con Quartus

Per la sintesi è stato utilizzato il software Quartus. A partire dai codici VHDL esso fornisce tutte le informazioni necessarie alla realizzazione su FPGA, in particolare sono importanti la quantità di elementi hardware usati e le caratteristiche di timing per conoscere l'area necessaria e la massima frequenza di funzionamento.

Family	Cyclone 10 LP
Device	10CL025YE144C8G
Timing Models	Final
Total logic elements	397 / 24,624 (2 %)
Total registers	192
Total pins	51 / 77 (66 %)
Total virtual pins	0
Total memory bits	529,200 / 608,256 (87 %)
Embedded Multiplier 9-bit elements	4 / 132 (3 %)
Total PLLs	0 / 2 (0 %)

Figura 18 Elementi logici usati dal circuito

Dall'analisi del percorso critico risulta che, come prevedibile, il percorso più lungo è quello di retroazione della memoria. Infatti, esso dopo essere uscito dalla memoria deve attraversare un moltiplicatore, un sommatore ed infine un multiplexer prima di tornare nella porta di ingresso.

Il tempo necessario per completare il percorso è di 22.4 ns, di conseguenza la massima frequenza di lavoro della macchina è di circa 44.5 MHz.

4.3. Funzionamento su FPGA

Per poter lavorare con l’FPGA reale è stato fornito un software che sfrutta il microcontrollore a bordo della scheda VirtLAB. In questo modo è possibile, tramite una porta di comunicazione su PC, trasmettere dei dati al mcu che a sua volta li trasmetterà alla FPGA.

Per testare il sistema è stata fatta una prima prova in cui è stato impostato un delay di pochi campioni e i dati sono stati inseriti manualmente in modo da poter verificare la correttezza dei risultati ad occhio.

Successivamente sono stati sfruttati dei file audio per eseguire delle prove più complete. Dando in ingresso al sistema tutti i campioni che compongono una traccia audio è possibile ottenere il risultato prodotto come sequenza dei campioni di uscita forniti dal sistema.

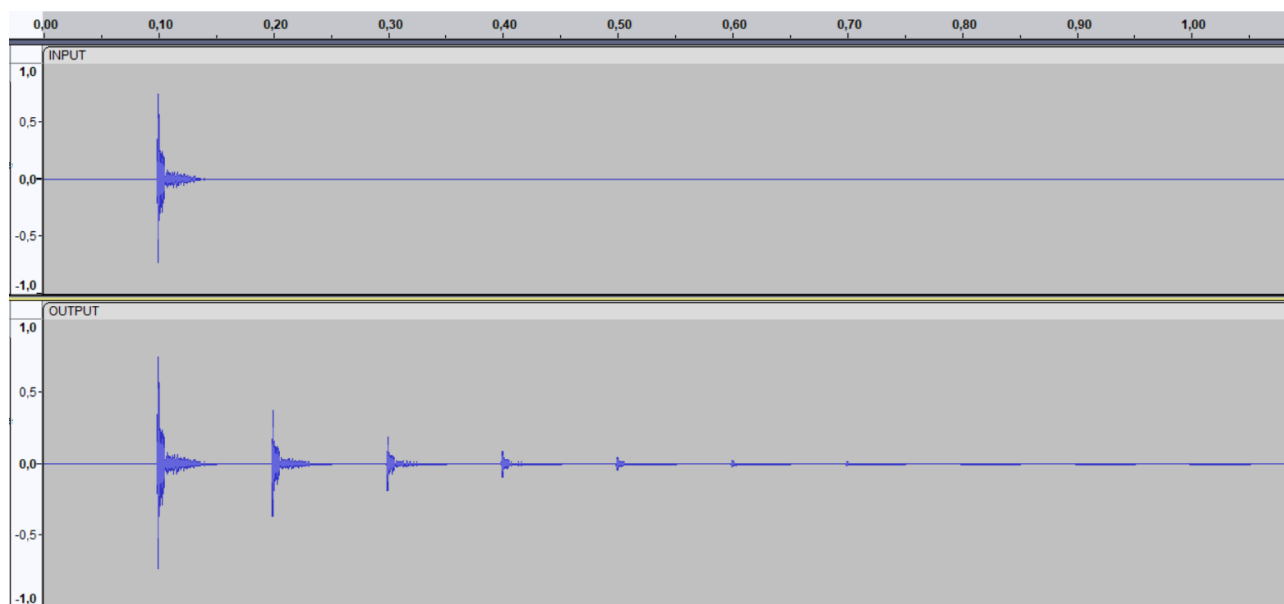


Figura 19 Forme d'onda ottenute (Delay = 5000, Decay = 0.5, Mix = 0.5)

Dando in ingresso al sistema un impulso di piccola durata è possibile vedere come esso venga replicato ad intervalli di tempo regolari di larghezza circa un decimo di secondo e ad ogni ripetizione si ha un dimezzamento del valore.

È però visibile che, anche dopo diverso tempo dall’impulso, siano ancora presenti delle piccole variazioni dell’uscita. Nel circuito si è scelto di eseguire tutti i calcoli minimizzando le approssimazioni per ridurre la distorsione del suono, questo significa che il valore negativo più piccolo rappresentabile non potrà mai annullarsi completamente e quindi continuerà a presentarsi in uscita. Sarebbe possibile eliminare questo disturbo approssimando il valore negativo minimo forzando uno zero.

Al fine di verificare che il sistema non introduca distorsioni del segnale è stato osservato lo spettro ottenuto inviando in ingresso uno sweep di frequenza compreso tra 100 e 20000 Hz. Vengono impostati dei valori di decay e mix elevati in modo da dover ottenere in uscita delle repliche con stesse caratteristiche del segnale originario.

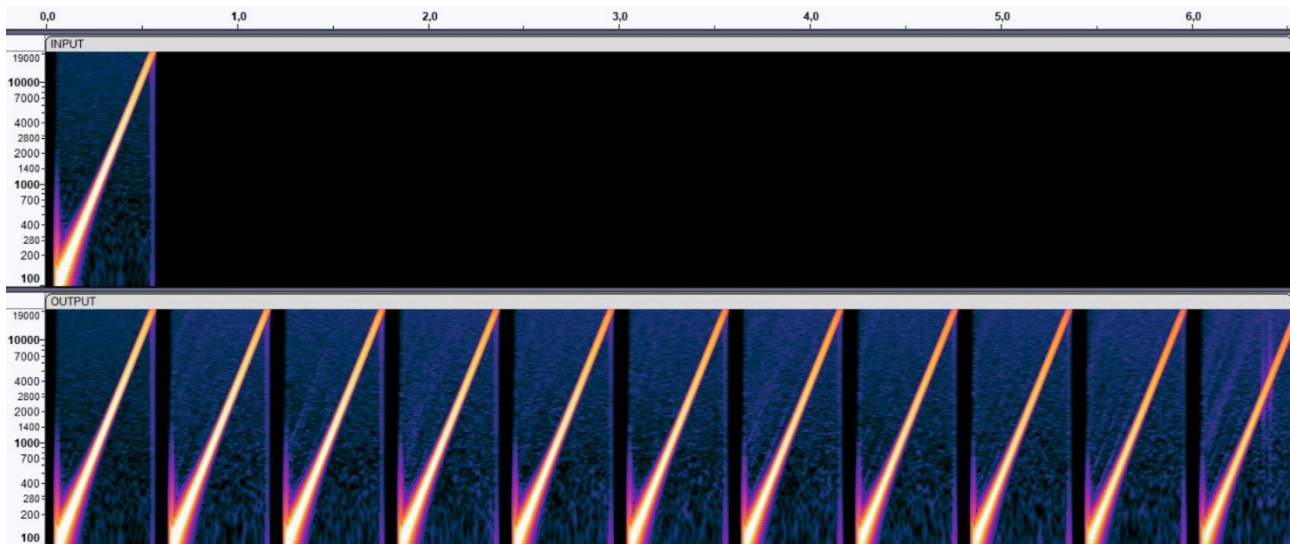


Figura 20 Spettro ottenuto (Delay = 26460, Decay = 0.9, Mix = 0.9)

Si può notare come anche dopo più ripetizioni lo spettro resti invariato. Il sistema è quindi in grado di replicare più volte un segnale con frequenza compresa nel range udibile senza introdurre delle distorsioni.