**POLITECNICO DI TORINO**

Laurea Magistrale in Ingegneria Elettronica

Corso di Sistemi Digitali Integrati

**Relazione Progetto San Silvestro**



Professore Studenti

Maurizio Zamboni Cavagnetto Matteo:290149

Cicero Rosalia: 282502

Rizzello Stefano:288013

Sommario

[Introduzione 4](#_Toc62383239)

[Data Flow Diagram 5](#_Toc62383240)

[Scheduling 8](#_Toc62383241)

[Tempo di vita delle variabili 9](#_Toc62383242)

[Data Path Butterfly 10](#_Toc62383243)

[Struttura dell’unità di controllo 10](#_Toc62383244)

[La struttura delle uROM 13](#_Toc62383245)

[Criteri da rispettare 15](#_Toc62383246)

[Scelte realizzative 16](#_Toc62383247)

[Segnali di controllo 16](#_Toc62383248)

[Rounding 18](#_Toc62383249)

[Timing 19](#_Toc62383250)

[Timing esecuzione continua 19](#_Toc62383251)

[Timing esecuzione singola 20](#_Toc62383252)

[Timing relativo agli ingressi 21](#_Toc62383253)

[Timing relativo alle uscite 22](#_Toc62383254)

[Script MATLAB per testare la FFT per ogni singolo campione 23](#_Toc62383255)

[Appendice 27](#_Toc62383256)

# Introduzione

L’esercitazione finale consiste nella progettazione di un’unità di elaborazione che esegua la FFT secondo l’algoritmo di Cooley Tukey.

La realizzazione della Butterfly, componente base e fondamentale, consiste nell’effettuare somme e moltiplicazioni su numeri complessi A, B e basandosi sul seguente schema:

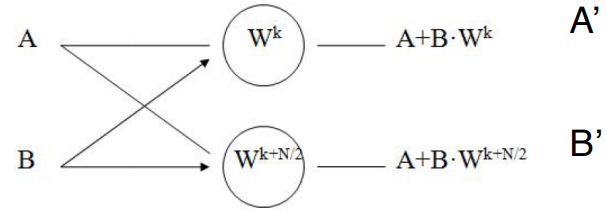


Figura : Struttura Butterfly

La Butterfly prenderà due valori complessi generici e ne produrrà delle uscite date da:

e

che possono essere espresse esplicitando la parte reale e la parte immaginaria come:

I dati che riceve in ingresso saranno dati da:

Mentre i termini WK differiscono di N/2 per cui saranno opposti.

Il risultato in uscita sarà dato quindi da dei risultati parziali, i quali identificano l’algoritmo da implementare secondo i seguenti passi:

Il progetto di tale struttura deve essere eseguito rispettando le seguenti specifiche:

* I dati A, B e definiti in forma frazionaria (-1<dato<1) in complemento a 2 su 20 bit utilizzando una tecnica mista tra “Guard bit” e “Unconditional Block Floating Point Scaling”.
* Si utilizzi un moltiplicatore con un livello di pipeline, in grado di effettuare la moltiplicazione tra due dati e la moltiplicazione di un dato per 2 tramite uno shift.
* Per le somme e le sottrazioni si ipotizzi di avere due blocchi sommatore/sottrattore in cui un segnale di controllo discrimina l’operazione da fare, e nello specifico anche il sommatore dovrà avere un livello di pipe.
* Si ipotizzi di avere le operazioni interne senza troncamento, ma che questo avvenga sul dato di uscita della Butterfly con la tecnica ROUND TO NEAREST EVEN, potendo usufruire di tutto l’hardware necessario per realizzare.
* L’unità di controllo deve essere definita mediante la tecnica della microprogrammazione con un sequencer con indirizzamento esplicito e tecnica del LATE STATUS.

# Data Flow Diagram

Per la realizzazione del progetto sono state realizzate due Contro Unit: la prima CU\_TOP, mostrata in Figura 2, atta a coordinare i 4 livelli di Butterfly, l’arrivo del segnale di start, la modalità in sequenza in cui può lavorare la macchina ed inoltre la fine dell’esecuzione; la seconda CU\_BF, mostrata in Figura 3, si occupa invece della gestione dell’algoritmo della singola Butterfly. Quindi vengono implementate quattro CU\_BF una per ogni livello di BF e una CU\_TOP.

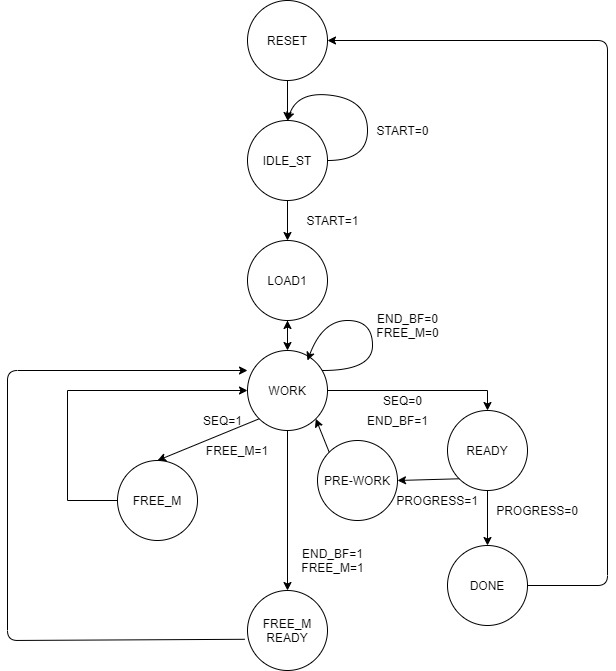
****

Figura : CU Top Level

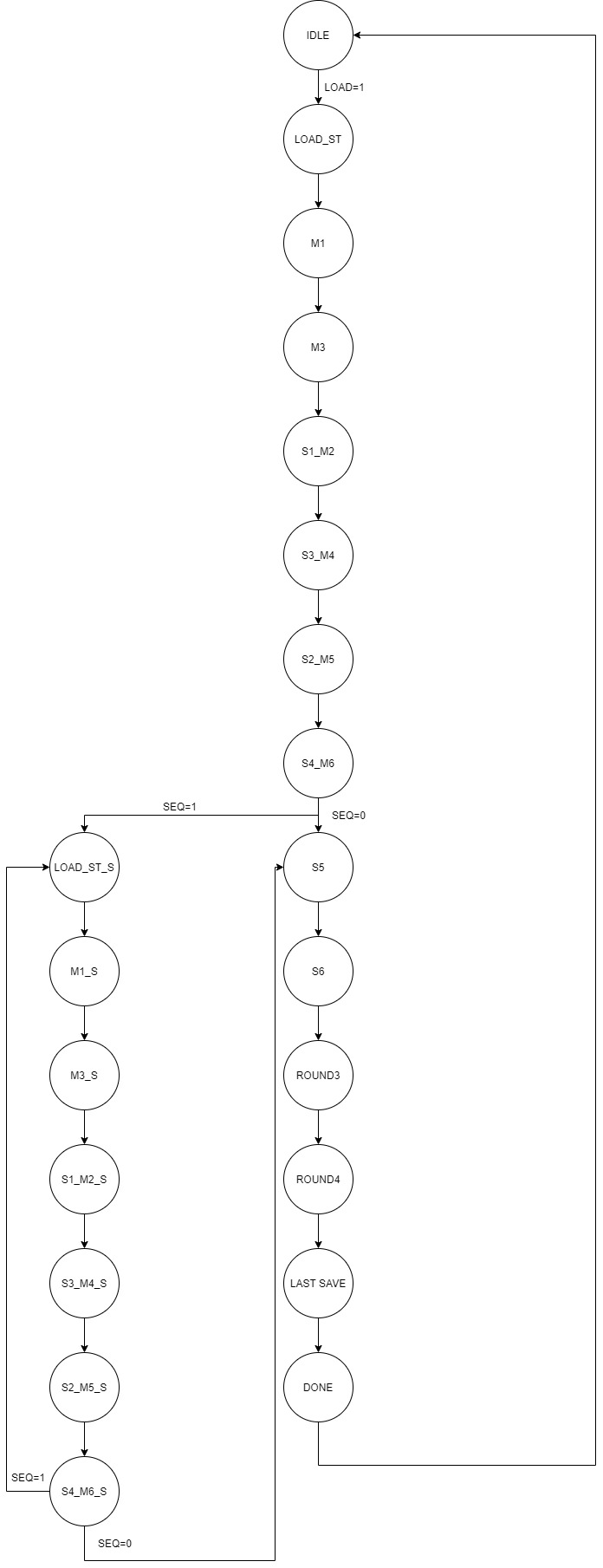
****

Figura : CU Butterfly

Nel dettaglio la CU\_TOP quando evolve negli stati LOAD o FREE\_M (caso per la sequenza) ha il compito di caricare i campioni all’interno delle BF di primo livello; a differenza, il caricamento dei dati tra gli altri livelli (secondo, terzo, quarto), avviene direttamente con segnali tra le CU\_BF. In dettaglio il segnale END\_BF del livello precedente viene usato per caricare all’ interno del successivo i campioni appena elaborati. Il segnale END\_BF del quarto livello viene invece letto dalla CU\_TOP per andare nello stato di READY e comunica all’ esterno che il dato è pronto in uscita alla fft16.

Nel caso si voglia fare delle elaborazioni in sequenza la CU\_BF del primo livello giunta al sesto colpo di clock genera un segnale FREE\_M per la CU\_TOP, la quale, se ha anche il segnale di sequenza attivo (segnale SEQ), andrà nello stato FREE\_M e provvederà a caricare nuovamente i dati nella BF di primo livello.

Quando si entra nella modalità in sequenza, la macchina eseguirà degli stati appositi al processamento di due campioni contemporaneamente. Questi sono gli stati con pedice ‘s’, presenti nella Figura 3, nel ramo di sinistra.

Quindi, oltre a generare i segnali per l’elaborazione del nuovo dato, dovranno generare ancora i segnali per completare i calcoli del campione caricato in precedenza.

Es: M1\_S dovrà generare i comandi di M1 per il nuovo dato e i comandi di S6 per il campione precedente.

I segnali attivi ad ogni stato sono rappresentati nelle tabelle.

Mentre i segnali di controllo vengono descritti successivamente.

# Scheduling

Analizzate le specifiche di progetto, lo Scheduling individuato per realizzare l’algoritmo della Butterfly è il seguente:

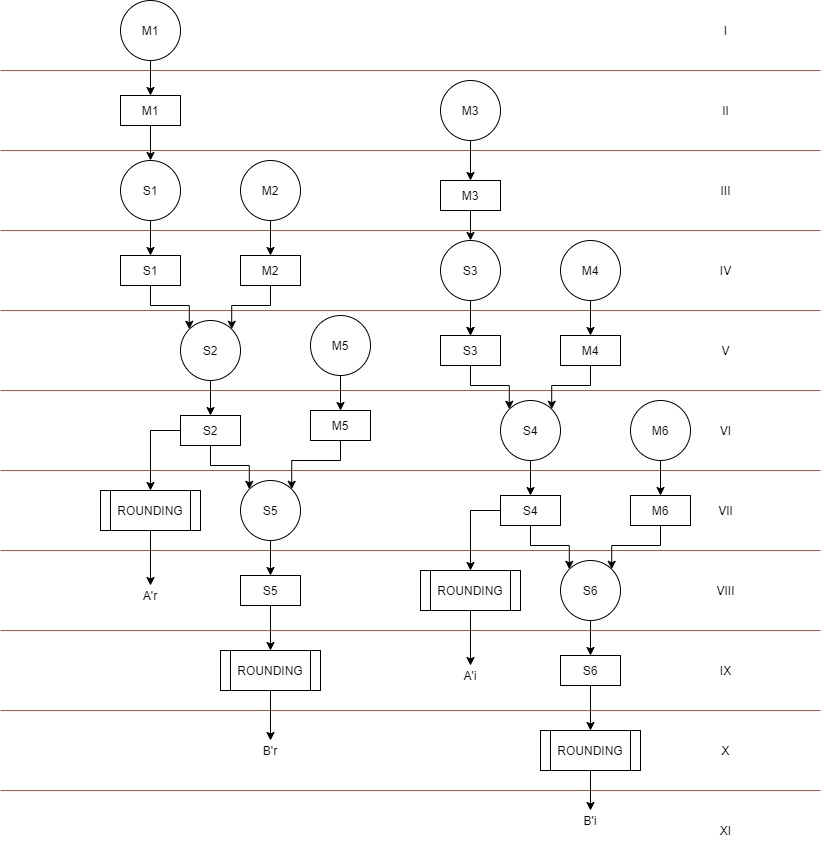


Figura : Scheduling

Come possibile osservare in Figura 4, al primo colpo di clock viene effettuata la moltiplicazione M1, il quale risultato come richiesto da specifica è disponibile al colpo di clock dopo ma, avendo inserito un registro di pipe tra il moltiplicatore e il sommatore, il risultato sarà disponibile per la somma al terzo colpo di clock. Lo stesso avviene per le altre moltiplicazioni.

Come è possibile osservare, non si verificheranno mai due moltiplicazioni nello stesso colpo di clock, rispettando la specifica del singolo moltiplicatore.

Il risultato della somma sarà una delle uscite della Butterfly, per cui avendo il risultato sul registro di pipe al VI colpo di clock, al colpo di clock successivo subirà l’arrotondamento e andrà in uscita.

Lo stesso avviene per i risultati in uscita dalla somma .

Lo Scheduling scelto è stato realizzato utilizzando la configurazione ALAP (As Late As Possible), infatti come è possibile osservare in Figura 4, si è scelto un ordine di realizzazione delle moltiplicazioni in cui dopo la prima moltiplicazione M1, viene realizzata la moltiplicazione M3, e successivamente le moltiplicazioni M2, M4, M5 ed M6, nei colpi di clock che precedono il loro impiego nelle rispettive somme. Questa soluzione è stata realizzata poiché si ha a disposizione un solo moltiplicatore, per cui si potrà avere al massimo una sola moltiplicazione per colpo di clock. Al sesto colpo di clock, il moltiplicatore sarà completamente libero e la Butterfly potrà accettare nuovi dati.

Per quanto riguarda le somme invece, pur avendo a disposizione due blocchi sommatore\sottrattore, si è scelto di utilizzarne solo uno, in cui attraverso due multiplexer si ha la possibilità di selezionare quale dato inviare al morsetto invertente per realizzare la sottrazione.

# Tempo di vita delle variabili

Di seguito è riportato il diagramma del tempo di vita delle variabili.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **CK1** | **CK2** | **CK3** | **CK4** | **CK5** | **CK6** | **CK7** | | **CK8** | | **CK9** | **CK10** | **CK11** |
| **Ar** |  |  |  |  |  |  |  | |  | |  |  |  | |
| **Ai** |  |  |  |  |  |  |  | |  | |  |  |  | |
| **Br** |  |  |  |  |  |  |  | |  | |  |  |  | |
| **Bi** |  |  |  |  |  |  |  | |  | |  |  |  | |
| **Wr** |  |  |  |  |  |  |  | |  | |  |  |  | |
| **Wi** |  |  |  |  |  |  |  | |  | |  |  |  | |
| **M1** |  |  |  |  |  |  |  | |  | |  |  |  | |
| **M2** |  |  |  |  |  |  |  | |  | |  |  |  | |
| **M3** |  |  |  |  |  |  |  | |  | |  |  |  | |
| **M4** |  |  |  |  |  |  |  | |  | |  |  |  | |
| **M5** |  |  |  |  |  |  |  | |  | |  |  |  | |
| **M6** |  |  |  |  |  |  |  | |  | |  |  |  | |
|  |  |  |  |  |  |  |  | |  | |  |  |  | |
|  |  |  |  |  |  |  |  | **R** | **OUT** | |  |  |  | |
|  |  |  |  |  |  |  |  | |  | |  |  |  | |
|  |  |  |  |  |  |  |  | |  | **R** | **OUT** |  |  | |
|  |  |  |  |  |  |  |  | |  | | **R** | **OUT** |  | |
|  |  |  |  |  |  |  |  | |  | |  | **R** | **OUT** | |

Per la realizzazione di tale algoritmo all’ interno della BF è stato scelto di salvare le variabili Ai, Ar, Bi, Br, Wi, Wr su un Register File a sei campi, mentre dal diagramma del tempo di vita delle variabili si è dedotto che sono necessari solamente un registro in cui salvare i risultati delle somme e un registro in cui salvare i risultati delle moltiplicazioni.

In tabella è stata evidenziata in rosso l’operazione di Rounding che viene eseguita sugli stati in uscita dalla Butterfly, mentre internamente si procede senza nessun troncamento. Come richiesto, viene realizzata la tecnica HALF UP ROUND TO NEAREST EVEN, che viene eseguita in un apposito colpo di clock.

Per la gestione dei valori di WR WI si è scelto di implementare al livello di top entity (FFT16) dei registri fissi denominati STONE, in cui vengono memorizzati solamente i valori positivi, dai quali tramite opportuna logica si è ricavato anche il valore negativo.

# Data Path Butterfly

Dopo aver realizzato ed esaminato lo Scheduling e il tempo di vita delle variabili, si è proceduto con l’implementazione del progetto, andando a realizzare il Data Path dell’intera Butterfly mostrato in Figura 5.

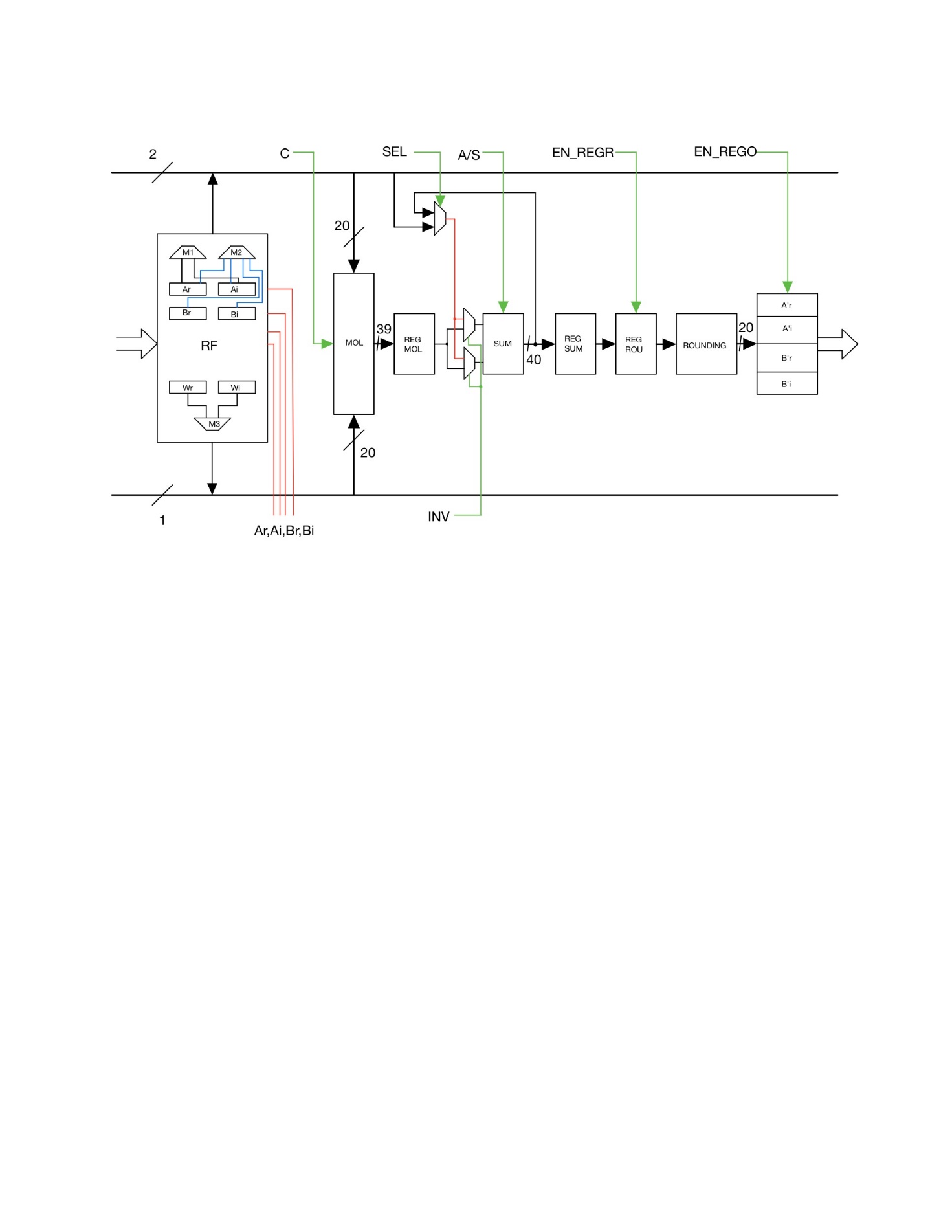


Figura : Data Path della singola BF

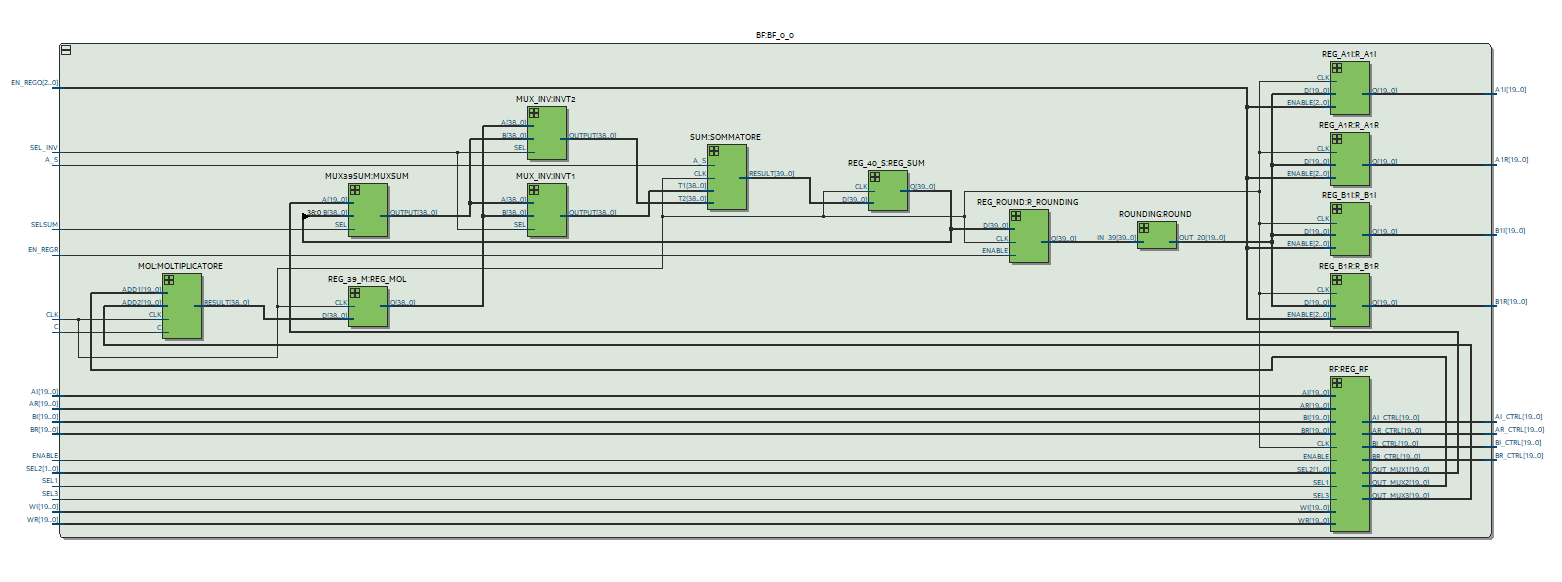


Figura : Datapath Butterfly

Per quanto riguarda l’ottimizzazione del numero dei BUS globali si è deciso di realizzare due BUS visibili nella parte superiore dello schema e uno nella parte inferiore, condiviso tramite MPX dai registri WR e WI. I BUS superiori vengono invece condivisi dai registri AR, AI, BR, BI il primo con un MPX a due vie e l’altro con un MPX a quattro vie.

# Struttura dell’unità di controllo

Le figure 7 e 10 mostrano lo schema di base usato per la realizzazione delle Control Unit secondo la logica del LATE STATUS. Tale soluzione prevede di dividere le righe della memoria in più campi in base al numero di salti richiesti dall’ algoritmo (nel caso specifico 2 per CU\_BF e 4 per CU\_TOP) e di discriminare tali campi utilizzando i 2 bit o LSB meno significativi tramite multiplexer. Così facendo l’indirizzo di riga è noto a priori a prescindere dai salti e ciò permette di anticipare la procedura di lettura in memoria. I 2 bit meno significativi o LSB, che funzionano come selettore del MPX sono invece gestiti da un blocco di logica combinatoria che è funzione degli ingressi dalla CU.

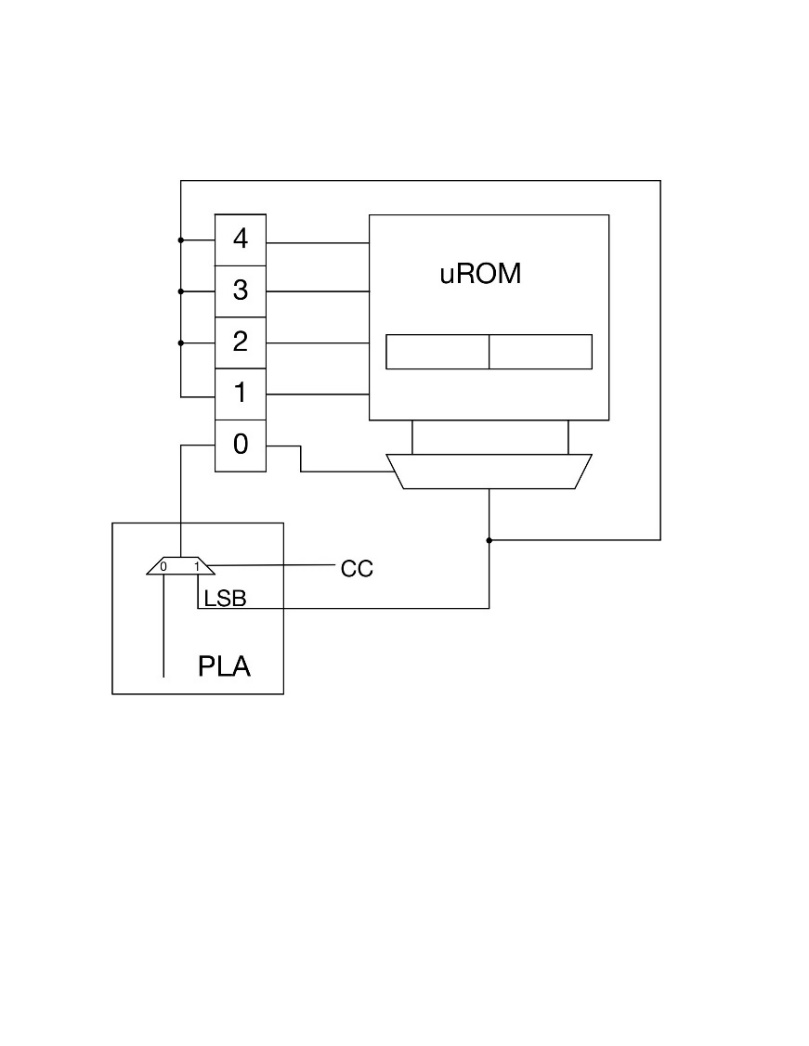


Figura : CU Butterfly

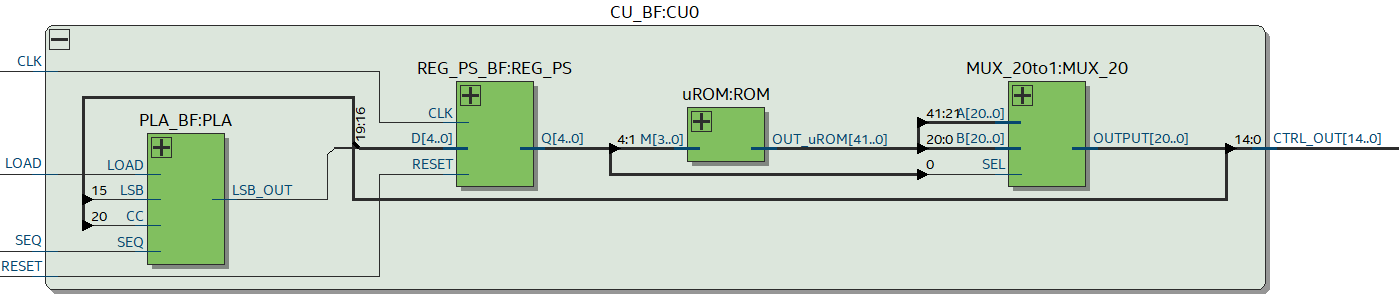


Figura :CU Butterfly

La Figura 8, presa dallo strumento netlist viewer di Quartus, mostra l’implementazione del blocco.

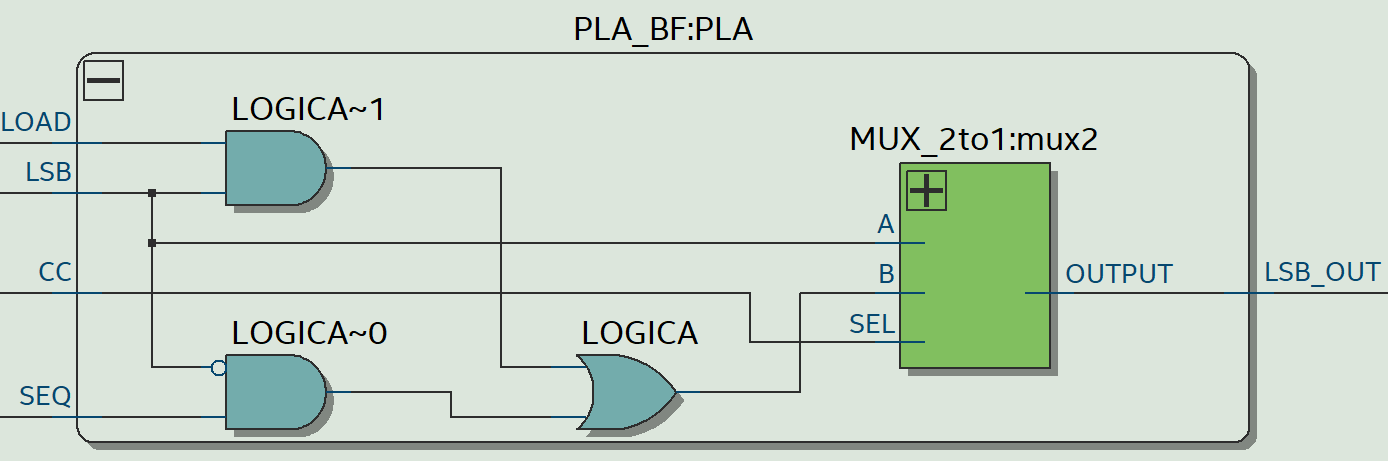


Figura : PLA Butterfly

L’immagine in Figura 9 mette in evidenza la logica interna del blocco PLA che è funzione degli ingressi.

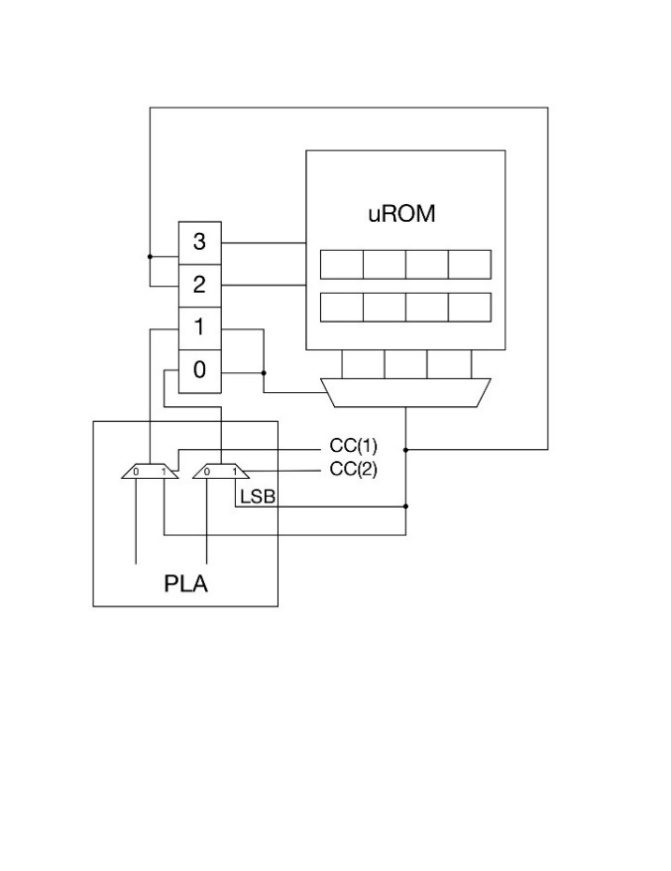


Figura : CU Top Level

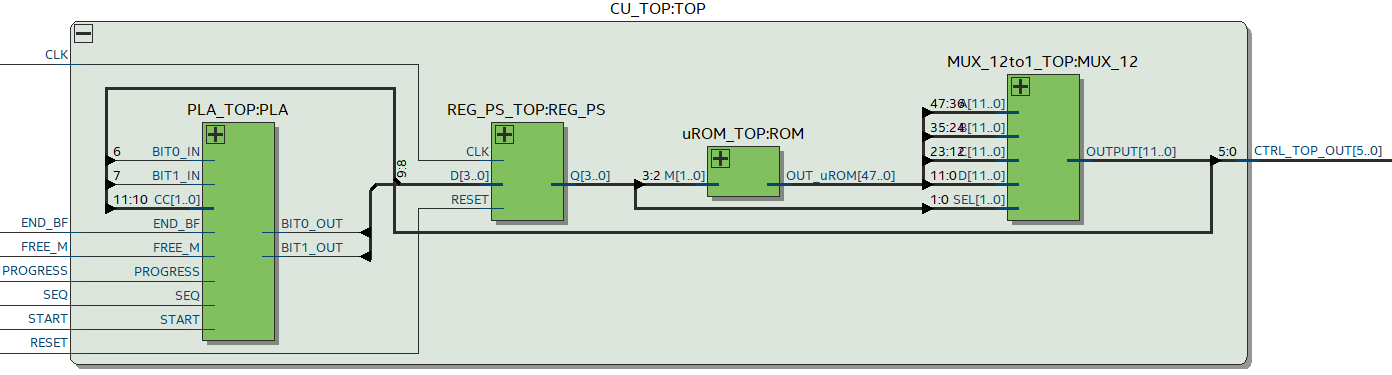


Figura :CU Top Level

La Figura 11 mostra l’implementazione del blocco su Quartus.

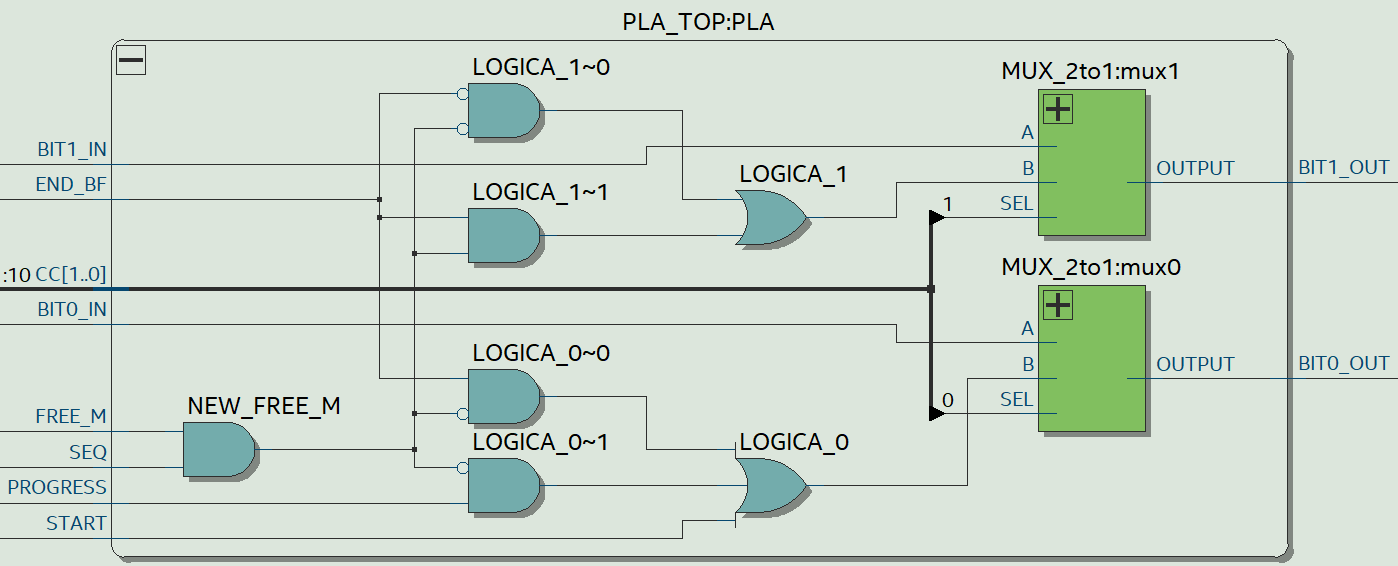


Figura : PLA Top Level

La Figura 12 indica la logica combinatoria funzione degli ingressi del blocco PLA della CU\_TOP.

Tale logica è stata sintetizzata secondo le seguenti mappe:

|  |  |  |  |
| --- | --- | --- | --- |
| **NEW\_FREE\_M** | **END\_BF** | **NEXT STATE** | **BIT 1** |
| 0 | 0 | WORK | 1 |
| 0 | 1 | READY | 0 |
| 1 | 0 | FREE\_M | 0 |
| 1 | 1 | FREE\_M\_READY | 1 |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **NEW\_FREE\_M** | **END\_BF** | **START** | **PROGRESS** | **NEXT STATE** | **LSB(BIT 0)** |
| 0 | 0 | 0 | 0 | DONE/IDLE | 0 |
| 0 | 0 | 0 | 1 | WORK | 1 |
| 0 | 0 | 1 | 0 | LOAD | 1 |
| 0 | 0 | 1 | 1 |  |  |
| 0 | 1 | 0 | 0 | READY | 1 |
| 0 | 1 | 0 | 1 | READY | 1 |
| 0 | 1 | 1 | 0 |  |  |
| 0 | 1 | 1 | 1 |  |  |
| 1 | 0 | 0 | 0 |  |  |
| 1 | 0 | 0 | 1 | FREE\_M | 0 |
| 1 | 0 | 1 | 0 |  |  |
| 1 | 0 | 1 | 1 |  |  |
| 1 | 1 | 0 | 0 |  |  |
| 1 | 1 | 0 | 1 | FREE\_M\_READY | 0 |
| 1 | 1 | 1 | 0 |  |  |
| 1 | 1 | 1 | 1 |  |  |

Le righe evidenziate in Rosso indicano una combinazione degli ingressi invalida che non si potrà verificare in quanto come spiegato nel paragrafo dei controlli la macchina sarà sensibile al segnale di start solamente nello stato di IDLE e in tale stato non è possibile avere PROGRESS ad 1, che viene infatti settato nello stato di LOAD. Questo ha reso possibile una notevole semplificazione nella gestione della logica combinatoria della PLA. Inoltre, FREE\_M e SEQ avendo significato solo quando sono entrambi ad 1, per ridurre il numero di variabili in ingresso, sono stati posti a prescindere in AND, in modo da gestire la Mappa con la sola variabile FREE\_M. In dettaglio FREE\_M indica che il livello di BF ha il moltiplicatore libero, ma solo se c’è attiva anche la richiesta di sequenza (SEQ) la CU\_TOP dovrà provvedere a caricare i nuovi campioni al primo livello di BF.

# La struttura delle uROM

La uROM della CU\_TOP per ogni riga (0,1,2 la numero 3 non è utilizzata) presenta una configurazione a 4 campi per effettuare i salti a 4 vie che viene riportata in Figura 13 .

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Indice di riga** | **2 LSB 00** | **2 LSB 01** | **2 LSB 10** | **2 LSB 11** |
| 00 | IDLE | LOAD | XXXXX | RESET |
| 01 | FREEM | READY | FREEMR | WORK |
| 10 | XXXX | XXXX | DONE | PREWORK |
| 11 | XXXX | XXXX | XXXXX | XXXX |

Figura : Struttura μROM 4 vie

In questo caso invece la uROM della CU\_BF presenta solamente due campi pari e dispari in quanto sono previsti solamente salti a due vie nell’algoritmo.

|  |  |  |
| --- | --- | --- |
| **Indice di riga** | **LSB 0** | **LSB 1** |
| 0000 | IDLE | LOAD\_ST |
| 0001 | M1 | M3 |
| 0010 | S1\_M2 | S3\_M4 |
| 0011 | S2\_M5 | S4\_M6 |
| 0100 | S5 | LOAD\_ST\_S |
| 0101 | S6 | ROUND3 |
| 0110 | ROUND4 | LAST\_SAVE |
| 0111 | M1\_S | M3\_S |
| 1000 | S1\_M2\_S | S3\_M4\_S |
| 1001 | S2\_M5\_S | S4\_M6\_S |
| 1010 | DONE | XXXXXXX |

Figura : Struttura della uROM CU\_BF

In tabella, Figura 15, è mostrato lo schema organizzativo della riga della μROM della singola BF in cui è presente un bit per la condition code, un campo per la codifica del next state e a seguire i segnali di controllo. Le X stanno ad indicare il valore Undefined implementato su Quartus.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **PRESENT STATE** | **CC** | **NEXT**  **STATE** | **SEL**  **INV** | **SEL**  **SUM** | **SEL**  **1** | **SEL**  **2** | **SEL**  **3** | **C** | **A/S** | **EN**  **REGS** | **EN**  **REGR** | **EN**  **REGO** | **FREE\_M** | **END\_BF** |
| **IDLE** | 1 | 00000 | 0 | X | X | X | X | X | X | 0 | 0 | 000 | 0 | 0 |
| **LOAD\_ST** | 0 | 00010 | 0 | X | X | X | X | X | X | 0 | 0 | 000 | 0 | 0 |
| **M1** | 0 | 00011 | 0 | X | X | 10 | 0 | 1 | X | 0 | 0 | 000 | 0 | 0 |
| **M3** | 0 | 00100 | 0 | X | X | 10 | 1 | 1 | X | 0 | 0 | 000 | 0 | 0 |
| **S1\_M2** | 0 | 00101 | 0 | 0 | 0 | 11 | 1 | 1 | 0 | 0 | 0 | 000 | 0 | 0 |
| **S3\_M4** | 0 | 00110 | 0 | 0 | 1 | 11 | 0 | 1 | 0 | 0 | 0 | 000 | 0 | 0 |
| **S2\_M5** | 0 | 00111 | 1 | 1 | X | 00 | X | 0 | 1 | 0 | 0 | 000 | 0 | 0 |
| **S4\_M6** | 1 | 01000 | 0 | 1 | X | 01 | X | 0 | 0 | 0 | 1 | 000 | 1 | 0 |
| **S5** | 0 | 01001 | 1 | 1 | X | X | X | X | 1 | 0 | 1 | 001 | 0 | 0 |
| **LOAD\_ST\_S** | 0 | 01110 | 1 | 1 | X | X | X | X | 1 | 0 | 1 | 001 | 0 | 0 |
| **S6** | 0 | 01011 | 1 | 1 | X | X | X | X | 1 | 0 | 1 | 010 | 0 | 0 |
| **ROUND3** | 0 | 01100 | 0 | X | X | X | X | X | X | 0 | 1 | 011 | 0 | 0 |
| **ROUND4** | 0 | 01101 | 0 | X | X | X | X | X | X | 0 | 0 | 100 | 0 | 0 |
| **LAST SAVE** | 0 | 10100 | 0 | X | X | X | X | X | X | 1 | 0 | 100 | 0 | 0 |
| **DONE** | 0 | 00000 | 0 | X | X | X | X | X | X | 0 | 0 | 000 | 0 | 1 |
| **M1\_S** | 0 | 01111 | 1 | 1 | X | 10 | 0 | 1 | 1 | 0 | 1 | 010 | 0 | 0 |
| **M3\_S** | 0 | 10000 | 0 | X | X | 10 | 1 | 1 | X | 0 | 1 | 011 | 0 | 0 |
| **SI\_M2\_S** | 0 | 10001 | 0 | 0 | 0 | 11 | 1 | 1 | 0 | 0 | 0 | 100 | 0 | 0 |
| **S3\_M4\_S** | 0 | 10010 | 0 | 0 | 1 | 11 | 0 | 1 | 0 | 0 | 0 | 000 | 0 | 0 |
| **S2\_M5\_S** | 0 | 10010 | 0 | 1 | X | 00 | X | 0 | 1 | 0 | 0 | 000 | 0 | 1 |
| **S4\_M6\_S** | 1 | 01000 | 0 | 1 | X | 01 | X | 0 | 0 | 0 | 1 | 000 | 1 | 0 |

Figura : μROM singola Butterfly

Schema organizzativo μROM del componente CU\_TOP:

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **PRESENT**  **STATE** | **CC** | **NEXT STATE** | **LOAD** | **EN\_FF** | **FF\_VALUE** | **DONE** | **READY** | **RESET** |
| **RESET** | 00 | 0000 | 0 | 0 | 0 | 0 | 0 | 0 |
| **IDLE** | 01 | 0001 | 0 | 0 | 0 | 0 | 0 | 0 |
| **LOAD** | 00 | 0111 | 1 | 1 | 0 | 0 | 0 | 0 |
| **WORK** | 11 | 0100 | 0 | 0 | 1 | 0 | 0 | 0 |
| **FREE\_M** | 00 | 0111 | 1 | 0 | 1 | 0 | 0 | 0 |
| **READY** | 01 | 1010 | 0 | 0 | 0 | 0 | 1 | 0 |
| **FREE\_M\_READY** | 00 | 0111 | 1 | 0 | 1 | 0 | 1 | 0 |
| **DONE** | 00 | 0011 | 0 | 0 | 0 | 1 | 0 | 1 |
| **PREWORK** | 00 | 0111 | 0 | 0 | 0 | 0 | 0 | 0 |

Figura 16: μROM CU Top Level

# Criteri da rispettare

Per far sì che la macchina funzioni correttamente, bisogna rispettare dei tempi specifici per inviare i nuovi dati.

Affinché ogni segnale venga letto correttamente senza essere sovrascritto da un nuovo segnale in arrivo e affinché si possa garantire il corretto funzionamento in sequenza, è necessario che intercorrano 7 colpi di clock tra un dato e il successivo.

Se più campioni vengono inviati durante questo intervallo, infatti, l’ultimo sovrascriverà i precedenti. Mentre se un campione viene inviato oltre i 7 colpi di clock, la macchina non lo campionerà in tempo e non avverrà la lettura in sequenza. Questo perché i dati devono cambiare prima che le BF di primo livello abbiano il moltiplicatore libero (segnale FREE\_M) in modo che la logica di controllo riconosca la richiesta di sequenza; dopo di che i dati devono rimanere validi e costanti per essere campionati quando la CU\_TOP giunge nello stato di FREE\_M.

# Scelte realizzative

Nella realizzazione del progetto si è scelto di implementare all’ingresso della FFT\_16 un blocco atto a ridurre la dinamica di ingresso tra -0.5 e 0.5 esclusi in modo da permettere all’ utente di inviare campioni compresi tra -1 e 1. In uscita, invece, il dato deve esser moltiplicato per un fattore 32 in quanto anche al quarto livello in uscita dalle BF è prevista un’operazione di adattamento della dinamica per la BF successiva.

# Segnali di controllo

Per quanto riguarda la gestione dei controlli per il segnale di START si è usata una porta OR su tutti i bit di tutti i campioni di ingresso, secondo il presupposto che l’assenza di campioni da processare sia data da tutti e quanti i campioni a 0. Inoltre, tramite una porta AND tra il segnale di start generato dalla logica combinatoria OR e un controllo SENSE, si è scelto di rendere la macchina sensibile al segnale di start solamente quando si trova nello stato di IDLE. In questo modo il segnale di start, che viene inviato alla PLA della CU\_TOP una volta attivata la macchina, rimane fisso a zero. Ciò ha permesso una notevole semplificazione nella gestione della logica combinatoria per i salti a 4 vie.

Il segnale di funzionamento in sequenza è realizzato con una logica XOR tra i campioni posti in ingresso e quelli precedentemente caricati nelle BF di primo livello (questo è reso possibile usando delle porte aggiuntive delle BF denominate CTRL sui dati AR, AI, BR, BI). Ciò permette di capire se agli ingressi sono stati inviati dei nuovi dati. Inoltre, tramite una AND con il segnale di start si verifica anche che i campioni non siano tutti a zero, condizione scelta per indicare l’assenza di campioni e l’evoluzione della macchina nello stato di IDLE al completamento dell’operazione in corso.

Per la gestione del segnale di sequenza è stato inserito un componente basato su dei FF, uno per ognuna delle quattro CU\_BF, dotati di segnali di set e reset. In questo modo è possibile settarli tutti e quattro ad 1 nel caso in cui ci sia l’esecuzione in sequenza, ma di rimuovere il segnale singolarmente per ogni CU\_BF in modo che le BF di livello successivo possano portare a termine le operazioni rimaste in coda nei livelli di BF precedenti. In dettaglio il segnale di sequenza per il livello successivo viene azzerato solamente quando il livello di BF precedente ha terminato tutte le sue esecuzioni. Ciò è reso possibile mettendo in relazione i FF di questo componente con quelli descritti nel paragrafo successivo che generano il segnale di PROGRESS.

Per il controllo dei processi in atto nelle BF si è studiato un componente basato nuovamente su dei FF con set e reset, uno per ogni CU\_BF, che vengono posti tutti ad 1 quando si inizia l’elaborazione di nuovi campioni, mentre quando un livello di BF termina l’esecuzione imposta a 0 il suo relativo FF. Le uscite dei quattro FF poste in OR permettono di generare il segnale di PROGRESS atto a discriminare l’evoluzione della CU\_TOP nello stato di DONE piuttosto che nello stato di PRE WORK, ovvero il caso in cui ci siano delle elaborazioni in coda qualora la macchina funzioni in sequenza.

In uscita è previsto un segnale di DONE per indicare che la macchina ha terminato completamente la sua esecuzione e un segnale di READY che avvisi che i dati in uscita sono validi

Es. Un’elaborazione in sequenza con 7 campioni in ingresso avrà in uscita 7 impulsi ad 1 del segnale di ready e un impulso ad 1 del segnale di DONE a termine operazione.

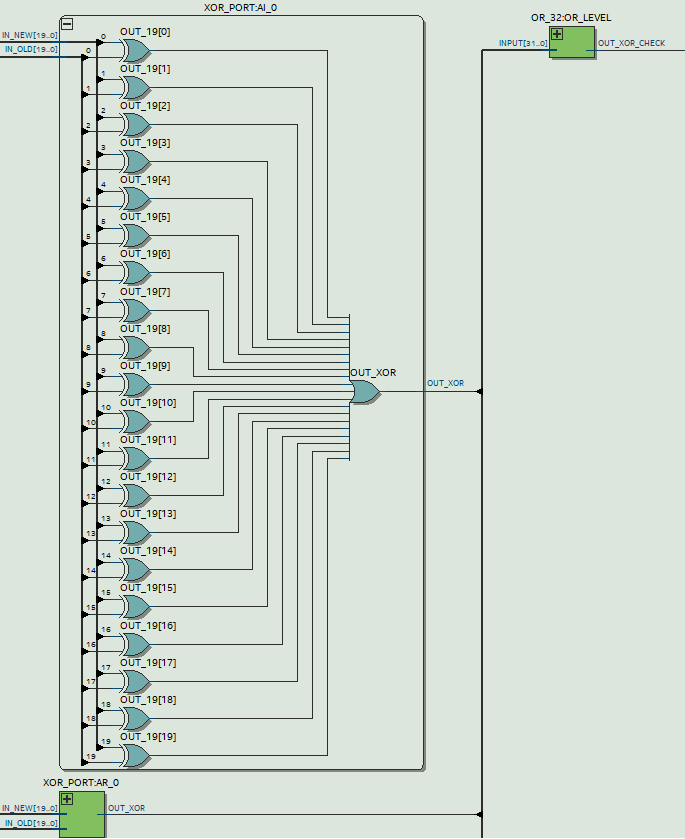


Figura : Sezione logica XOR per la sequenza

Codifica CU TOP

|  |  |
| --- | --- |
| **CODIFICA** | **STATO** |
| RESET | 0011 |
| IDLE | 0000 |
| LOAD | 0001 |
| WORK | 0111 |
| FREE\_M | 0100 |
| READY | 0101 |
| FREE\_M\_READY | 0110 |
| DONE | 1010 |
| PREWORK | 1011 |

Codifica CU BF

|  |  |
| --- | --- |
| **CODIFICA** | **STATO** |
| IDLE | 00000 |
| LAOD\_ST | 00001 |
| M1 | 00010 |
| M3 | 00011 |
| S1\_M2 | 00100 |
| S3\_M4 | 00101 |
| S2\_M5 | 00110 |
| S4\_M6 | 00111 |
| S5 | 01000 |
| LOAD\_ST\_S | 01001 |
| S6 | 01010 |
| ROUND3 | 01011 |
| ROUND4 | 01100 |
| LAST SAVE | 01101 |
| DONE | 10100 |
| M1\_S | 01110 |
| M3\_S | 01111 |
| S1\_M2\_S | 10000 |
| S3\_M4\_S | 10001 |
| S2\_M5\_S | 10010 |
| S4\_M6\_S | 10011 |

# Rounding

Di seguito vengono riportati i risultati del Rounding realizzato secondo la tecnica to nearest even.





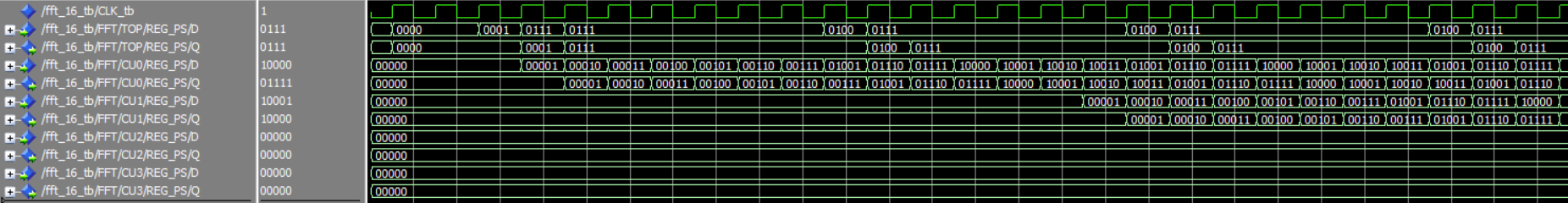




# Timing

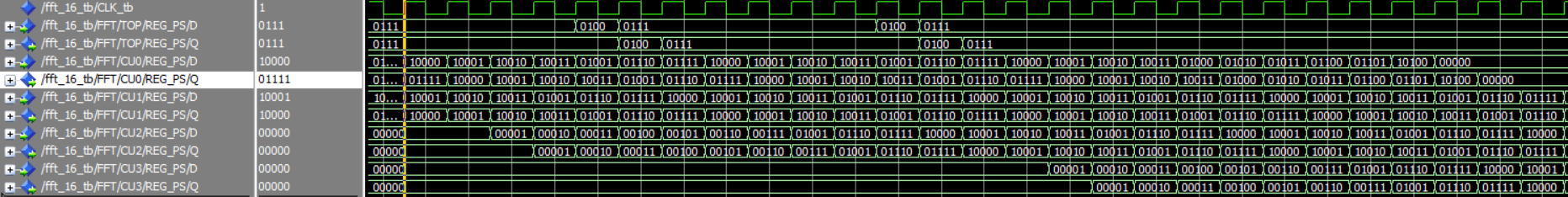
## Timing esecuzione continua

Nelle figure sottostanti sono riportati in sequenza tutti gli stati che vengono svolti dalle varie control unit nel caso di lettura in sequenza.



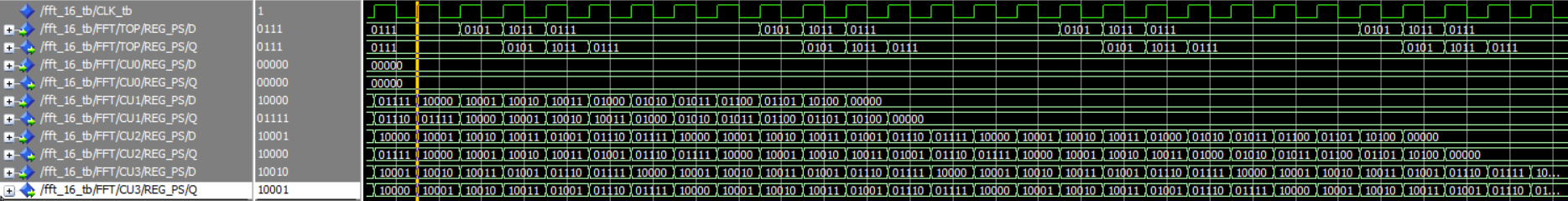
Figura

In particolare, la Figura 18 mostra il passaggio dallo stato di IDLE a quello di LOAD ed il caricamento dei dati nel primo livello di BF. Essendoci un altro dato da caricare in sequenza, la BF va nello stato LOAD\_ST\_S e quando viene terminato il primo campione inizia a fare le sue operazioni anche la BF del secondo livello.



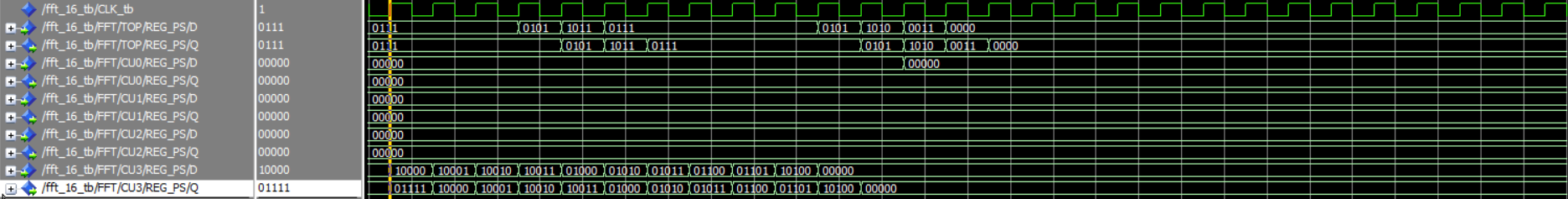
Figura

In Figura 19 si può notare che le BF del terzo e del quarto livello si attivano.



Figura

In Figura 20, invece, è possibile notare che le BF del primo e del secondo livello, terminati i campioni da processare, vanno nello stato di IDLE mentre quelle dei livelli successivi continuano a processare i dati rimasti.



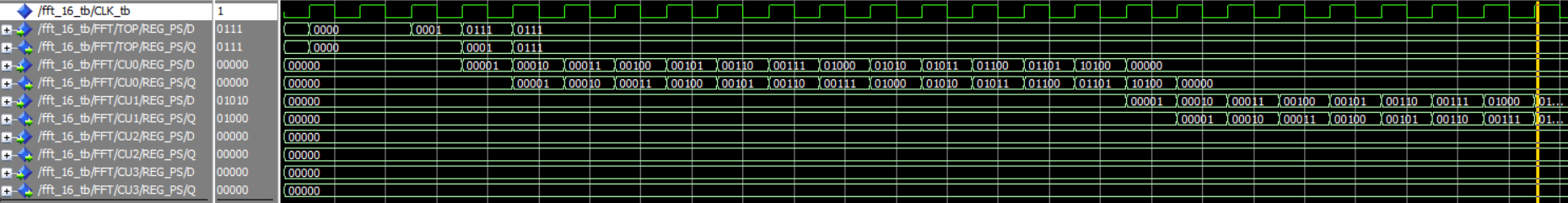
Figura

Infine, in Figura 21 si osserva che, terminata anche l’elaborazione dell’ultimo livello di BF la CU\_TOP passa nello stato di DONE, reimposta i parametri iniziali nello stato di RESET e termina in IDLE pronta a leggere un nuovo segnale di START.

## Timing esecuzione singola

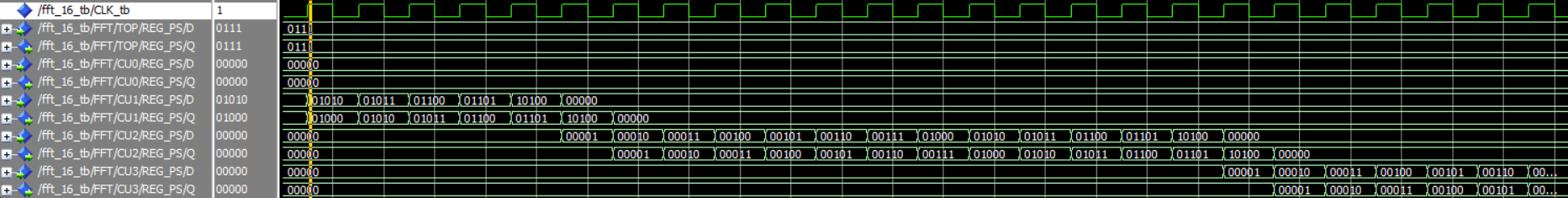
Nelle seguenti figure, sono riportati in sequenza tutti gli stati che vengono svolti dalle varie control unit nel caso di lettura singola.

A differenza del caso continuo, le singole BF eseguono da LOAD a DONE l’elaborazione di un solo campione, senza passare dagli stati di sequenza.

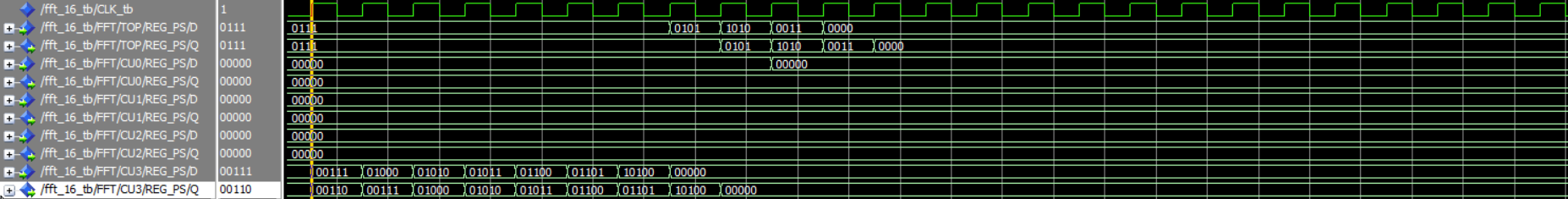


Figura

Inoltre, quando il livello precedente termina l’elaborazione quello successivo inizia l’elaborazione dei dati.



Figura



Figura

In Figura 24, terminata l’operazione sul quarto ed ultimo livello di BF, la CU\_TOP transisce da WORK a READY e poi in DONE.

## Timing relativo agli ingressi

Testo alternativo generato dal computer:
00000000000000000000
10000000000000000000
01111111111111111111
10000000000000000000
01000000000000000000
00000000000000000000
00000000000000000000
10000000000000000000
00000000000000000000
10000000000000000000
01000000000000000000
00000000000000000000
00000000000000000000
10000000000000000000
01111111111111111111
00000000000000000000
01111111111111111111
01000000000000000000
00000000000000000000
00000000000000000000
10000000000000000000
00000000000000000000
01111111111111111111
01000000000000000000
00000000000000000000
00000000000000000000
10000000000000000000
00000000000000000000
10000000000000000000
01000000000000000000
00000000000000000000
00000000000000000000
10000000000000000000
00000000000000000000
10000000000000000000
01000000000000000000
00000000000000000000
00000000000000000000
10000000000000000000
01111111111111111111
00000000000000000000
01111111111111111111
01000000000000000000
00000000000000000000
00000000000000000000
10000000000000000000
00000000000000000000
01111111111111111111
01000000000000000000
00000000000000000000
00000000000000000000
00000000000000000000
00000000000000000000
00000000000000000000
00000000000000000000
00000000000000000000
00000000000000000000
00000000000000000000
00000000000000000000
10000000000000000000
00000000000000000000
10000000000000000000
01000000000000000000
01100000000000000000
00000000000000000000
10000000000000000000
00000000000000000000
10000000000000000000
11000000000000000000
00000000000000000000
00000000000000000000
10000000000000000000
01111111111111111111
00000000000000000000
01111111111111111111
11000000000000000000
00000000000000000000
00000000000000000000
10000000000000000000
00000000000000000000
01111111111111111111
11000000000000000000
00000000000000000000
00000000000000000000
10000000000000000000
00000000000000000000
10000000000000000000
11000000000000000000
00000000000000000000
00000000000000000000
10000000000000000000
00000000000000000000
10000000000000000000
11000000000000000000
00000000000000000000
00000000000000000000
10000000000000000000
01111111111111111111
00000000000000000000
01111111111111111111
11000000000000000000
00000000000000000000
00000000000000000000
10000000000000000000
00000000000000000000
01111111111111111111
11000000000000000000
00000000000000000000
00000000000000000000
00000000000000000000
00000000000000000000
00000000000000000000
00000000000000000000
00000000000000000000
00000000000000000000
00000000000000000000
0 ps
200000 ps
400000 ps
600000 ps
800000 ps
/fft_16_tb/CLK_tb
/fft_16_tb/RESET_tb
/fft_16_tb/DONE_tb
/fft_16_tb/READY_tb
/fft_16_tb/C0AR_tb
00000000000000000000
10000000000000000000
01111111111111111111
10000000000000000000
01000000000000000000
00000000000000000000
/fft_16_tb/C1AR_tb
00000000000000000000
10000000000000000000
00000000000000000000
10000000000000000000
01000000000000000000
00000000000000000000
/fft_16_tb/C2AR_tb
00000000000000000000
10000000000000000000
01111111111111111111
00000000000000000000
01111111111111111111
01000000000000000000
00000000000000000000
/fft_16_tb/C3AR_tb
00000000000000000000
10000000000000000000
00000000000000000000
01111111111111111111
01000000000000000000
00000000000000000000
/fft_16_tb/C4AR_tb
00000000000000000000
10000000000000000000
00000000000000000000
10000000000000000000
01000000000000000000
00000000000000000000
/fft_16_tb/C5AR_tb
00000000000000000000
10000000000000000000
00000000000000000000
10000000000000000000
01000000000000000000
00000000000000000000
/fft_16_tb/C6AR_tb
00000000000000000000
10000000000000000000
01111111111111111111
00000000000000000000
01111111111111111111
01000000000000000000
00000000000000000000
/fft_16_tb/C7AR_tb
00000000000000000000
10000000000000000000
00000000000000000000
01111111111111111111
01000000000000000000
00000000000000000000
/fft_16_tb/C0AI_tb
00000000000000000000
/fft_16_tb/C1AI_tb
00000000000000000000
/fft_16_tb/C2AI_tb
00000000000000000000
/fft_16_tb/C3AI_tb
00000000000000000000
/fft_16_tb/C4AI_tb
00000000000000000000
/fft_16_tb/C5AI_tb
00000000000000000000
/fft_16_tb/C6AI_tb
00000000000000000000
/fft_16_tb/C7AI_tb
00000000000000000000
/fft_16_tb/C0BR_tb
00000000000000000000
10000000000000000000
00000000000000000000
10000000000000000000
01000000000000000000
01100000000000000000
/fft_16_tb/C1BR_tb
00000000000000000000
10000000000000000000
00000000000000000000
10000000000000000000
11000000000000000000
00000000000000000000
/fft_16_tb/C2BR_tb
00000000000000000000
10000000000000000000
01111111111111111111
00000000000000000000
01111111111111111111
11000000000000000000
00000000000000000000
/fft_16_tb/C3BR_tb
00000000000000000000
10000000000000000000
00000000000000000000
01111111111111111111
11000000000000000000
00000000000000000000
/fft_16_tb/C4BR_tb
00000000000000000000
10000000000000000000
00000000000000000000
10000000000000000000
11000000000000000000
00000000000000000000
/fft_16_tb/C5BR_tb
00000000000000000000
10000000000000000000
00000000000000000000
10000000000000000000
11000000000000000000
00000000000000000000
/fft_16_tb/C6BR_tb
00000000000000000000
10000000000000000000
01111111111111111111
00000000000000000000
01111111111111111111
11000000000000000000
00000000000000000000
/fft_16_tb/C7BR_tb
00000000000000000000
10000000000000000000
00000000000000000000
01111111111111111111
11000000000000000000
00000000000000000000
/fft_16_tb/C0BI_tb
00000000000000000000
/fft_16_tb/C1BI_tb
00000000000000000000
/fft_16_tb/C2BI_tb
00000000000000000000
/fft_16_tb/C3BI_tb
00000000000000000000
/fft_16_tb/C4BI_tb
00000000000000000000
/fft_16_tb/C5BI_tb
00000000000000000000
/fft_16_tb/C6BI_tb
00000000000000000000
/fft_16_tb/C7BI_tb
00000000000000000000
Entity:fft_16_tb  Architecture:behav  Date: Sat Jan 23 18:54:25 CET 2021   Row: 1 Page: 1


**6° campione**

**5° campione**

**° campione**

**4° campione**

**3° campione**

**° campione**

**2° campione**

**° campione**

**1° campione**

**Ingressi nulli**

## Timing relativo alle uscite

Testo alternativo generato dal computer:
UUUUUUUUUUUUUUUUUUUU
11000000000000000000
11111111111111111111
00000011111111111111
11111111111111111111
00000100000000000000
00000011000000000000
UUUUUUUUUUUUUUUUUUUU
00000000000000000000
11100000000000000000
00000100000000000000
11100000000000000000
00000100000000000000
00000011000000000000
UUUUUUUUUUUUUUUUUUUU
00000000000000000000
00000100000000000000
11111111111111111111
00000100000000000000
00000011000000000000
UUUUUUUUUUUUUUUUUUUU
00000000000000000000
00000100000000000000
00000000000000000000
00000100000000000000
00000011000000000000
UUUUUUUUUUUUUUUUUUUU
00000000000000000000
11111111111111111111
00000100000000000000
11111111111111111111
00000000000000000000
11111101000000000000
UUUUUUUUUUUUUUUUUUUU
00000000000000000000
00000100000000000000
00000000000000000000
11111101000000000000
UUUUUUUUUUUUUUUUUUUU
00000000000000000000
11111111111111111111
00000100000000000000
11111111111111111111
00000000000000000000
11111101000000000000
UUUUUUUUUUUUUUUUUUUU
00000000000000000000
00000100000000000000
00000000000000000000
11111101000000000000
UUUUUUUUUUUUUUUUUUUU
00000000000000000000
UUUUUUUUUUUUUUUUUUUU
00000000000000000000
00100000000000000000
00000000000000000000
UUUUUUUUUUUUUUUUUUUU
00000000000000000000
11111111111111111111
00000000000000000000
11111111111111111111
00000000000000000000
UUUUUUUUUUUUUUUUUUUU
00000000000000000000
UUUUUUUUUUUUUUUUUUUU
00000000000000000000
11111111111111111111
00000000000000000000
11111111111111111111
11101011111001000000
00000000000000000000
UUUUUUUUUUUUUUUUUUUU
00000000000000000000
11111101010100111100
00000000000000000000
UUUUUUUUUUUUUUUUUUUU
00000000000000000000
11111111111111111111
00000000000000000000
11111010000000110111
00000000000000000000
UUUUUUUUUUUUUUUUUUUU
00000000000000000000
11111111001101000101
00000000000000000000
UUUUUUUUUUUUUUUUUUUU
00000000000000000000
00000100000000000000
00000000000000000001
00000100000000000000
00000011000000000000
UUUUUUUUUUUUUUUUUUUU
00000000000000000000
11100000000000000000
00000100000000000000
11100000000000000000
00000100000000000000
00000011000000000000
UUUUUUUUUUUUUUUUUUUU
00000000000000000000
00000100000000000000
00000000000000000000
00000100000000000000
00000011000000000000
UUUUUUUUUUUUUUUUUUUU
00000000000000000000
00000100000000000000
00000000000000000000
00000100000000000000
00000011000000000000
UUUUUUUUUUUUUUUUUUUU
00000000000000000000
00000100000000000000
00000000000000000001
00000000000000000000
11111101000000000000
UUUUUUUUUUUUUUUUUUUU
00000000000000000000
00000100000000000000
00000000000000000000
11111101000000000000
UUUUUUUUUUUUUUUUUUUU
00000000000000000000
00000100000000000000
00000000000000000001
00000000000000000000
11111101000000000000
UUUUUUUUUUUUUUUUUUUU
00000000000000000000
00000100000000000000
00000000000000000000
11111101000000000000
UUUUUUUUUUUUUUUUUUUU
00000000000000000000
UUUUUUUUUUUUUUUUUUUU
00000000000000000000
11100000000000000000
00000000000000000000
UUUUUUUUUUUUUUUUUUUU
00000000000000000000
00000000000000000001
00000000000000000000
UUUUUUUUUUUUUUUUUUUU
00000000000000000000
UUUUUUUUUUUUUUUUUUUU
00000000000000000000
00000000000000000001
00000000110010111100
00000000000000000000
UUUUUUUUUUUUUUUUUUUU
00000000000000000000
00000101111111001001
00000000000000000000
UUUUUUUUUUUUUUUUUUUU
00000000000000000000
00000010101011000100
00000000000000000000
UUUUUUUUUUUUUUUUUUUU
00000000000000000000
00010100000111000001
00000000000000000000
1000000 ps
1200000 ps
1400000 ps
1600000 ps
1800000 ps
/fft_16_tb/CLK_tb
/fft_16_tb/RESET_tb
/fft_16_tb/DONE_tb
/fft_16_tb/READY_tb
/fft_16_tb/C0AR_OUT_tb
UUUUUUUUUUUUUUUUUUUU
11000000000000000000
11111111111111111111
00000011111111111111
11111111111111111111
00000100000000000000
00000011000000000000
/fft_16_tb/C1AR_OUT_tb
UUUUUUUUUUUUUUUUUUUU
00000000000000000000
11100000000000000000
00000100000000000000
11100000000000000000
00000100000000000000
00000011000000000000
/fft_16_tb/C2AR_OUT_tb
UUUUUUUUUUUUUUUUUUUU
00000000000000000000
00000100000000000000
11111111111111111111
00000100000000000000
00000011000000000000
/fft_16_tb/C3AR_OUT_tb
UUUUUUUUUUUUUUUUUUUU
00000000000000000000
00000100000000000000
00000000000000000000
00000100000000000000
00000011000000000000
/fft_16_tb/C4AR_OUT_tb
UUUUUUUUUUUUUUUUUUUU
00000000000000000000
11111111111111111111
00000100000000000000
11111111111111111111
00000000000000000000
11111101000000000000
/fft_16_tb/C5AR_OUT_tb
UUUUUUUUUUUUUUUUUUUU
00000000000000000000
00000100000000000000
00000000000000000000
11111101000000000000
/fft_16_tb/C6AR_OUT_tb
UUUUUUUUUUUUUUUUUUUU
00000000000000000000
11111111111111111111
00000100000000000000
11111111111111111111
00000000000000000000
11111101000000000000
/fft_16_tb/C7AR_OUT_tb
UUUUUUUUUUUUUUUUUUUU
00000000000000000000
00000100000000000000
00000000000000000000
11111101000000000000
/fft_16_tb/C0AI_OUT_tb
UUUUUUUUUUUUUUUUUUUU
00000000000000000000
/fft_16_tb/C1AI_OUT_tb
UUUUUUUUUUUUUUUUUUUU
00000000000000000000
00100000000000000000
00000000000000000000
/fft_16_tb/C2AI_OUT_tb
UUUUUUUUUUUUUUUUUUUU
00000000000000000000
11111111111111111111
00000000000000000000
11111111111111111111
00000000000000000000
/fft_16_tb/C3AI_OUT_tb
UUUUUUUUUUUUUUUUUUUU
00000000000000000000
/fft_16_tb/C4AI_OUT_tb
UUUUUUUUUUUUUUUUUUUU
00000000000000000000
11111111111111111111
00000000000000000000
11111111111111111111
11101011111001000000
00000000000000000000
/fft_16_tb/C5AI_OUT_tb
UUUUUUUUUUUUUUUUUUUU
00000000000000000000
11111101010100111100
00000000000000000000
/fft_16_tb/C6AI_OUT_tb
UUUUUUUUUUUUUUUUUUUU
00000000000000000000
11111111111111111111
00000000000000000000
11111010000000110111
00000000000000000000
/fft_16_tb/C7AI_OUT_tb
UUUUUUUUUUUUUUUUUUUU
00000000000000000000
11111111001101000101
00000000000000000000
/fft_16_tb/C0BR_OUT_tb
UUUUUUUUUUUUUUUUUUUU
00000000000000000000
00000100000000000000
00000000000000000001
00000100000000000000
00000011000000000000
/fft_16_tb/C1BR_OUT_tb
UUUUUUUUUUUUUUUUUUUU
00000000000000000000
11100000000000000000
00000100000000000000
11100000000000000000
00000100000000000000
00000011000000000000
/fft_16_tb/C2BR_OUT_tb
UUUUUUUUUUUUUUUUUUUU
00000000000000000000
00000100000000000000
00000000000000000000
00000100000000000000
00000011000000000000
/fft_16_tb/C3BR_OUT_tb
UUUUUUUUUUUUUUUUUUUU
00000000000000000000
00000100000000000000
00000000000000000000
00000100000000000000
00000011000000000000
/fft_16_tb/C4BR_OUT_tb
UUUUUUUUUUUUUUUUUUUU
00000000000000000000
00000100000000000000
00000000000000000001
00000000000000000000
11111101000000000000
/fft_16_tb/C5BR_OUT_tb
UUUUUUUUUUUUUUUUUUUU
00000000000000000000
00000100000000000000
00000000000000000000
11111101000000000000
/fft_16_tb/C6BR_OUT_tb
UUUUUUUUUUUUUUUUUUUU
00000000000000000000
00000100000000000000
00000000000000000001
00000000000000000000
11111101000000000000
/fft_16_tb/C7BR_OUT_tb
UUUUUUUUUUUUUUUUUUUU
00000000000000000000
00000100000000000000
00000000000000000000
11111101000000000000
/fft_16_tb/C0BI_OUT_tb
UUUUUUUUUUUUUUUUUUUU
00000000000000000000
/fft_16_tb/C1BI_OUT_tb
UUUUUUUUUUUUUUUUUUUU
00000000000000000000
11100000000000000000
00000000000000000000
/fft_16_tb/C2BI_OUT_tb
UUUUUUUUUUUUUUUUUUUU
00000000000000000000
00000000000000000001
00000000000000000000
/fft_16_tb/C3BI_OUT_tb
UUUUUUUUUUUUUUUUUUUU
00000000000000000000
/fft_16_tb/C4BI_OUT_tb
UUUUUUUUUUUUUUUUUUUU
00000000000000000000
00000000000000000001
00000000110010111100
00000000000000000000
/fft_16_tb/C5BI_OUT_tb
UUUUUUUUUUUUUUUUUUUU
00000000000000000000
00000101111111001001
00000000000000000000
/fft_16_tb/C6BI_OUT_tb
UUUUUUUUUUUUUUUUUUUU
00000000000000000000
00000010101011000100
00000000000000000000
/fft_16_tb/C7BI_OUT_tb
UUUUUUUUUUUUUUUUUUUU
00000000000000000000
00010100000111000001
00000000000000000000
Entity:fft_16_tb  Architecture:behav  Date: Sat Jan 23 18:56:46 CET 2021   Row: 1 Page: 1


**6°uscita**

**5°uscita**

**4°uscita**

**3°uscita**

**1° uscita**

**2°uscita**

# Script MATLAB per testare la FFT per ogni singolo campione

clc**;**

clear all**;**

campione **=** "inserire numero del campione da voler testare"**;**

**if** campione **==** 1

AR **=** **[-**1**,-**1**,-**1**,-**1**,-**1**,-**1**,-**1**,-**1**];**

**elseif** campione **==** 2

AR **=** **[-**1**,**0**,**1**,**0**,-**1**,**0**,**1**,**0**];**

**elseif** campione **==** 3

AR **=** **[**1**,**0**,**0**,**0**,**0**,**0**,**0**,**0**];**

**elseif** campione **==** 4

AR **=** **[-**1**,-**1**,**1**,**1**,-**1**,-**1**,**1**,**1**];**

**elseif** campione **==** 5

AR **=** **[**0.5**,**0.5**,**0.5**,**0.5**,**0.5**,**0.5**,**0.5**,**0.5**];**

**elseif** campione **==** 6

AR **=** **[**0**,**0**,**0**,**0**,**0**,**0**,**0**,**0**];**

**end**

AI **=** **[**0**,**0**,**0**,**0**,**0**,**0**,**0**,**0**];**

**if** campione **==** 1

BR **=** **[-**1**,-**1**,-**1**,-**1**,-**1**,-**1**,-**1**,-**1**];**

**elseif** campione **==** 2

BR **=** **[-**1**,**0**,**1**,**0**,-**1**,**0**,**1**,**0**];**

**elseif** campione **==** 3

BR **=** **[**0**,**0**,**0**,**0**,**0**,**0**,**0**,**0**];**

**elseif** campione **==** 4

BR **=** **[-**1**,-**1**,**1**,**1**,-**1**,-**1**,**1**,**1**];**

**elseif** campione **==** 5

BR **=** **[**0.5**,-**0.5**,-**0.5**,-**0.5**,-**0.5**,-**0.5**,-**0.5**,-**0.5**];**

**elseif** campione **==** 6

BR **=** **[**0.75**,**0**,**0**,**0**,**0**,**0**,**0**,**0**];**

**end**

BI **=** **[**0**,**0**,**0**,**0**,**0**,**0**,**0**,**0**];**

A1R**=** **[**0**,**0**,**0**,**0**,**0**,**0**,**0**,**0**];** A1I**=** **[**0**,**0**,**0**,**0**,**0**,**0**,**0**,**0**];**

B1R**=** **[**0**,**0**,**0**,**0**,**0**,**0**,**0**,**0**];** B1I**=** **[**0**,**0**,**0**,**0**,**0**,**0**,**0**,**0**];**

A1R\_temp**=** **[**0**,**0**,**0**,**0**,**0**,**0**,**0**,**0**];** A1I\_temp**=** **[**0**,**0**,**0**,**0**,**0**,**0**,**0**,**0**];**

B1R\_temp**=** **[**0**,**0**,**0**,**0**,**0**,**0**,**0**,**0**];** B1I\_temp**=** **[**0**,**0**,**0**,**0**,**0**,**0**,**0**,**0**];**

WR **=** **[**1**,**0.92387953251129**,**0.70710678118655**,**0.38268343236509**,**0**,-**0.38268343236509**,-**0.70710678118655**,-**0.92387953251129**];**

WI **=** **[**0**,-**0.38268343236509**,-**0.70710678118655**,-**0.92387953251129**,-**1**,-**0.92387953251129**,-**0.70710678118655**,-**0.38268343236509**];**

%% bf0

**for** i**=**1**:**8

M1 **=** BR**(**i**)\***WR**(**1**);** M2 **=** BI**(**i**)\***WI**(**1**);** M3 **=** BR**(**i**)\***WI**(**1**);** M4 **=** BI**(**i**)\***WR**(**1**);** M5 **=** 2**\***AR**(**i**);**

M6 **=** 2**\***AI**(**i**);** S1 **=** AR**(**i**)+**M1**;** S2 **=** S1**-**M2**;** S3 **=** AI**(**i**)+**M3**;** S4 **=** S3**+**M4**;**

S5 **=** M5**-**S2**;** S6 **=** M6**-**S4**;**

A1R\_temp**(**i**)** **=** S2**;** A1I\_temp**(**i**)** **=** S4**;** B1R\_temp**(**i**)** **=** S5**;** B1I\_temp**(**i**)** **=** S6**;**

**end**

A1R **=** A1R\_temp**;** A1I **=** A1I\_temp**;** B1R **=** B1R\_temp**;** B1I **=** B1I\_temp**;**

%% bf1

**for** i**=**1**:**8

**if** i **>** 4

M1 **=** B1R**(**i**)\***WR**(**5**);** M2 **=** B1I**(**i**)\***WI**(**5**);** M3 **=** B1R**(**i**)\***WI**(**5**);** M4 **=** B1I**(**i**)\***WR**(**5**);** M5 **=** 2**\***B1R**(**i**-**4**);**

M6 **=** 2**\***B1I**(**i**-**4**);** S1 **=** B1R**(**i**-**4**)+**M1**;** S2 **=** S1**-**M2**;** S3 **=** B1I**(**i**-**4**)+**M3**;** S4 **=** S3**+**M4**;**

S5 **=** M5**-**S2**;** S6 **=** M6**-**S4**;**

**else**

M1 **=** A1R**(**i**+**4**)\***WR**(**1**);**M2 **=** A1I**(**i**+**4**)\***WI**(**1**);**M3 **=** A1R**(**i**+**4**)\***WI**(**1**);**M4 **=** A1I**(**i**+**4**)\***WR**(**1**);**M5 **=** 2**\***A1R**(**i**);**

M6 **=** 2**\***A1I**(**i**);** S1 **=** A1R**(**i**)+**M1**;** S2 **=** S1**-**M2**;** S3 **=** A1I**(**i**)+**M3**;** S4 **=** S3**+**M4**;**

S5 **=** M5**-**S2**;** S6 **=** M6**-**S4**;**

**end**

A1R\_temp**(**i**)** **=** S2**;** A1I\_temp**(**i**)** **=** S4**;** B1R\_temp**(**i**)** **=** S5**;** B1I\_temp**(**i**)** **=** S6**;**

**end**

A1R **=** A1R\_temp**;** A1I **=** A1I\_temp**;** B1R **=** B1R\_temp**;** B1I **=** B1I\_temp**;**

%% bf2

**for** i**=**1**:**8

**if** i **>** 6

M1 **=** B1R**(**i**)\***WR**(**7**);** M2 **=** B1I**(**i**)\***WI**(**7**);** M3 **=** B1R**(**i**)\***WI**(**7**);** M4 **=** B1I**(**i**)\***WR**(**7**);** M5 **=** 2**\***B1R**(**i**-**2**);**

M6 **=** 2**\***B1I**(**i**-**2**);** S1 **=** B1R**(**i**-**2**)+**M1**;** S2 **=** S1**-**M2**;** S3 **=** B1I**(**i**-**2**)+**M3**;** S4 **=** S3**+**M4**;**

S5 **=** M5**-**S2**;** S6 **=** M6**-**S4**;**

**elseif** i **>** 4

M1 **=** A1R**(**i**+**2**)\***WR**(**3**);**M2 **=** A1I**(**i**+**2**)\***WI**(**3**);**M3 **=** A1R**(**i**+**2**)\***WI**(**3**);**M4 **=** A1I**(**i**+**2**)\***WR**(**3**);**M5 **=** 2**\***A1R**(**i**);**

M6 **=** 2**\***A1I**(**i**);** S1 **=** A1R**(**i**)+**M1**;** S2 **=** S1**-**M2**;** S3 **=** A1I**(**i**)+**M3**;** S4 **=** S3**+**M4**;**

S5 **=** M5**-**S2**;** S6 **=** M6**-**S4**;**

**elseif** i **>** 2

M1 **=** B1R**(**i**)\***WR**(**5**);** M2 **=** B1I**(**i**)\***WI**(**5**);** M3 **=** B1R**(**i**)\***WI**(**5**);** M4 **=** B1I**(**i**)\***WR**(**5**)** **;**M5 **=** 2**\***B1R**(**i**-**2**);**

M6 **=** 2**\***B1I**(**i**-**2**);** S1 **=** B1R**(**i**-**2**)+**M1**;** S2 **=** S1**-**M2**;** S3 **=** B1I**(**i**-**2**)+**M3**;** S4 **=** S3**+**M4**;**

S5 **=** M5**-**S2**;** S6 **=** M6**-**S4**;**

**else**

M1 **=** A1R**(**i**+**2**)\***WR**(**1**);**M2 **=** A1I**(**i**+**2**)\***WI**(**1**);**M3 **=** A1R**(**i**+**2**)\***WI**(**1**);**M4 **=** A1I**(**i**+**2**)\***WR**(**1**);**M5 **=** 2**\***A1R**(**i**);**

M6 **=** 2**\***A1I**(**i**);** S1 **=** A1R**(**i**)+**M1**;** S2 **=** S1**-**M2**;** S3 **=** A1I**(**i**)+**M3**;** S4 **=** S3**+**M4**;**

S5 **=** M5**-**S2**;** S6 **=** M6**-**S4**;**

**end**

A1R\_temp**(**i**)** **=** S2**;** A1I\_temp**(**i**)** **=** S4**;** B1R\_temp**(**i**)** **=** S5**;** B1I\_temp**(**i**)** **=** S6**;**

**end**

A1R **=** A1R\_temp**;** A1I **=** A1I\_temp**;** B1R **=** B1R\_temp**;** B1I **=** B1I\_temp**;**

%% bf3

**for** i**=**1**:**8

**if** i **>** 7

M1 **=** B1R**(**i**)\***WR**(**8**);** M2 **=** B1I**(**i**)\***WI**(**8**);** M3 **=** B1R**(**i**)\***WI**(**8**);** M4 **=** B1I**(**i**)\***WR**(**8**);** M5 **=** 2**\***B1R**(**i**-**1**);**

M6 **=** 2**\***B1I**(**i**-**1**);** S1 **=** B1R**(**i**-**1**)+**M1**;** S2 **=** S1**-**M2**;** S3 **=** B1I**(**i**-**1**)+**M3**;** S4 **=** S3**+**M4**;**

S5 **=** M5**-**S2**;** S6 **=** M6**-**S4**;**

**elseif** i **>** 6

M1 **=** A1R**(**i**+**1**)\***WR**(**4**);**M2 **=** A1I**(**i**+**1**)\***WI**(**4**);**M3 **=** A1R**(**i**+**1**)\***WI**(**4**);**M4 **=** A1I**(**i**+**1**)\***WR**(**4**);**M5 **=** 2**\***A1R**(**i**);**

M6 **=** 2**\***A1I**(**i**);** S1 **=** A1R**(**i**)+**M1**;** S2 **=** S1**-**M2**;** S3 **=** A1I**(**i**)+**M3**;** S4 **=** S3**+**M4**;**

S5 **=** M5**-**S2**;** S6 **=** M6**-**S4**;**

**elseif** i **>** 5

M1 **=** B1R**(**i**)\***WR**(**6**);** M2 **=** B1I**(**i**)\***WI**(**6**);** M3 **=** B1R**(**i**)\***WI**(**6**);** M4 **=** B1I**(**i**)\***WR**(**6**);** M5 **=** 2**\***B1R**(**i**-**1**);**

M6 **=** 2**\***B1I**(**i**-**1**);** S1 **=** B1R**(**i**-**1**)+**M1**;** S2 **=** S1**-**M2**;** S3 **=** B1I**(**i**-**1**)+**M3**;** S4 **=** S3**+**M4**;**

S5 **=** M5**-**S2**;** S6 **=** M6**-**S4**;**

**elseif** i **>** 4

M1 **=** A1R**(**i**+**1**)\***WR**(**2**);**M2 **=** A1I**(**i**+**1**)\***WI**(**2**);**M3 **=** A1R**(**i**+**1**)\***WI**(**2**);**M4 **=** A1I**(**i**+**1**)\***WR**(**2**);**M5 **=** 2**\***A1R**(**i**);**

M6 **=** 2**\***A1I**(**i**);** S1 **=** A1R**(**i**)+**M1**;** S2 **=** S1**-**M2**;** S3 **=** A1I**(**i**)+**M3**;** S4 **=** S3**+**M4**;**

S5 **=** M5**-**S2**;** S6 **=** M6**-**S4**;**

**elseif** i **>** 3

M1 **=** B1R**(**i**)\***WR**(**7**);** M2 **=** B1I**(**i**)\***WI**(**7**);** M3 **=** B1R**(**i**)\***WI**(**7**);** M4 **=** B1I**(**i**)\***WR**(**7**);** M5 **=** 2**\***B1R**(**i**-**1**);**

M6 **=** 2**\***B1I**(**i**-**1**);** S1 **=** B1R**(**i**-**1**)+**M1**;** S2 **=** S1**-**M2**;** S3 **=** B1I**(**i**-**1**)+**M3**;** S4 **=** S3**+**M4**;**

S5 **=** M5**-**S2**;** S6 **=** M6**-**S4**;**

**elseif** i **>** 2

M1 **=** A1R**(**i**+**1**)\***WR**(**3**);**M2 **=** A1I**(**i**+**1**)\***WI**(**3**);**M3 **=** A1R**(**i**+**1**)\***WI**(**3**);**M4 **=** A1I**(**i**+**1**)\***WR**(**3**);**M5 **=** 2**\***A1R**(**i**);**

M6 **=** 2**\***A1I**(**i**);** S1 **=** A1R**(**i**)+**M1**;** S2 **=** S1**-**M2**;** S3 **=** A1I**(**i**)+**M3**;** S4 **=** S3**+**M4**;**

S5 **=** M5**-**S2**;** S6 **=** M6**-**S4**;**

**elseif** i **>** 1

M1 **=** B1R**(**i**)\***WR**(**5**);** M2 **=** B1I**(**i**)\***WI**(**5**);** M3 **=** B1R**(**i**)\***WI**(**5**);** M4 **=** B1I**(**i**)\***WR**(**5**);** M5 **=** 2**\***B1R**(**i**-**1**);**

M6 **=** 2**\***B1I**(**i**-**1**);** S1 **=** B1R**(**i**-**1**)+**M1**;** S2 **=** S1**-**M2**;** S3 **=** B1I**(**i**)+**M3**;** S4 **=** S3**+**M4**;**

S5 **=** M5**-**S2**;** S6 **=** M6**-**S4**;**

**else**

M1 **=** A1R**(**i**+**1**)\***WR**(**1**);**M2 **=** A1I**(**i**+**1**)\***WI**(**1**);**M3 **=** A1R**(**i**+**1**)\***WI**(**1**);**M4 **=** A1I**(**i**+**1**)\***WR**(**1**);**M5 **=** 2**\***A1R**(**i**);**

M6 **=** 2**\***A1I**(**i**);** S1 **=** A1R**(**i**)+**M1**;** S2 **=** S1**-**M2**;** S3 **=** A1I**(**i**)+**M3**;** S4 **=** S3**+**M4**;**

S5 **=** M5**-**S2**;** S6 **=** M6**-**S4**;**

**end**

A1R\_temp**(**i**)** **=** S2**;** A1I\_temp**(**i**)** **=** S4**;** B1R\_temp**(**i**)** **=** S5**;** B1I\_temp**(**i**)** **=** S6**;**

**end**

A1R **=** A1R\_temp**;** A1I **=** A1I\_temp**;** B1R **=** B1R\_temp**;** B1I **=** B1I\_temp**;**

A1R

A1I

B1R

B1I

**CAMPIONE 1**

**INGRESSI: USCITE:**

AR **=** **[-**1**,-**1**,-**1**,-**1**,-**1**,-**1**,-**1**,-**1**]** A1R **=** **[-**16**,**0**,**0**,**0**,**0**,**0**,**0**,**0**]**

AI **=** **[**0**,**0**,**0**,**0**,**0**,**0**,**0**,**0**]** A1I **=** **[**0**,**0**,**0**,**0**,**0**,**0**,**0**,**0**]**

BR **=** **[-**1**,-**1**,-**1**,-**1**,-**1**,-**1**,-**1**,-**1**]** B1R **=** **[**0**,**0**,**0**,**0**,**0**,**0**,**0**,**0**]**

BR **=** **[**0**,**0**,**0**,**0**,**0**,**0**,**0**,**0**]** B1I **=** **[**0**,**0**,**0**,**0**,**0**,**0**,**0**,**0**]**

**CAMPIONE 2**

**INGRESSI: USCITE:**

AR **=** **[-**1**,**0**,**1**,**0**,-**1**,**0**,**1**,**0**]** A1R **=** **[**0**,-**8**,**0**,**0**,**0**,**0**,**0**,**0**]**

AI **=** **[**0**,**0**,**0**,**0**,**0**,**0**,**0**,**0**]** A1I **=** **[**0**,**0**,**0**,**0**,**0**,**0**,**0**,**0**]**

BR **=** **[-**1**,**0**,**1**,**0**,-**1**,**0**,**1**,**0**]** B1R **=** **[**0**,-**8**,**0**,**0**,**0**,**0**,**0**,**0**]**

BR **=** **[**0**,**0**,**0**,**0**,**0**,**0**,**0**,**0**]** B1I **=** **[**0**,**0**,**0**,**0**,**0**,**0**,**0**,**0**]**

**CAMPIONE 3**

**INGRESSI: USCITE:**

AR **=** **[**1**,**0**,**0**,**0**,**0**,**0**,**0**,**0**]** A1R **=** **[**1**,**1**,**1**,**1**,**1**,**1**,**1**,**1**]**

AI **=** **[**0**,**0**,**0**,**0**,**0**,**0**,**0**,**0**]** A1I **=** **[**0**,**0**,**0**,**0**,**0**,**0**,**0**,**0**]**

BR **=** **[**0**,**0**,**0**,**0**,**0**,**0**,**0**,**0**]** B1R **=** **[**1**,**1**,**1**,**1**,**1**,**1**,**1**,**1**]**

BR **=** **[**0**,**0**,**0**,**0**,**0**,**0**,**0**,**0**]** B1I **=** **[**0**,**0**,**0**,**0**,**0**,**0**,**0**,**0**]**

**CAMPIONE 4**

**INGRESSI: USCITE:**

AR **=** **[-**1**,-**1**,**1**,**1**,-**1**,-**1**,**1**,**1**]** A1R **=** **[**0**,-**8**,**0**,**0**,**0**,**0**,**0**,**0**]**

AI **=** **[**0**,**0**,**0**,**0**,**0**,**0**,**0**,**0**]** A1I **=** **[**0**,**8**,**0**,**0**,**0**,**0**,**0**,**0**]**

BR **=** **[-**1**,-**1**,**1**,**1**,-**1**,-**1**,**1**,**1**]** B1R **=** **[**0**,-**8**,**0**,**0**,**0**,**0**,**0**,**0**]**

BR **=** **[**0**,**0**,**0**,**0**,**0**,**0**,**0**,**0**]** B1I **=** **[**0**,-**8**,**0**,**0**,**0**,**0**,**0**,**0**]**

**CAMPIONE 5**

**INGRESSI:**

AR **=** **[**0.5**,**0.5**,**0.5**,**0.5**,**0.5**,**0.5**,**0.5**,**0.5**]**

AI **=** **[**0**,**0**,**0**,**0**,**0**,**0**,**0**,**0**]**

BR **=** **[**0.5**,-**0.5**,-**0.5**,-**0.5**,-**0.5**,-**0.5**,-**0.5**,-**0.5**]**

BR **=** **[**0**,**0**,**0**,**0**,**0**,**0**,**0**,**0**]**

**USCITE:**

A1R **=** **[**1**,**1**,**1**,**1**,**6.661338147750939e-16**,**5.884182030513330e-15**,-**5.717648576819556e-15**,-**6.661338147750939e-16**]**

A1I **=** **[**0**,**0**,**0**,**0**,-**5.027339492125867**,-**6.681786379192990e-01**,-**1.496605762665499**,-**1.989123673796658e-01**]**

B1R **=** **[**1**,**1**,**1**,**1**,-**6.661338147750939e-16**,-**5.884182030513330e-15**,**5.717648576819556e-15**,**6.661338147750939e-16**]**

B1I **=** **[**0**,**0**,**0**,**0**,**1.989123673796662e-01**,**1.496605762665499**,**6.681786379192991e-01**,**5.027339492125867**]**

**CAMPIONE 6**

**INGRESSI:**

AR **=** **[**0**,**0**,**0**,**0**,**0**,**0**,**0**,**0**]**

AI **=** **[**0**,**0**,**0**,**0**,**0**,**0**,**0**,**0**]**

BR **=** **[**0.75**,**0**,**0**,**0**,**0**,**0**,**0**,**0**]**

BR **=** **[**0**,**0**,**0**,**0**,**0**,**0**,**0**,**0**]**

**USCITE:**

A1R **=** **[**0.75**,**0.75**,**0.75**,**0.75**,-**0.75**,-**0.75**,-**0.75**,-**0.75**]**

A1I **=** **[**0**,**0**,**0**,**0**,**0**,**0**,**0**,**0**]**

B1R **=** **[**0.75**,**0.75**,**0.75**,**0.75**,-**0.75**,-**0.75**,-**0.75**,-**0.75**]**

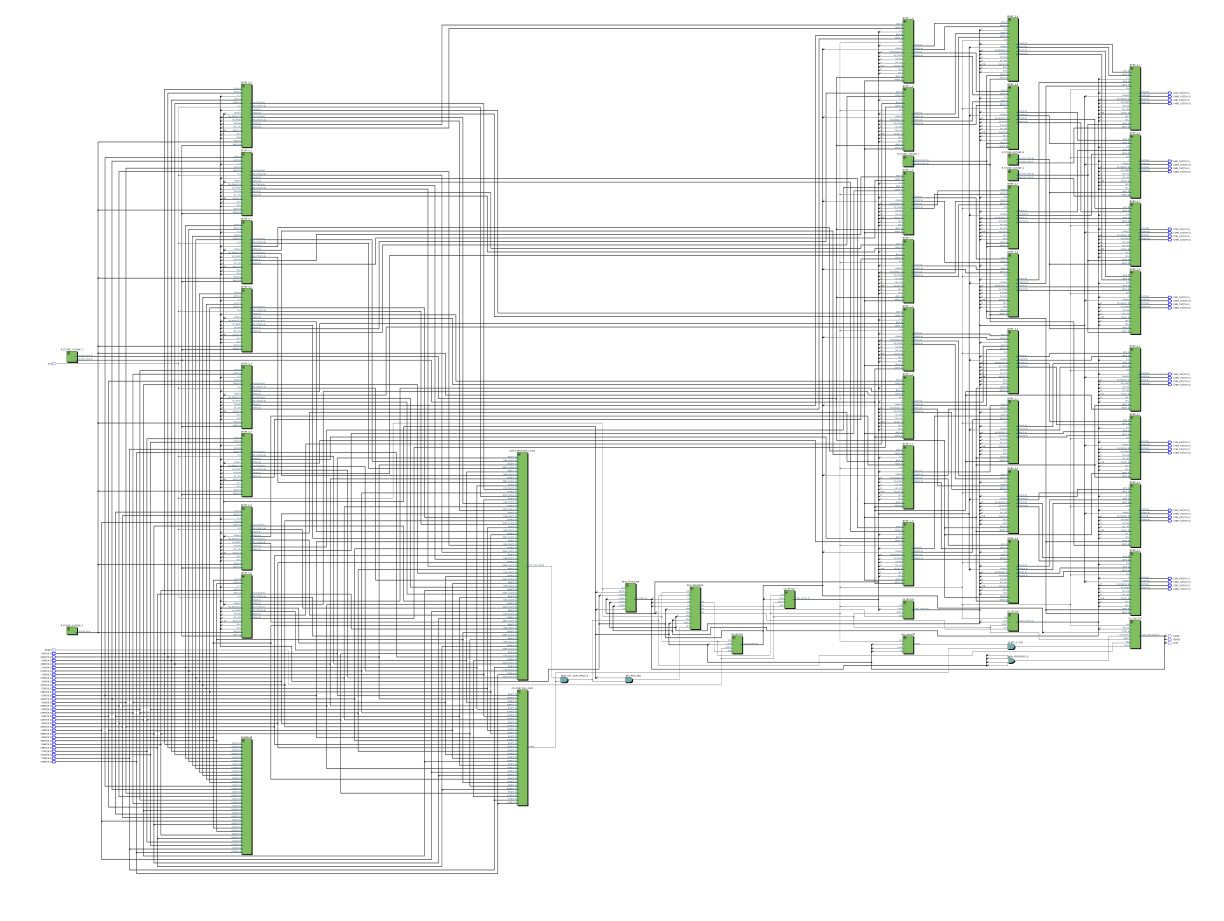
B1I **=** **[**0**,**0**,**0**,**0**,**0**,**0**,**0**,**0**]**

Confrontando i risultati ottenuti dallo script di Matlab e i valori in uscita dalla FFT, simulati con

Modelsim, possono notarsi delle piccole incongruenze per quanto riguarda l’arrotondamento e

l’errore macchina dovuto all’aritmetica finita della soluzione implementata. Nonostante ciò, i risultati ottenuti sono congruenti con quanto previsto teoricamente nella simulazione MATLAB.

# Appendice



**FFT\_16\_tb**

**LIBRARY** ieee**;**

**USE** ieee**.**std\_logic\_1164**.all;**

**USE** ieee**.**numeric\_std**.all;**

**ENTITY** FFT\_16\_tb **IS**

**END** FFT\_16\_tb**;**

**ARCHITECTURE** behav **OF** FFT\_16\_tb **IS**

**SIGNAL** CLK\_tb**,**RESET\_tb**,**DONE\_tb**,**READY\_tb**:**STD\_LOGIC**;**

**SIGNAL** C0AR\_tb**,**C1AR\_tb**,**C2AR\_tb**,**C3AR\_tb**,**C4AR\_tb**,**C5AR\_tb**,**C6AR\_tb**,**C7AR\_tb**:** SIGNED**(**19 **downto** 0**);**

**SIGNAL** C0AI\_tb**,**C1AI\_tb**,**C2AI\_tb**,**C3AI\_tb**,**C4AI\_tb**,**C5AI\_tb**,**C6AI\_tb**,**C7AI\_tb**:** SIGNED**(**19 **downto** 0**);**

**SIGNAL** C0BR\_tb**,**C1BR\_tb**,**C2BR\_tb**,**C3BR\_tb**,**C4BR\_tb**,**C5BR\_tb**,**C6BR\_tb**,**C7BR\_tb**:** SIGNED**(**19 **downto** 0**);**

**SIGNAL** C0BI\_tb**,**C1BI\_tb**,**C2BI\_tb**,**C3BI\_tb**,**C4BI\_tb**,**C5BI\_tb**,**C6BI\_tb**,**C7BI\_tb**:** SIGNED**(**19 **downto** 0**);**

**SIGNAL** C0AR\_OUT\_tb**,**C1AR\_OUT\_tb**,**C2AR\_OUT\_tb**,**C3AR\_OUT\_tb**,**C4AR\_OUT\_tb**,**C5AR\_OUT\_tb**,**C6AR\_OUT\_tb**,**C7AR\_OUT\_tb**:** SIGNED**(**19 **downto** 0**);**

**SIGNAL** C0AI\_OUT\_tb**,**C1AI\_OUT\_tb**,**C2AI\_OUT\_tb**,**C3AI\_OUT\_tb**,**C4AI\_OUT\_tb**,**C5AI\_OUT\_tb**,**C6AI\_OUT\_tb**,**C7AI\_OUT\_tb**:** SIGNED**(**19 **downto** 0**);**

**SIGNAL** C0BR\_OUT\_tb**,**C1BR\_OUT\_tb**,**C2BR\_OUT\_tb**,**C3BR\_OUT\_tb**,**C4BR\_OUT\_tb**,**C5BR\_OUT\_tb**,**C6BR\_OUT\_tb**,**C7BR\_OUT\_tb**:** SIGNED**(**19 **downto** 0**);**

**SIGNAL** C0BI\_OUT\_tb**,**C1BI\_OUT\_tb**,**C2BI\_OUT\_tb**,**C3BI\_OUT\_tb**,**C4BI\_OUT\_tb**,**C5BI\_OUT\_tb**,**C6BI\_OUT\_tb**,**C7BI\_OUT\_tb**:** SIGNED**(**19 **downto** 0**);**

**COMPONENT** FFT\_16 **IS**

**PORT** **(** CLK**,**RESET**:** **IN** STD\_LOGIC**;**

DONE**,**READY**:** **OUT** STD\_LOGIC**;**

C0AR**,**C1AR**,**C2AR**,**C3AR**,**C4AR**,**C5AR**,**C6AR**,**C7AR**:** **IN** SIGNED**(**19 **downto** 0**);**

C0AI**,**C1AI**,**C2AI**,**C3AI**,**C4AI**,**C5AI**,**C6AI**,**C7AI**:** **IN** SIGNED**(**19 **downto** 0**);**

C0BR**,**C1BR**,**C2BR**,**C3BR**,**C4BR**,**C5BR**,**C6BR**,**C7BR**:** **IN** SIGNED**(**19 **downto** 0**);**

C0BI**,**C1BI**,**C2BI**,**C3BI**,**C4BI**,**C5BI**,**C6BI**,**C7BI**:** **IN** SIGNED**(**19 **downto** 0**);**

C0AR\_OUT**,**C1AR\_OUT**,**C2AR\_OUT**,**C3AR\_OUT**,**C4AR\_OUT**,**C5AR\_OUT**,**C6AR\_OUT**,**C7AR\_OUT**:**

**OUT** SIGNED**(**19 **downto** 0**);**

C0AI\_OUT**,**C1AI\_OUT**,**C2AI\_OUT**,**C3AI\_OUT**,**C4AI\_OUT**,**C5AI\_OUT**,**C6AI\_OUT**,**C7AI\_OUT**:**

**OUT** SIGNED**(**19 **downto** 0**);**

C0BR\_OUT**,**C1BR\_OUT**,**C2BR\_OUT**,**C3BR\_OUT**,**C4BR\_OUT**,**C5BR\_OUT**,**C6BR\_OUT**,**C7BR\_OUT**:**

**OUT** SIGNED**(**19 **downto** 0**);**

C0BI\_OUT**,**C1BI\_OUT**,**C2BI\_OUT**,**C3BI\_OUT**,**C4BI\_OUT**,**C5BI\_OUT**,**C6BI\_OUT**,**C7BI\_OUT**:**

**OUT** SIGNED**(**19 **downto** 0**));**

**END** **COMPONENT;**

**BEGIN**

FFT**:** FFT\_16 **PORT** **MAP(**CLK**=>**CLK\_tb**,**RESET**=>**RESET\_tb**,**DONE**=>**DONE\_tb**,**READY**=>**READY\_tb**,**

C0AR**=>**C0AR\_tb**,**C1AR**=>**C1AR\_tb**,**C2AR**=>**C2AR\_tb**,**C3AR**=>**C3AR\_tb**,**C4AR**=>**C4AR\_tb**,**C5AR**=>**C5AR\_tb**,**C6AR**=>**C6AR\_tb**,**C7AR**=>**C7AR\_tb**,**

C0AI**=>**C0AI\_tb**,**C1AI**=>**C1AI\_tb**,**C2AI**=>**C2AI\_tb**,**C3AI**=>**C3AI\_tb**,**C4AI**=>**C4AI\_tb**,**C5AI**=>**C5AI\_tb**,**C6AI**=>**C6AI\_tb**,**C7AI**=>**C7AI\_tb**,**

C0BR**=>**C0BR\_tb**,**C1BR**=>**C1BR\_tb**,**C2BR**=>**C2BR\_tb**,**C3BR**=>**C3BR\_tb**,**C4BR**=>**C4BR\_tb**,**C5BR**=>**C5BR\_tb**,**C6BR**=>**C6BR\_tb**,**C7BR**=>**C7BR\_tb**,**

C0BI**=>**C0BI\_tb**,**C1BI**=>**C1BI\_tb**,**C2BI**=>**C2BI\_tb**,**C3BI**=>**C3BI\_tb**,**C4BI**=>**C4BI\_tb**,**C5BI**=>**C5BI\_tb**,**C6BI**=>**C6BI\_tb**,**C7BI**=>**C7BI\_tb**,**

C0AR\_OUT**=>**C0AR\_OUT\_tb**,**C1AR\_OUT**=>**C1AR\_OUT\_tb**,**C2AR\_OUT**=>**C2AR\_OUT\_tb**,**C3AR\_OUT**=>**C3AR\_OUT\_tb**,**

C4AR\_OUT**=>**C4AR\_OUT\_tb**,**C5AR\_OUT**=>**C5AR\_OUT\_tb**,**C6AR\_OUT**=>**C6AR\_OUT\_tb**,**C7AR\_OUT**=>**C7AR\_OUT\_tb**,**

C0AI\_OUT**=>**C0AI\_OUT\_tb**,**C1AI\_OUT**=>**C1AI\_OUT\_tb**,**C2AI\_OUT**=>**C2AI\_OUT\_tb**,**C3AI\_OUT**=>**C3AI\_OUT\_tb**,**

C4AI\_OUT**=>**C4AI\_OUT\_tb**,**C5AI\_OUT**=>**C5AI\_OUT\_tb**,**C6AI\_OUT**=>**C6AI\_OUT\_tb**,**C7AI\_OUT**=>**C7AI\_OUT\_tb**,**

C0BR\_OUT**=>**C0BR\_OUT\_tb**,**C1BR\_OUT**=>**C1BR\_OUT\_tb**,**C2BR\_OUT**=>**C2BR\_OUT\_tb**,**C3BR\_OUT**=>**C3BR\_OUT\_tb**,**

C4BR\_OUT**=>**C4BR\_OUT\_tb**,**C5BR\_OUT**=>**C5BR\_OUT\_tb**,**C6BR\_OUT**=>**C6BR\_OUT\_tb**,**C7BR\_OUT**=>**C7BR\_OUT\_tb**,**

C0BI\_OUT**=>**C0BI\_OUT\_tb**,**C1BI\_OUT**=>**C1BI\_OUT\_tb**,**C2BI\_OUT**=>**C2BI\_OUT\_tb**,**C3BI\_OUT**=>**C3BI\_OUT\_tb**,**

C4BI\_OUT**=>**C4BI\_OUT\_tb**,**C5BI\_OUT**=>**C5BI\_OUT\_tb**,**C6BI\_OUT**=>**C6BI\_OUT\_tb**,**C7BI\_OUT**=>**C7BI\_OUT\_tb**);**

clk\_process**:** **PROCESS**

**BEGIN**

CLK\_tb**<=** '0'**;**

**wait** **for** 10 ns**;**

CLK\_tb **<=** '1'**;**

**wait** **for** 10 ns**;**

**end** **process;**

-- 0 : 00000000000000000000

-- -1 : 10000000000000000000

-- 1 : 01111111111111111111

-- 0.5: 01000000000000000000

-- -0.5:11000000000000000000

-- 0.75:01100000000000000000

ingressi**:** **PROCESS**

**BEGIN**

C0AR\_tb**<=**"00000000000000000000"**;**

C1AR\_tb**<=**"00000000000000000000"**;**

C2AR\_tb**<=**"00000000000000000000"**;**

C3AR\_tb**<=**"00000000000000000000"**;**

C4AR\_tb**<=**"00000000000000000000"**;**

C5AR\_tb**<=**"00000000000000000000"**;**

C6AR\_tb**<=**"00000000000000000000"**;**

C7AR\_tb**<=**"00000000000000000000"**;**

C0AI\_tb**<=**"00000000000000000000"**;**

C1AI\_tb**<=**"00000000000000000000"**;**

C2AI\_tb**<=**"00000000000000000000"**;**

C3AI\_tb**<=**"00000000000000000000"**;**

C4AI\_tb**<=**"00000000000000000000"**;**

C5AI\_tb**<=**"00000000000000000000"**;**

C6AI\_tb**<=**"00000000000000000000"**;**

C7AI\_tb**<=**"00000000000000000000"**;**

C0BR\_tb**<=**"00000000000000000000"**;**

C1BR\_tb**<=**"00000000000000000000"**;**

C2BR\_tb**<=**"00000000000000000000"**;**

C3BR\_tb**<=**"00000000000000000000"**;**

C4BR\_tb**<=**"00000000000000000000"**;**

C5BR\_tb**<=**"00000000000000000000"**;**

C6BR\_tb**<=**"00000000000000000000"**;**

C7BR\_tb**<=**"00000000000000000000"**;**

C0BI\_tb**<=**"00000000000000000000"**;**

C1BI\_tb**<=**"00000000000000000000"**;**

C2BI\_tb**<=**"00000000000000000000"**;**

C3BI\_tb**<=**"00000000000000000000"**;**

C4BI\_tb**<=**"00000000000000000000"**;**

C5BI\_tb**<=**"00000000000000000000"**;**

C6BI\_tb**<=**"00000000000000000000"**;**

C7BI\_tb**<=**"00000000000000000000"**;**

**wait** **for** 50 ns**;**

-- CAMPIONE 1

C0AR\_tb**<=**"10000000000000000000"**;**

C1AR\_tb**<=**"10000000000000000000"**;**

C2AR\_tb**<=**"10000000000000000000"**;**

C3AR\_tb**<=**"10000000000000000000"**;**

C4AR\_tb**<=**"10000000000000000000"**;**

C5AR\_tb**<=**"10000000000000000000"**;**

C6AR\_tb**<=**"10000000000000000000"**;**

C7AR\_tb**<=**"10000000000000000000"**;**

C0AI\_tb**<=**"00000000000000000000"**;**

C1AI\_tb**<=**"00000000000000000000"**;**

C2AI\_tb**<=**"00000000000000000000"**;**

C3AI\_tb**<=**"00000000000000000000"**;**

C4AI\_tb**<=**"00000000000000000000"**;**

C5AI\_tb**<=**"00000000000000000000"**;**

C6AI\_tb**<=**"00000000000000000000"**;**

C7AI\_tb**<=**"00000000000000000000"**;**

C0BR\_tb**<=**"10000000000000000000"**;**

C1BR\_tb**<=**"10000000000000000000"**;**

C2BR\_tb**<=**"10000000000000000000"**;**

C3BR\_tb**<=**"10000000000000000000"**;**

C4BR\_tb**<=**"10000000000000000000"**;**

C5BR\_tb**<=**"10000000000000000000"**;**

C6BR\_tb**<=**"10000000000000000000"**;**

C7BR\_tb**<=**"10000000000000000000"**;**

C0BI\_tb**<=**"00000000000000000000"**;**

C1BI\_tb**<=**"00000000000000000000"**;**

C2BI\_tb**<=**"00000000000000000000"**;**

C3BI\_tb**<=**"00000000000000000000"**;**

C4BI\_tb**<=**"00000000000000000000"**;**

C5BI\_tb**<=**"00000000000000000000"**;**

C6BI\_tb**<=**"00000000000000000000"**;**

C7BI\_tb**<=**"00000000000000000000"**;**

**wait** **for** 140 ns**;**

-- CAMPIONE 2

C0AR\_tb**<=**"10000000000000000000"**;**

C1AR\_tb**<=**"00000000000000000000"**;**

C2AR\_tb**<=**"01111111111111111111"**;**

C3AR\_tb**<=**"00000000000000000000"**;**

C4AR\_tb**<=**"10000000000000000000"**;**

C5AR\_tb**<=**"00000000000000000000"**;**

C6AR\_tb**<=**"01111111111111111111"**;**

C7AR\_tb**<=**"00000000000000000000"**;**

C0AI\_tb**<=**"00000000000000000000"**;**

C1AI\_tb**<=**"00000000000000000000"**;**

C2AI\_tb**<=**"00000000000000000000"**;**

C3AI\_tb**<=**"00000000000000000000"**;**

C4AI\_tb**<=**"00000000000000000000"**;**

C5AI\_tb**<=**"00000000000000000000"**;**

C6AI\_tb**<=**"00000000000000000000"**;**

C7AI\_tb**<=**"00000000000000000000"**;**

C0BR\_tb**<=**"10000000000000000000"**;**

C1BR\_tb**<=**"00000000000000000000"**;**

C2BR\_tb**<=**"01111111111111111111"**;**

C3BR\_tb**<=**"00000000000000000000"**;**

C4BR\_tb**<=**"10000000000000000000"**;**

C5BR\_tb**<=**"00000000000000000000"**;**

C6BR\_tb**<=**"01111111111111111111"**;**

C7BR\_tb**<=**"00000000000000000000"**;**

C0BI\_tb**<=**"00000000000000000000"**;**

C1BI\_tb**<=**"00000000000000000000"**;**

C2BI\_tb**<=**"00000000000000000000"**;**

C3BI\_tb**<=**"00000000000000000000"**;**

C4BI\_tb**<=**"00000000000000000000"**;**

C5BI\_tb**<=**"00000000000000000000"**;**

C6BI\_tb**<=**"00000000000000000000"**;**

C7BI\_tb**<=**"00000000000000000000"**;**

**wait** **for** 140 ns**;**

-- CAMPIONE 3

C0AR\_tb**<=**"01111111111111111111"**;**

C1AR\_tb**<=**"00000000000000000000"**;**

C2AR\_tb**<=**"00000000000000000000"**;**

C3AR\_tb**<=**"00000000000000000000"**;**

C4AR\_tb**<=**"00000000000000000000"**;**

C5AR\_tb**<=**"00000000000000000000"**;**

C6AR\_tb**<=**"00000000000000000000"**;**

C7AR\_tb**<=**"00000000000000000000"**;**

C0AI\_tb**<=**"00000000000000000000"**;**

C1AI\_tb**<=**"00000000000000000000"**;**

C2AI\_tb**<=**"00000000000000000000"**;**

C3AI\_tb**<=**"00000000000000000000"**;**

C4AI\_tb**<=**"00000000000000000000"**;**

C5AI\_tb**<=**"00000000000000000000"**;**

C6AI\_tb**<=**"00000000000000000000"**;**

C7AI\_tb**<=**"00000000000000000000"**;**

C0BR\_tb**<=**"00000000000000000000"**;**

C1BR\_tb**<=**"00000000000000000000"**;**

C2BR\_tb**<=**"00000000000000000000"**;**

C3BR\_tb**<=**"00000000000000000000"**;**

C4BR\_tb**<=**"00000000000000000000"**;**

C5BR\_tb**<=**"00000000000000000000"**;**

C6BR\_tb**<=**"00000000000000000000"**;**

C7BR\_tb**<=**"00000000000000000000"**;**

C0BI\_tb**<=**"00000000000000000000"**;**

C1BI\_tb**<=**"00000000000000000000"**;**

C2BI\_tb**<=**"00000000000000000000"**;**

C3BI\_tb**<=**"00000000000000000000"**;**

C4BI\_tb**<=**"00000000000000000000"**;**

C5BI\_tb**<=**"00000000000000000000"**;**

C6BI\_tb**<=**"00000000000000000000"**;**

C7BI\_tb**<=**"00000000000000000000"**;**

**wait** **for** 140 ns**;**

-- CAMPIONE 4

C0AR\_tb**<=**"10000000000000000000"**;**

C1AR\_tb**<=**"10000000000000000000"**;**

C2AR\_tb**<=**"01111111111111111111"**;**

C3AR\_tb**<=**"01111111111111111111"**;**

C4AR\_tb**<=**"10000000000000000000"**;**

C5AR\_tb**<=**"10000000000000000000"**;**

C6AR\_tb**<=**"01111111111111111111"**;**

C7AR\_tb**<=**"01111111111111111111"**;**

C0AI\_tb**<=**"00000000000000000000"**;**

C1AI\_tb**<=**"00000000000000000000"**;**

C2AI\_tb**<=**"00000000000000000000"**;**

C3AI\_tb**<=**"00000000000000000000"**;**

C4AI\_tb**<=**"00000000000000000000"**;**

C5AI\_tb**<=**"00000000000000000000"**;**

C6AI\_tb**<=**"00000000000000000000"**;**

C7AI\_tb**<=**"00000000000000000000"**;**

C0BR\_tb**<=**"10000000000000000000"**;**

C1BR\_tb**<=**"10000000000000000000"**;**

C2BR\_tb**<=**"01111111111111111111"**;**

C3BR\_tb**<=**"01111111111111111111"**;**

C4BR\_tb**<=**"10000000000000000000"**;**

C5BR\_tb**<=**"10000000000000000000"**;**

C6BR\_tb**<=**"01111111111111111111"**;**

C7BR\_tb**<=**"01111111111111111111"**;**

C0BI\_tb**<=**"00000000000000000000"**;**

C1BI\_tb**<=**"00000000000000000000"**;**

C2BI\_tb**<=**"00000000000000000000"**;**

C3BI\_tb**<=**"00000000000000000000"**;**

C4BI\_tb**<=**"00000000000000000000"**;**

C5BI\_tb**<=**"00000000000000000000"**;**

C6BI\_tb**<=**"00000000000000000000"**;**

C7BI\_tb**<=**"00000000000000000000"**;**

**wait** **for** 140 ns**;**

-- CAMPIONE 5

C0AR\_tb**<=**"01000000000000000000"**;**

C1AR\_tb**<=**"01000000000000000000"**;**

C2AR\_tb**<=**"01000000000000000000"**;**

C3AR\_tb**<=**"01000000000000000000"**;**

C4AR\_tb**<=**"01000000000000000000"**;**

C5AR\_tb**<=**"01000000000000000000"**;**

C6AR\_tb**<=**"01000000000000000000"**;**

C7AR\_tb**<=**"01000000000000000000"**;**

C0AI\_tb**<=**"00000000000000000000"**;**

C1AI\_tb**<=**"00000000000000000000"**;**

C2AI\_tb**<=**"00000000000000000000"**;**

C3AI\_tb**<=**"00000000000000000000"**;**

C4AI\_tb**<=**"00000000000000000000"**;**

C5AI\_tb**<=**"00000000000000000000"**;**

C6AI\_tb**<=**"00000000000000000000"**;**

C7AI\_tb**<=**"00000000000000000000"**;**

C0BR\_tb**<=**"01000000000000000000"**;**

C1BR\_tb**<=**"11000000000000000000"**;**

C2BR\_tb**<=**"11000000000000000000"**;**

C3BR\_tb**<=**"11000000000000000000"**;**

C4BR\_tb**<=**"11000000000000000000"**;**

C5BR\_tb**<=**"11000000000000000000"**;**

C6BR\_tb**<=**"11000000000000000000"**;**

C7BR\_tb**<=**"11000000000000000000"**;**

C0BI\_tb**<=**"00000000000000000000"**;**

C1BI\_tb**<=**"00000000000000000000"**;**

C2BI\_tb**<=**"00000000000000000000"**;**

C3BI\_tb**<=**"00000000000000000000"**;**

C4BI\_tb**<=**"00000000000000000000"**;**

C5BI\_tb**<=**"00000000000000000000"**;**

C6BI\_tb**<=**"00000000000000000000"**;**

C7BI\_tb**<=**"00000000000000000000"**;**

**wait** **for** 140 ns**;**

-- CAMPIONE 6

C0AR\_tb**<=**"00000000000000000000"**;**

C1AR\_tb**<=**"00000000000000000000"**;**

C2AR\_tb**<=**"00000000000000000000"**;**

C3AR\_tb**<=**"00000000000000000000"**;**

C4AR\_tb**<=**"00000000000000000000"**;**

C5AR\_tb**<=**"00000000000000000000"**;**

C6AR\_tb**<=**"00000000000000000000"**;**

C7AR\_tb**<=**"00000000000000000000"**;**

C0AI\_tb**<=**"00000000000000000000"**;**

C1AI\_tb**<=**"00000000000000000000"**;**

C2AI\_tb**<=**"00000000000000000000"**;**

C3AI\_tb**<=**"00000000000000000000"**;**

C4AI\_tb**<=**"00000000000000000000"**;**

C5AI\_tb**<=**"00000000000000000000"**;**

C6AI\_tb**<=**"00000000000000000000"**;**

C7AI\_tb**<=**"00000000000000000000"**;**

C0BR\_tb**<=**"01100000000000000000"**;**

C1BR\_tb**<=**"00000000000000000000"**;**

C2BR\_tb**<=**"00000000000000000000"**;**

C3BR\_tb**<=**"00000000000000000000"**;**

C4BR\_tb**<=**"00000000000000000000"**;**

C5BR\_tb**<=**"00000000000000000000"**;**

C6BR\_tb**<=**"00000000000000000000"**;**

C7BR\_tb**<=**"00000000000000000000"**;**

C0BI\_tb**<=**"00000000000000000000"**;**

C1BI\_tb**<=**"00000000000000000000"**;**

C2BI\_tb**<=**"00000000000000000000"**;**

C3BI\_tb**<=**"00000000000000000000"**;**

C4BI\_tb**<=**"00000000000000000000"**;**

C5BI\_tb**<=**"00000000000000000000"**;**

C6BI\_tb**<=**"00000000000000000000"**;**

C7BI\_tb**<=**"00000000000000000000"**;**

**wait** **for** 140 ns**;**

C0AR\_tb**<=**"00000000000000000000"**;**

C1AR\_tb**<=**"00000000000000000000"**;**

C2AR\_tb**<=**"00000000000000000000"**;**

C3AR\_tb**<=**"00000000000000000000"**;**

C4AR\_tb**<=**"00000000000000000000"**;**

C5AR\_tb**<=**"00000000000000000000"**;**

C6AR\_tb**<=**"00000000000000000000"**;**

C7AR\_tb**<=**"00000000000000000000"**;**

C0AI\_tb**<=**"00000000000000000000"**;**

C1AI\_tb**<=**"00000000000000000000"**;**

C2AI\_tb**<=**"00000000000000000000"**;**

C3AI\_tb**<=**"00000000000000000000"**;**

C4AI\_tb**<=**"00000000000000000000"**;**

C5AI\_tb**<=**"00000000000000000000"**;**

C6AI\_tb**<=**"00000000000000000000"**;**

C7AI\_tb**<=**"00000000000000000000"**;**

C0BR\_tb**<=**"00000000000000000000"**;**

C1BR\_tb**<=**"00000000000000000000"**;**

C2BR\_tb**<=**"00000000000000000000"**;**

C3BR\_tb**<=**"00000000000000000000"**;**

C4BR\_tb**<=**"00000000000000000000"**;**

C5BR\_tb**<=**"00000000000000000000"**;**

C6BR\_tb**<=**"00000000000000000000"**;**

C7BR\_tb**<=**"00000000000000000000"**;**

C0BI\_tb**<=**"00000000000000000000"**;**

C1BI\_tb**<=**"00000000000000000000"**;**

C2BI\_tb**<=**"00000000000000000000"**;**

C3BI\_tb**<=**"00000000000000000000"**;**

C4BI\_tb**<=**"00000000000000000000"**;**

C5BI\_tb**<=**"00000000000000000000"**;**

C6BI\_tb**<=**"00000000000000000000"**;**

C7BI\_tb**<=**"00000000000000000000"**;**

**wait** **for** 5000 ns**;**

**END** **PROCESS;**

**PROCESS**

**BEGIN**

RESET\_tb**<=**'1'**;**

**wait** **for** 10 ns**;**

RESET\_tb**<=**'0'**;**

**wait** **for** 10000 ns**;**

**END** **PROCESS;**

**END** behav**;**

**BF.VHD**

**LIBRARY** ieee**;**

**USE** ieee**.**std\_logic\_1164**.all;**

**USE** ieee**.**numeric\_std**.all;**

**ENTITY** BF **IS**

**PORT** **(** CLK**,** ENABLE**:** **IN** std\_logic**;**

AI**,**AR**,**BI**,**BR**,**WR**,**WI**:** **IN** SIGNED**(**19 **downto** 0**);**

SEL\_INV**,**SEL3**,**SEL1**,**SELSUM**,**C**,**A\_S**,**EN\_REGR **:** **IN** std\_logic**;**

SEL2 **:** **IN** std\_logic\_vector**(**1 **downto** 0**);**

EN\_REGO **:** **IN** std\_logic\_vector**(**2 **downto** 0**);**

A1R**,**A1I**,**B1R**,**B1I**,**AI\_CTRL**,**AR\_CTRL**,**BI\_CTRL**,**BR\_CTRL**:** **OUT** SIGNED**(**19 **downto** 0**));**

**END** BF**;**

**ARCHITECTURE** BEHAV **OF** BF **IS**

**COMPONENT** RF **IS**

**PORT** **(** CLK**,** ENABLE**:** **IN** std\_logic**;**

AI**,**AR**,**BI**,**BR**,**WR**,**WI**:** **IN** SIGNED**(**19 **downto** 0**);**

SEL3**,**SEL1 **:** **IN** std\_logic**;**

SEL2 **:** **IN** std\_logic\_vector**(**1 **downto** 0**);**

OUT\_MUX1**,**OUT\_MUX2**,**OUT\_MUX3**,**AI\_CTRL**,**AR\_CTRL**,**BI\_CTRL**,**BR\_CTRL**:** **OUT** SIGNED**(**19 **downto** 0**));**

**END** **COMPONENT;**

**COMPONENT** MOL **IS**

**PORT** **(** C**,**CLK**:** **IN** std\_logic**;**

ADD1**,**ADD2**:** **IN** SIGNED**(**19 **downto** 0**);**

RESULT**:** **OUT** SIGNED**(**38 **downto** 0**));**

**END** **COMPONENT;**

**COMPONENT** SUM **IS**

**PORT** **(** A\_S**,**CLK**:** **IN** std\_logic**;**

T1**,**T2**:** **IN** SIGNED**(**38 **downto** 0**);**

RESULT**:** **OUT** SIGNED**(**39 **downto** 0**));**

**END** **COMPONENT;**

**COMPONENT** REG\_39\_M **IS**

**PORT** **(** CLK**:** **IN** std\_logic**;**

D**:** **IN** SIGNED**(**38 **downto** 0**);**

Q**:** **OUT** SIGNED**(**38 **downto** 0**));**

**END** **COMPONENT;**

**COMPONENT** REG\_40\_S **IS**

**PORT** **(** CLK**:** **IN** std\_logic**;**

D**:** **IN** SIGNED**(**39 **downto** 0**);**

Q**:** **OUT** SIGNED**(**39 **downto** 0**));**

**END** **COMPONENT;**

**COMPONENT** REG\_ROUND **IS**

**PORT** **(** CLK**:** **IN** std\_logic**;**

ENABLE**:** **IN** std\_logic**;**

D**:** **IN** SIGNED**(**39 **downto** 0**);**

Q**:** **OUT** SIGNED**(**39 **downto** 0**));**

**END** **COMPONENT;**

**COMPONENT** MUX39SUM **IS**

**PORT(** A**:** **IN** SIGNED **(**19 **downto** 0**);**

B**:** **IN** SIGNED**(**38 **downto** 0**);**

SEL **:** **IN** std\_logic**;**

OUTPUT**:** **OUT** SIGNED**(**38 **downto** 0**));**

**END** **COMPONENT;**

**COMPONENT** REG\_A1R **IS**

**PORT** **(** CLK**:** **IN** std\_logic**;**

ENABLE**:** **IN** std\_logic\_vector**(**2 **downto** 0**);**

D**:** **IN** SIGNED**(**19 **downto** 0**);**

Q**:** **OUT** SIGNED**(**19 **downto** 0**));**

**END** **COMPONENT;**

**COMPONENT** REG\_A1I **IS**

**PORT** **(** CLK**:** **IN** std\_logic**;**

ENABLE**:** **IN** std\_logic\_vector**(**2 **downto** 0**);**

D**:** **IN** SIGNED**(**19 **downto** 0**);**

Q**:** **OUT** SIGNED**(**19 **downto** 0**));**

**END** **COMPONENT;**

**COMPONENT** REG\_B1R **IS**

**PORT** **(** CLK**:** **IN** std\_logic**;**

ENABLE**:** **IN** std\_logic\_vector**(**2 **downto** 0**);**

D**:** **IN** SIGNED**(**19 **downto** 0**);**

Q**:** **OUT** SIGNED**(**19 **downto** 0**));**

**END** **COMPONENT;**

**COMPONENT** REG\_B1I **IS**

**PORT** **(** CLK**:** **IN** std\_logic**;**

ENABLE**:** **IN** std\_logic\_vector**(**2 **downto** 0**);**

D**:** **IN** SIGNED**(**19 **downto** 0**);**

Q**:** **OUT** SIGNED**(**19 **downto** 0**));**

**END** **COMPONENT;**

**COMPONENT** ROUNDING **IS**

**PORT** **(** IN\_39**:** **IN** SIGNED**(**39 **downto** 0**);**

OUT\_20**:** **OUT** SIGNED**(**19 **downto** 0**));**

**END** **COMPONENT;**

**COMPONENT** MUX\_INV **IS**

**PORT(** A**:** **IN** SIGNED **(**38 **downto** 0**);**

B**:** **IN** SIGNED**(**38 **downto** 0**);**

SEL **:** **IN** std\_logic**;**

OUTPUT**:** **OUT** SIGNED**(**38 **downto** 0**));**

**END** **COMPONENT;**

**SIGNAL** BUSW**,**BUS1**,**BUS2**,**ROUNDED**:** SIGNED **(**19 **downto** 0**);**

**SIGNAL** MUXSUM\_OUT**,**MOL\_RES\_IN**,**MOL\_RES\_OUT**,**OUTMUXINVT1**,**OUTMUXINVT2**:** SIGNED**(**38 **downto** 0**);**

**SIGNAL** SUM\_RES\_IN**,**SUM\_RES\_OUT**,**REG\_ROUND\_OUT**:** SIGNED**(**39 **DOWNTO** 0**);**

**SIGNAL** NOTHING**:** std\_logic**;**

**BEGIN**

ROUND**:** ROUNDING **PORT** **MAP(**IN\_39**=>**REG\_ROUND\_OUT**,**OUT\_20**=>**ROUNDED**);**

MUXSUM**:** MUX39SUM **PORT** **MAP(**A**=>**BUS1**,**B**=>**SUM\_RES\_OUT**(**38 **DOWNTO** 0**),**SEL**=>**SELSUM**,** OUTPUT**=>**MUXSUM\_OUT**);**

REG\_RF**:** RF **PORT** **MAP** **(**CLK**=>**CLK**,**ENABLE**=>**ENABLE**,**AI**=>**AI**,**AR**=>**AR**,**BI**=>**BI**,**BR**=>**BR**,**WR**=>**WR**,** WI**=>**WI**,**SEL1**=>**SEL1**,**SEL2**=>**SEL2**,**SEL3**=>**SEL3**,**AI\_CTRL**=>**AI\_CTRL**,**AR\_CTRL**=>**AR\_CTRL**,**BI\_CTRL**=>**BI\_CTRL**,**BR\_CTRL**=>**BR\_CTRL**,**OUT\_MUX1**=>**BUS1**,**OUT\_MUX2**=>**BUS2**,**OUT\_MUX3**=>** BUSW**);**

MOLTIPLICATORE**:** MOL **PORT** **MAP(**CLK**=>**CLK**,**C**=>**C**,**ADD1**=>**BUS2**,**ADD2**=>**BUSW**,**RESULT**=>** MOL\_RES\_IN**);**

SOMMATORE**:** SUM **PORT** **MAP(**CLK**=>**CLK**,**A\_S**=>**A\_S**,**T1**=>**OUTMUXINVT1**,**T2**=>**OUTMUXINVT2**,** RESULT**=>**SUM\_RES\_IN**);**

REG\_MOL**:** REG\_39\_M **PORT** **MAP(**CLK**=>**CLK**,**D**=>**MOL\_RES\_IN**,**Q**=>**MOL\_RES\_OUT**);**

REG\_SUM **:** REG\_40\_S **PORT** **MAP(**CLK**=>**CLK**,**D**=>**SUM\_RES\_IN**,**Q**=>**SUM\_RES\_OUT**);**

R\_ROUNDING**:** REG\_ROUND **PORT** **MAP(**CLK**=>**CLK**,** ENABLE**=>**EN\_REGR**,** D**=>**SUM\_RES\_OUT**,**Q**=>**REG\_ROUND\_OUT**);**

R\_A1R**:** REG\_A1R **PORT** **MAP(**CLK**=>**CLK**,**ENABLE**=>**EN\_REGO**,**D**=>**ROUNDED**(**19 **DOWNTO**0**),**Q**=>**A1R**);**

R\_A1I**:** REG\_A1I **PORT** **MAP(**CLK**=>**CLK**,**ENABLE**=>**EN\_REGO**,**D**=>**ROUNDED**(**19 **DOWNTO**0**),**Q**=>**A1I**);**

R\_B1R**:** REG\_B1R **PORT** **MAP(**CLK**=>**CLK**,**ENABLE**=>**EN\_REGO**,**D**=>**ROUNDED**(**19 **DOWNTO**0**),**Q**=>**B1R**);**

R\_B1I**:** REG\_B1I **PORT** **MAP(**CLK**=>**CLK**,**ENABLE**=>**EN\_REGO**,**D**=>**ROUNDED**(**19 **DOWNTO**0**),**Q**=>**B1I**);**

INVT1**:** MUX\_INV **PORT MAP(**SEL**=>**SEL\_INV**,**A**=>**MUXSUM\_OUT**,**B**=>**MOL\_RES\_OUT**,** OUTPUT**=>**OUTMUXINVT1**);**

INVT2**:** MUX\_INV **PORT** **MAP(**SEL**=>**SEL\_INV**,**A**=>**MOL\_RES\_OUT**,**B**=>**MUXSUM\_OUT**,** OUTPUT**=>**OUTMUXINVT2**);**

**end** behav**;**

**MUX\_INV.VHD**

**LIBRARY** ieee**;**

**USE** ieee**.**std\_logic\_1164**.all;**

**USE** ieee**.**numeric\_std**.all;**

**ENTITY** MUX\_INV **IS**

**PORT(** A**:** **IN** SIGNED **(**38 **downto** 0**);**

B**:** **IN** SIGNED**(**38 **downto** 0**);**

SEL **:** **IN** std\_logic**;**

OUTPUT**:** **OUT** SIGNED**(**38 **downto** 0**));**

**END** MUX\_INV**;**

**ARCHITECTURE** behav **OF** MUX\_INV **IS**

**BEGIN**

**with** SEL **select**

OUTPUT **<=** A **when** '0'**,**

B **when** '1'**,**

"000000000000000000000000000000000000000" **when** **others;**

**end** Behav**;**

**MUX39SUM**

**LIBRARY** ieee**;**

**USE** ieee**.**std\_logic\_1164**.all;**

**USE** ieee**.**numeric\_std**.all;**

**ENTITY** MUX39SUM **IS**

**PORT(** A**:** **IN** SIGNED **(**19 **downto** 0**);** -- Q1.19

B**:** **IN** SIGNED**(**38 **downto** 0**);** -- Q1.38

SEL **:** **IN** std\_logic**;**

OUTPUT**:** **OUT** SIGNED**(**38 **downto** 0**));**

**END** MUX39SUM**;**

**ARCHITECTURE** behav **OF** MUX39SUM **IS**

**SIGNAL** sign\_ext\_a **:** SIGNED**(**18 **DOWNTO** 0**);**

**SIGNAL** new\_a **:** SIGNED **(**38 **DOWNTO** 0**);**

**BEGIN**

--sign\_ext\_a <=(others=> A(0)); -- estensione LSB nella parte decimale

sign\_ext\_a **<=(others=>** '0'**);** -- estensione zeri nella parte decimale

new\_a**(**38 **downto** 19**)** **<=** A**;**

new\_a**(**18 **downto** 0**)** **<=** sign\_ext\_a**;**

**with** SEL **select**

OUTPUT **<=** new\_a **when** '0'**,**

B **when** '1'**,**

"000000000000000000000000000000000000000" **when** **others;**

**end** Behav**;**

**REG\_39\_M.VHD**

**LIBRARY** ieee**;**

**USE** ieee**.**std\_logic\_1164**.all;**

**USE** ieee**.**numeric\_std**.all;**

**ENTITY** REG\_39\_M **IS**

**PORT** **(** CLK**:** **IN** std\_logic**;**

D**:** **IN** SIGNED**(**38 **downto** 0**);**

Q**:** **OUT** SIGNED**(**38 **downto** 0**));**

**END** REG\_39\_M**;**

**ARCHITECTURE** behav **OF** REG\_39\_M **IS**

**BEGIN**

**PROCESS(**CLK**)**

**BEGIN**

**IF** **(**CLK'**EVENT** **AND** CLK **=** '1'**)** **THEN**

Q**<=** D**;**

**END** **IF;**

**END** **PROCESS;**

**END** behav**;**

**REG\_40\_S.VHD**

**LIBRARY** ieee**;**

**USE** ieee**.**std\_logic\_1164**.all;**

**USE** ieee**.**numeric\_std**.all;**

**ENTITY** REG\_40\_S **IS**

**PORT** **(** CLK**:** **IN** std\_logic**;**

D**:** **IN** SIGNED**(**39 **downto** 0**);**

Q**:** **OUT** SIGNED**(**39 **downto** 0**));**

**END** REG\_40\_S**;**

**ARCHITECTURE** behav **OF** REG\_40\_S **IS**

**BEGIN**

**PROCESS(**CLK**)**

**BEGIN**

**IF** **(**CLK'**EVENT** **AND** CLK **=** '1'**)** **THEN**

Q**<=** D**;**

**END** **IF;**

**END** **PROCESS;**

**END** behav**;**

**REG\_A1I.VHD**

**LIBRARY** ieee**;**

**USE** ieee**.**std\_logic\_1164**.all;**

**USE** ieee**.**numeric\_std**.all;**

**ENTITY** REG\_A1I **IS**

**PORT** **(** CLK**:** **IN** std\_logic**;**

ENABLE**:** **IN** std\_logic\_vector**(**2 **downto** 0**);**

D**:** **IN** SIGNED**(**19 **downto** 0**);**

Q**:** **OUT** SIGNED**(**19 **downto** 0**));**

**END** REG\_A1I**;**

**ARCHITECTURE** behav **OF** REG\_A1I **IS**

**BEGIN**

**PROCESS(**CLK**)**

**BEGIN**

**IF** **(**CLK'**EVENT** **AND** CLK **=** '1'**)** **THEN**

**IF(**ENABLE **=** "010"**)** **THEN**

Q**<=** D**;**

**END** **IF;**

**END** **IF;**

**END** **PROCESS;**

**END** behav**;**

**REG\_A1R.VHD**

**LIBRARY** ieee**;**

**USE** ieee**.**std\_logic\_1164**.all;**

**USE** ieee**.**numeric\_std**.all;**

**ENTITY** REG\_A1R **IS**

**PORT** **(** CLK**:** **IN** std\_logic**;**

ENABLE**:** **IN** std\_logic\_vector**(**2 **downto** 0**);**

D**:** **IN** SIGNED**(**19 **downto** 0**);**

Q**:** **OUT** SIGNED**(**19 **downto** 0**));**

**END** REG\_A1R**;**

**ARCHITECTURE** behav **OF** REG\_A1R **IS**

**BEGIN**

**PROCESS(**CLK**)**

**BEGIN**

**IF** **(**CLK'**EVENT** **AND** CLK **=** '1'**)** **THEN**

**IF(**ENABLE **=** "001"**)** **THEN**

Q**<=** D**;**

**END** **IF;**

**END** **IF;**

**END** **PROCESS;**

**END** behav**;**

**REG\_B1I.VHD**

**LIBRARY** ieee**;**

**USE** ieee**.**std\_logic\_1164**.all;**

**USE** ieee**.**numeric\_std**.all;**

**ENTITY** REG\_B1I **IS**

**PORT** **(** CLK**:** **IN** std\_logic**;**

ENABLE**:** **IN** std\_logic\_vector**(**2 **downto** 0**);**

D**:** **IN** SIGNED**(**19 **downto** 0**);**

Q**:** **OUT** SIGNED**(**19 **downto** 0**));**

**END** REG\_B1I**;**

**ARCHITECTURE** behav **OF** REG\_B1I **IS**

**BEGIN**

**PROCESS(**CLK**)**

**BEGIN**

**IF** **(**CLK'**EVENT** **AND** CLK **=** '1'**)** **THEN**

**IF(**ENABLE **=** "100"**)** **THEN**

Q**<=** D**;**

**END** **IF;**

**END** **IF;**

**END** **PROCESS;**

**END** behav**;**

**REG\_B1R**

**LIBRARY** ieee**;**

**USE** ieee**.**std\_logic\_1164**.all;**

**USE** ieee**.**numeric\_std**.all;**

**ENTITY** REG\_B1R **IS**

**PORT** **(** CLK**:** **IN** std\_logic**;**

ENABLE**:** **IN** std\_logic\_vector**(**2 **downto** 0**);**

D**:** **IN** SIGNED**(**19 **downto** 0**);**

Q**:** **OUT** SIGNED**(**19 **downto** 0**));**

**END** REG\_B1R**;**

**ARCHITECTURE** behav **OF** REG\_B1R **IS**

**BEGIN**

**PROCESS(**CLK**)**

**BEGIN**

**IF** **(**CLK'**EVENT** **AND** CLK **=** '1'**)** **THEN**

**IF(**ENABLE **=** "011"**)** **THEN**

Q**<=** D**;**

**END** **IF;**

**END** **IF;**

**END** **PROCESS;**

**END** behav**;**

**REG\_ROUND.VHD**

**LIBRARY** ieee**;**

**USE** ieee**.**std\_logic\_1164**.all;**

**USE** ieee**.**numeric\_std**.all;**

**ENTITY** REG\_ROUND **IS**

**PORT** **(** CLK**:** **IN** std\_logic**;**

ENABLE**:** **IN** std\_logic**;**

D**:** **IN** SIGNED**(**39 **downto** 0**);**

Q**:** **OUT** SIGNED**(**39 **downto** 0**));**

**END** REG\_ROUND**;**

**ARCHITECTURE** behav **OF** REG\_ROUND **IS**

**BEGIN**

**PROCESS(**CLK**)**

**BEGIN**

**IF** **(**CLK'**EVENT** **AND** CLK **=** '1'**)** **THEN**

**IF(**ENABLE **=** '1'**)** **THEN**

Q**<=** D**;**

**END** **IF;**

**END** **IF;**

**END** **PROCESS;**

**END** behav**;**

**ROUNDING.VHD**

**LIBRARY** ieee**;**

**USE** ieee**.**std\_logic\_1164**.all;**

**USE** ieee**.**numeric\_std**.all;**

**ENTITY** ROUNDING **IS**

**PORT** **(** IN\_39**:** **IN** SIGNED**(**39 **downto** 0**);** -- Q2.38

OUT\_20**:** **OUT** SIGNED**(**19 **downto** 0**));**

**END** ROUNDING**;**

**ARCHITECTURE** behav **of** ROUNDING **IS**

**SIGNAL** OUT\_20\_S**:** SIGNED **(**19 **DOWNTO** 0**);**

**SIGNAL** IN\_39\_S**:** SIGNED**(**39 **DOWNTO** 0**);**

**BEGIN**

IN\_39\_S**(**39 **DOWNTO** 1**)** **<=** IN\_39**(**38 **DOWNTO** 0**);** -- sposto la virgola

IN\_39\_S**(**0**)** **<=**'0'**;** -- Q1.19

**PROCESS** **(**IN\_39\_S**)**

**BEGIN**

**case** IN\_39\_S**(**19 **DOWNTO** 0**)** **is**

**when** "10000000000000000000" **=>**

**IF** **(**IN\_39\_S**(**20**)** **=** '1'**)** **THEN**

OUT\_20\_S **<=** IN\_39\_S**(**39 **DOWNTO** 20**)** **+** 1**;**

**ELSE**

OUT\_20\_S **<=** IN\_39\_S**(**39 **DOWNTO** 20**);**

**END** **IF;**

**when** **OTHERS** **=>**

**IF** **(**IN\_39\_S**(**19**)** **=** '1'**)** **THEN**

OUT\_20\_S **<=** IN\_39\_S**(**39 **DOWNTO** 20**)** **+** 1**;**

**ELSE**

OUT\_20\_S **<=** IN\_39\_S**(**39 **DOWNTO** 20**);**

**END** **IF;**

**end** **case;**

**END** **PROCESS;**

OUT\_20**(**19**)** **<=** OUT\_20\_S**(**19**);** -- riduco la dinamica

OUT\_20**(**18 **DOWNTO** 0**)** **<=** OUT\_20\_S**(**19 **DOWNTO** 1**);**

**end** behav**;**

**CU\_BF.VHD**

**LIBRARY** ieee**;**

**USE** ieee**.**std\_logic\_1164**.all;**

**USE** ieee**.**numeric\_std**.all;**

**ENTITY** CU\_BF **IS**

**PORT(** LOAD**,**SEQ**,**CLK**,**RESET**:** **IN** std\_logic**;**

CTRL\_OUT**:** **OUT** std\_logic\_vector**(**14 **downto** 0**));**

**END** CU\_BF**;**

**ARCHITECTURE** behav **OF** CU\_BF **IS**

**COMPONENT** MUX\_20to1 **IS**

**PORT(** A**,**B**:** **IN** std\_logic\_vector**(**20 **downto** 0**);**

SEL **:** **IN** std\_logic**;**

OUTPUT**:** **OUT** std\_logic\_vector**(**20 **downto** 0**));**

**END** **COMPONENT;**

**COMPONENT** REG\_PS\_BF **IS**

**PORT** **(** CLK**,**RESET**:** **IN** std\_logic**;**

D**:** **IN** std\_logic\_vector**(**4 **downto** 0**);**

Q**:** **OUT** std\_logic\_vector**(**4 **downto** 0**));**

**END** **COMPONENT;**

**COMPONENT** PLA\_BF **IS**

**PORT(** LOAD**,**SEQ**,**LSB**,**CC **:** **IN** std\_logic**;**

LSB\_OUT**:** **OUT** std\_logic**);**

**END** **COMPONENT;**

**COMPONENT** uROM **is**

**generic(** row **:** integer**:=** 10**;**

column **:** integer**:=** 42

**);**

**port(** M**:** **in** std\_logic\_vector **(**3 **downto** 0**);**

OUT\_uROM**:** **out** std\_logic\_vector **(**column**-**1 **downto** 0**)**

**);**

**end** **COMPONENT;**

**SIGNAL** add**,**NEXTS **:**std\_logic\_vector**(**3 **downto** 0**);**

**SIGNAL** AS**,**BS **:**std\_logic\_vector**(**20 **downto** 0**);**

**SIGNAL** CCs**,** LSB\_MP**,**LSB\_OUTS**,**lsb\_sel **:** std\_logic**;**

**BEGIN**

ROM**:** uROM **PORT** **MAP** **(**M**=>** add**,** OUT\_uROM**(**42**-**1 **downto** 42**-**21**)** **=>** AS**,** OUT\_uROM**(**42**-**22 **downto** 0**)=>** BS**);**

PLA**:** PLA\_BF **PORT** **MAP(** LOAD**=>**LOAD**,**SEQ**=>**SEQ**,**CC**=>**CCs**,**LSB**=>**LSB\_MP**,** LSB\_OUT**=>**LSB\_OUTS**);**

REG\_PS**:** REG\_PS\_BF **PORT** **MAP(**RESET**=>**RESET**,**CLK**=>**CLK**,**D**(**0**)=>**LSB\_OUTS**,**D**(**4**downto**1**) =>**NEXTS**,**Q**(**4 **downto** 1**)=>** add**,**Q**(**0**)=>**lsb\_sel**);**

MUX\_20**:**MUX\_20to1 **PORT** **MAP** **(** A**=>**AS**,**B**=>**BS**,** SEL**=>**lsb\_sel**,**OUTPUT**(**20**)=>**CCS**,**

OUTPUT**(**19 **downto** 16**)=>**NEXTS**,**OUTPUT**(**15**)=>**LSB\_MP**,**OUTPUT**(**14 **downto** 0**)=>** CTRL\_OUT**);**

**END** BEHAV**;**

**MUX\_2TO1.VHD**

**LIBRARY** ieee**;**

**USE** ieee**.**std\_logic\_1164**.all;**

**USE** ieee**.**numeric\_std**.all;**

**ENTITY** MUX\_2to1 **IS**

**PORT(** A**,**B**,**SEL **:** **IN** std\_logic**;**

OUTPUT**:** **OUT** std\_logic**);**

**END** MUX\_2to1**;**

**ARCHITECTURE** behav **OF** MUX\_2to1 **IS**

**BEGIN**

**with** SEL **select**

OUTPUT **<=** A **when** '0'**,**

B **when** '1'**,**

'0' **when** **others;**

**end** Behav**;**

**MUX\_20TO1.VHD**

**LIBRARY** ieee**;**

**USE** ieee**.**std\_logic\_1164**.all;**

**USE** ieee**.**numeric\_std**.all;**

**ENTITY** MUX\_20to1 **IS**

**PORT(** A**,**B**:** **IN** std\_logic\_vector**(**20 **downto** 0**);**

SEL **:** **IN** std\_logic**;**

OUTPUT**:** **OUT** std\_logic\_vector**(**20 **downto** 0**));**

**END** MUX\_20to1**;**

**ARCHITECTURE** behav **OF** MUX\_20to1 **IS**

**BEGIN**

**with** SEL **select**

OUTPUT **<=** A **when** '0'**,**

B **when** '1'**,**

"000000000000000000000" **when** **OTHERS;**

**end** Behav**;**

**PLA\_BF.VHD**

**LIBRARY** ieee**;**

**USE** ieee**.**std\_logic\_1164**.all;**

**USE** ieee**.**numeric\_std**.all;**

**ENTITY** PLA\_BF **IS**

**PORT(** LOAD**,**SEQ**,**LSB**,**CC **:** **IN** std\_logic**;**

LSB\_OUT**:** **OUT** std\_logic**);**

**END** PLA\_BF**;**

**ARCHITECTURE** behav **OF** PLA\_BF **IS**

**COMPONENT** MUX\_2to1 **IS**

**PORT(** A**,**B**,**SEL **:** **IN** std\_logic**;**

OUTPUT**:** **OUT** std\_logic**);**

**END** **COMPONENT;**

**SIGNAL** LOGICA **:** std\_logic**;**

**BEGIN**

LOGICA **<=** **(**SEQ **AND** **NOT(**LSB**))** **OR** **(**LOAD **AND** LSB**);**

mux2**:** MUX\_2to1 **PORT** **MAP(** A**=>** LSB**,** B**=>** LOGICA**,** SEL**=>** CC**,** OUTPUT**=>** LSB\_OUT**);**

**END** behav**;**

**REG\_PS\_BF.VHD**

**LIBRARY** ieee**;**

**USE** ieee**.**std\_logic\_1164**.all;**

**USE** ieee**.**numeric\_std**.all;**

**ENTITY** REG\_PS\_BF **IS**

**PORT** **(** CLK**,**RESET**:** **IN** std\_logic**;**

D**:** **IN** std\_logic\_vector**(**4 **downto** 0**);**

Q**:** **OUT** std\_logic\_vector**(**4 **downto** 0**));**

**END** REG\_PS\_BF**;**

**ARCHITECTURE** behav **OF** REG\_PS\_BF **IS**

**BEGIN**

**PROCESS(**CLK**,**RESET**)**

**BEGIN**

**IF** **(**RESET **=** '1'**)** **THEN**

Q**<=**"00000"**;**

**ELSE** **IF** **(**CLK'**EVENT** **AND** CLK **=** '1'**)** **THEN**

Q**<=** D**;**

**END** **IF;**

**END** **IF;**

**END** **PROCESS;**

**END** behav**;**

**uROM.vhd**

**library** ieee**;**

**use** ieee**.**std\_logic\_1164**.all;**

**use** ieee**.**numeric\_std**.all;**

**entity** uROM **is**

**generic(** row **:** integer**:=** 11**;**

column **:** integer**:=** 42

**);**

**port(** M**:** **in** std\_logic\_vector **(**3 **downto** 0**);**

OUT\_uROM**:** **out** std\_logic\_vector **(**column**-**1 **downto** 0**)**

**);**

**end** uROM**;**

**architecture** Behavioral **of** uROM **is**

**type** uROM\_MATRIX **is** **array** **(**0 **to** 10**)** **of** std\_logic\_vector**(**column**-**1 **downto** 0**);**

**signal** uROM\_LINES**:** uROM\_MATRIX**;**

**signal** r11**,**r12**,**r21**,**r22**,**r31**,**r32**,**r41**,**r42**,**r51**,**r52**,**r61**,**r62**,**r71**,**r72**,**r81**,**r82**,**r91**,**r92**,**r101**,**r102**,**r111 **:** std\_logic\_vector **(**20 **downto** 0**);**

**begin**

-- CC(1) | NA(5) | COMAND (15)

r11 **<=** "1000010UUUUUUU0000000"**;** -- IDLE

r12 **<=** "0000100UUUUUUU0000000"**;** -- LOAD\_ST

r21 **<=** "0000110UU1001U0000000"**;** -- M1

r22 **<=** "0001000UU1011U0000000"**;** -- M3

r31 **<=** "000101000111100000000"**;** -- S1\_M2

r32 **<=** "000110001110100000000"**;** -- S3\_M4

r41 **<=** "00011101U00U010000000"**;** -- S2\_M5

r42 **<=** "10100001U01U000000010"**;** -- S4\_M6

r51 **<=** "00101011UUUUU10100000"**;** -- S5

r52 **<=** "00111011UUUUU10100000"**;** -- LOAD\_ST\_S

r61 **<=** "00101111UUUUU10100100"**;** -- S6

r62 **<=** "0011000UUUUUUU0101000"**;** -- ROUND3

r71 **<=** "0011010UUUUUUU0101100"**;** -- ROUND4

r72 **<=** "0101000UUUUUUU1010000"**;** -- LAST\_SAVE

r81 **<=** "00111111U100110100100"**;** -- M1\_S

r82 **<=** "0100000UU1011U0101000"**;** -- M3\_S

r91 **<=** "010001000111100101100"**;** -- S1\_M2\_S

r92 **<=** "010010001110100010000"**;** -- S3\_M4\_S

r101 **<=** "01001101U00U010000001"**;** -- S2\_M5\_S

r102 **<=** "10100001U01U000000010"**;** -- S4\_M6\_S

r111 **<=** "0000000UUUUUUU0000001"**;** -- DONE

uROM\_LINES**(**0**)** **<=** r11 **&** r12**;**

uROM\_LINES**(**1**)** **<=** r21 **&** r22**;**

uROM\_LINES**(**2**)** **<=** r31 **&** r32**;**

uROM\_LINES**(**3**)** **<=** r41 **&** r42**;**

uROM\_LINES**(**4**)** **<=** r51 **&** r52**;**

uROM\_LINES**(**5**)** **<=** r61 **&** r62**;**

uROM\_LINES**(**6**)** **<=** r71 **&** r72**;**

uROM\_LINES**(**7**)** **<=** r81 **&** r82**;**

uROM\_LINES**(**8**)** **<=** r91 **&** r92**;**

uROM\_LINES**(**9**)** **<=** r101 **&** r102**;**

uROM\_LINES**(**10**)** **<=** r111 **&** r111**;**

OUT\_uROM **<=** uROM\_LINES**(to\_integer(**unsigned**(**M**)));**

**end** Behavioral**;**

**CU\_TOP.VHD**

**LIBRARY** ieee**;**

**USE** ieee**.**std\_logic\_1164**.all;**

**USE** ieee**.**numeric\_std**.all;**

**ENTITY** CU\_TOP **IS**

**PORT(** START**,**PROGRESS**,**FREE\_M**,**END\_BF**,**SEQ**,**CLK**,**RESET**:** **IN** std\_logic**;**

CTRL\_TOP\_OUT**:** **OUT** std\_logic\_vector**(**5 **downto** 0**));**

**END** CU\_TOP**;**

**ARCHITECTURE** behav **OF** CU\_TOP **IS**

**COMPONENT** MUX\_12to1\_TOP **IS**

**PORT(** A**,**B**,**C**,**D**:** **IN** std\_logic\_vector**(**11 **downto** 0**);**

SEL **:** **IN** std\_logic\_vector**(**1 **downto** 0**);**

OUTPUT**:** **OUT** std\_logic\_vector**(**11 **downto** 0**));**

**END** **COMPONENT;**

**COMPONENT** REG\_PS\_TOP **IS**

**PORT** **(** CLK**,**RESET**:** **IN** std\_logic**;**

D**:** **IN** std\_logic\_vector**(**3 **downto** 0**);**

Q**:** **OUT** std\_logic\_vector**(**3 **downto** 0**));**

**END** **COMPONENT;**

**COMPONENT** PLA\_TOP **IS**

**PORT(** START**,**PROGRESS**,**FREE\_M**,**END\_BF**,**SEQ**:** **IN** std\_logic**;**

BIT0\_IN**,**BIT1\_IN**:** **IN** std\_logic**;**

CC**:** **IN** std\_logic\_vector**(**1 **downto** 0**);**

BIT0\_OUT**,**BIT1\_OUT**:** **OUT** std\_logic**);**

**END** **COMPONENT;**

**COMPONENT** uROM\_TOP **is**

**generic(** row **:** integer**:=** 3**;**

column **:** integer**:=** 48

**);**

**port(** M**:** **in** std\_logic\_vector **(**1 **downto** 0**);**

OUT\_uROM**:** **out** std\_logic\_vector **(**column**-**1 **downto** 0**)**

**);**

**end** **COMPONENT;**

**SIGNAL** add**,**lsb\_sel**:**std\_logic\_vector**(**1 **downto** 0**);**

**SIGNAL** AS**,**BS**,**CS**,**DS **:**std\_logic\_vector**(**11 **downto** 0**);**

**SIGNAL** CCS**:** STD\_LOGIC\_VECTOR**(**1 **downto** 0**);**

**SIGNAL** BIT3**,**BIT0\_MP**,**BIT1\_MP**,**BIT2**,**BIT0\_OUTS**,**BIT1\_OUTS **:** std\_logic**;**

**BEGIN**

ROM**:** uROM\_TOP **PORT** **MAP** **(**M**=>** add**,**OUT\_uROM**(**47 **downto** 36**)** **=>** AS**,** OUT\_uROM**(**35 **downto** 24**)=>** BS**,**

OUT\_uROM**(**23 **downto** 12**)** **=>** CS**,** OUT\_uROM**(**11 **downto** 0**)=>** DS**);**

PLA**:** PLA\_TOP **PORT** **MAP(** SEQ**=>**SEQ**,**CC**=>**CCs**,**BIT0\_IN**=>**BIT0\_MP**,**BIT1\_IN**=>**BIT1\_MP**,**BIT0\_OUT**=>**BIT0\_OUTS**,**

BIT1\_OUT**=>**BIT1\_OUTS**,**START**=>**START**,**PROGRESS**=>**PROGRESS**,**FREE\_M**=>**FREE\_M**,**

END\_BF**=>**END\_BF**);**

REG\_PS**:** REG\_PS\_TOP **PORT** **MAP(** CLK**=>**CLK**,**RESET**=>**RESET**,**D**(**0**)=>**BIT0\_OUTS**,**D**(**1**)=>**BIT1\_OUTS**,**D**(**2**)=>**BIT2**,**D**(**3**)=>**BIT3**,**

Q**(**3 **downto** 2**)=>** add**,**Q**(**1**)=>**lsb\_sel**(**1**),**Q**(**0**)=>**lsb\_sel**(**0**));**

MUX\_12**:** MUX\_12to1\_TOP **PORT** **MAP** **(** A**=>**AS**,**B**=>**BS**,**C**=>**CS**,**D**=>**DS**,**SEL**=>**lsb\_sel**,**OUTPUT**(**11 **DOWNTO** 10**)=>**CCS**,**

OUTPUT**(**9**)=>**BIT3**,**OUTPUT**(**8**)=>**BIT2**,**OUTPUT**(**7**)=>**BIT1\_MP**,**OUTPUT**(**6**)=>**BIT0\_MP**,**

OUTPUT**(**5 **downto** 0**)=>** CTRL\_TOP\_OUT**);**

**END** BEHAV**;**

**MUX\_12TO1\_TOP.VHD**

**LIBRARY** ieee**;**

**USE** ieee**.**std\_logic\_1164**.all;**

**USE** ieee**.**numeric\_std**.all;**

**ENTITY** MUX\_12to1\_TOP **IS**

**PORT(** A**,**B**,**C**,**D**:** **IN** std\_logic\_vector**(**11 **downto** 0**);**

SEL **:** **IN** std\_logic\_vector**(**1 **downto** 0**);**

OUTPUT**:** **OUT** std\_logic\_vector**(**11 **downto** 0**));**

**END** MUX\_12to1\_TOP**;**

**ARCHITECTURE** behav **OF** MUX\_12to1\_TOP **IS**

**BEGIN**

**with** SEL **select**

OUTPUT **<=** A **when** "00"**,**

B **when** "01"**,**

c **when** "10"**,**

D **when** "11"**,**

"000000000000" **when** **others;**

**end** Behav**;**

**PLA\_TOP.VHD**

**LIBRARY** ieee**;**

**USE** ieee**.**std\_logic\_1164**.all;**

**USE** ieee**.**numeric\_std**.all;**

**ENTITY** PLA\_TOP **IS**

**PORT(** START**,**PROGRESS**,**FREE\_M**,**END\_BF**,**SEQ**:** **IN** std\_logic**;**

BIT0\_IN**,**BIT1\_IN**:** **IN** std\_logic**;**

CC**:** **IN** std\_logic\_vector**(**1 **downto** 0**);**

BIT0\_OUT**,**BIT1\_OUT**:** **OUT** std\_logic**);**

**END** PLA\_TOP**;**

**ARCHITECTURE** behav **OF** PLA\_TOP **IS**

**COMPONENT** MUX\_2to1 **IS**

**PORT(** A**,**B**,**SEL **:** **IN** std\_logic**;**

OUTPUT**:** **OUT** std\_logic**);**

**END** **COMPONENT;**

**SIGNAL** LOGICA\_0**,**LOGICA\_1**,**NEW\_FREE\_M **:** std\_logic**;**

**BEGIN**

NEW\_FREE\_M **<=** FREE\_M **AND** SEQ**;**

LOGICA\_0 **<=** START **OR** **(NOT(**NEW\_FREE\_M**)** **AND** END\_BF**)** **OR** **(NOT(**NEW\_FREE\_M**)** **AND** PROGRESS**)** **;** -- C + A' \* B + A' \* D

LOGICA\_1 **<=** **(NOT(**NEW\_FREE\_M**)** **AND** **NOT(**END\_BF**))** **OR** **(**NEW\_FREE\_M **AND** END\_BF**);** -- A' \* B' + A \* B

mux0**:** MUX\_2to1 **PORT** **MAP(** A**=>** BIT0\_IN**,** B**=>** LOGICA\_0**,** SEL**=>** CC**(**0**),** OUTPUT**=>** BIT0\_OUT**);**

mux1**:** MUX\_2to1 **PORT** **MAP(** A**=>** BIT1\_IN**,** B**=>** LOGICA\_1**,** SEL**=>** CC**(**1**),** OUTPUT**=>** BIT1\_OUT**);**

**END** behav**;**

**REG\_PS\_TOP.VHD**

**LIBRARY** ieee**;**

**USE** ieee**.**std\_logic\_1164**.all;**

**USE** ieee**.**numeric\_std**.all;**

**ENTITY** REG\_PS\_TOP **IS**

**PORT** **(** CLK**,**RESET**:** **IN** std\_logic**;**

D**:** **IN** std\_logic\_vector**(**3 **downto** 0**);**

Q**:** **OUT** std\_logic\_vector**(**3 **downto** 0**));**

**END** REG\_PS\_TOP**;**

**ARCHITECTURE** behav **OF** REG\_PS\_TOP **IS**

**BEGIN**

**PROCESS(**CLK**,**RESET**)**

**BEGIN**

**IF** **(**RESET **=** '1'**)** **THEN**

Q**<=** "0011"**;**

**ELSE** **IF** **(**CLK'**EVENT** **AND** CLK **=** '1'**)** **THEN**

Q**<=** D**;**

**END** **IF;**

**END** **IF;**

**END** **PROCESS;**

**END** behav**;**

**uROM\_TOP.VHD**

**library** ieee**;**

**use** ieee**.**std\_logic\_1164**.all;**

**use** ieee**.**numeric\_std**.all;**

**entity** uROM\_TOP **is**

**generic(** row **:** integer**:=** 3**;**

column **:** integer**:=** 48

**);**

**port(** M**:** **in** std\_logic\_vector **(**1 **downto** 0**);**

OUT\_uROM**:** **out** std\_logic\_vector **(**column**-**1 **downto** 0**)**

**);**

**end** uROM\_TOP**;**

**architecture** Behavioral **of** uROM\_TOP **is**

**type** uROM\_MATRIX **is** **array** **(**0 **to** 2**)** **of** std\_logic\_vector**(**column**-**1 **downto** 0**);**

**signal** uROM\_LINES**:** uROM\_MATRIX**;**

**signal** r0011**,**r0000**,**r0001**,**r0111**,**r0100**,**r0101**,**r0110**,**r1010**,**r1011**,**rnull **:** std\_logic\_vector **(**11 **downto** 0**);**

**begin**

-- CC(1) | NA(4) | COMAND (12)

r0011 **<=** "000000000001"**;** -- RESET

r0000 **<=** "010001000000"**;** -- IDLE

r0001 **<=** "000111110000"**;** -- LOAD

r0111 **<=** "110100001000"**;** -- WORK

r0100 **<=** "000111101000"**;** -- FREE\_M

r0101 **<=** "011010000010"**;** -- READY

r0110 **<=** "000111101010"**;** -- FREE\_M-READY

r1010 **<=** "000011000100"**;** -- DONE

r1011 **<=** "000111000000"**;** -- PREWORK

rnull **<=** "UUUUUUUUUUUU"**;** -- NULL

uROM\_LINES**(**0**)** **<=** r0000 **&** r0001 **&** rnull **&** r0011**;**

uROM\_LINES**(**1**)** **<=** r0100 **&** r0101 **&** r0110 **&** r0111**;**

uROM\_LINES**(**2**)** **<=** rnull **&** rnull **&** r1010 **&** r1011**;**

OUT\_uROM **<=** uROM\_LINES**(to\_integer(**unsigned**(**M**)));**

**end** Behavioral**;**

**MOL.VHD**

**LIBRARY** ieee**;**

**USE** ieee**.**std\_logic\_1164**.all;**

**USE** ieee**.**numeric\_std**.all;**

**ENTITY** MOL **IS**

**PORT** **(** C**,**CLK**:** **IN** std\_logic**;**

ADD1**,**ADD2**:** **IN** SIGNED**(**19 **downto** 0**);**

RESULT**:** **OUT** SIGNED**(**38 **downto** 0**));**

**END** MOL**;**

**ARCHITECTURE** behav **OF** MOL **IS**

**COMPONENT** MOLTI **IS**

**PORT** **(** C**:** **IN** std\_logic**;**

ADD1**,**ADD2**:** **IN** SIGNED**(**19 **downto** 0**);**

RESULT**:** **OUT** SIGNED**(**39 **downto** 0**));**

**END** **COMPONENT;**

**COMPONENT** REG\_MOL **IS**

**PORT** **(** CLK**:** **IN** std\_logic**;**

D**:** **IN** SIGNED**(**38 **downto** 0**);**

Q**:** **OUT** SIGNED**(**38 **downto** 0**));**

**END** **COMPONENT;**

**SIGNAL** partial**:** SIGNED**(**39 **DOWNTO** 0**);**

**BEGIN**

moltiplicatore**:** MOLTI **PORT** **MAP(**C**=>**C**,**ADD1**=>**ADD1**,**ADD2**=>**ADD2**,**RESULT**=>**partial**);**

reg**:** REG\_MOL **PORT** **MAP(** CLK**=>**CLK**,**D**=>**partial**(**38 **DOWNTO** 0**),**Q**=>** RESULT**);**

**END** behav**;**

**MOLTI.VHD**

**LIBRARY** ieee**;**

**USE** ieee**.**std\_logic\_1164**.all;**

**USE** ieee**.**numeric\_std**.all;**

**ENTITY** MOLTI **IS**

**PORT** **(** C**:** **IN** std\_logic**;**

ADD1**,**ADD2**:** **IN** SIGNED**(**19 **downto** 0**);**

RESULT**:** **OUT** SIGNED**(**39 **downto** 0**));**

**END** MOLTI**;**

**ARCHITECTURE** behav **OF** MOLTI **IS**

**SIGNAL** sign\_ext**:** SIGNED**(**19 **DOWNTO** 0**);**

**BEGIN**

sign\_ext **<=(others=>** ADD1**(**0**));**

**PROCESS(**C**,**ADD1**,**ADD2**,**sign\_ext**)**

**BEGIN**

**IF** **(**C**=** '1'**)** **THEN** -- ADD1\*ADD2

RESULT **<=** ADD1 **\*** ADD2**;**

**END** **IF;**

**IF** **(**C **=** '0'**)** **THEN** -- ADD1\*2

RESULT**(**38 **DOWNTO** 20**)** **<=** ADD1**(**18 **DOWNTO** 0**);**

RESULT**(**19 **DOWNTO** 0**)** **<=** sign\_ext**;**

RESULT**(**39**)** **<=** ADD1**(**0**);**

**END** **IF;**

**END** **PROCESS;**

**END** behav**;**

**REG\_MOL.VHD**

**LIBRARY** ieee**;**

**USE** ieee**.**std\_logic\_1164**.all;**

**USE** ieee**.**numeric\_std**.all;**

**ENTITY** REG\_MOL **IS**

**PORT** **(** CLK**:** **IN** std\_logic**;**

D**:** **IN** SIGNED**(**38 **downto** 0**);**

Q**:** **OUT** SIGNED**(**38 **downto** 0**));**

**END** REG\_MOL**;**

**ARCHITECTURE** behav **OF** REG\_MOL **IS**

**BEGIN**

**PROCESS(**CLK**)**

**BEGIN**

**IF** **(**CLK'**EVENT** **AND** CLK **=** '1'**)** **THEN**

Q**<=** D**;**

**END** **IF;**

**END** **PROCESS;**

**END** behav**;**

**OR\_PORT.VHD**

**LIBRARY** ieee**;**

**USE** ieee**.**std\_logic\_1164**.all;**

**USE** ieee**.**numeric\_std**.all;**

**ENTITY** OR\_PORT **IS**

**PORT** **(**

B0AR**,**B1AR**,**B2AR**,**B3AR**,**B4AR**,**B5AR**,**B6AR**,**B7AR**:** **IN** SIGNED**(**19 **downto** 0**);**

B0AI**,**B1AI**,**B2AI**,**B3AI**,**B4AI**,**B5AI**,**B6AI**,**B7AI**:** **IN** SIGNED**(**19 **downto** 0**);**

B0BR**,**B1BR**,**B2BR**,**B3BR**,**B4BR**,**B5BR**,**B6BR**,**B7BR**:** **IN** SIGNED**(**19 **downto** 0**);**

B0BI**,**B1BI**,**B2BI**,**B3BI**,**B4BI**,**B5BI**,**B6BI**,**B7BI**:** **IN** SIGNED**(**19 **downto** 0**);**

START**:** **OUT** std\_logic**);**

**END** OR\_PORT**;**

**ARCHITECTURE** behav **OF** OR\_PORT **IS**

**SIGNAL** B0ARS**,**B1ARS**,**B2ARS**,**B3ARS**,**B4ARS**,**B5ARS**,**B6ARS**,**B7ARS**:** STD\_LOGIC\_VECTOR**(**19 **downto** 0**);**

**SIGNAL** B0AIS**,**B1AIS**,**B2AIS**,**B3AIS**,**B4AIS**,**B5AIS**,**B6AIS**,**B7AIS**:** STD\_LOGIC\_VECTOR**(**19 **downto** 0**);**

**SIGNAL** B0BRS**,**B1BRS**,**B2BRS**,**B3BRS**,**B4BRS**,**B5BRS**,**B6BRS**,**B7BRS**:** STD\_LOGIC\_VECTOR**(**19 **downto** 0**);**

**SIGNAL** B0BIS**,**B1BIS**,**B2BIS**,**B3BIS**,**B4BIS**,**B5BIS**,**B6BIS**,**B7BIS**:** STD\_LOGIC\_VECTOR**(**19 **downto** 0**);**

**SIGNAL** START0**,**START1**,**START2**,**START3**,**START4**,**START5**,**START6**,**START7**,**START8**,**

START9**,**START10**,**START11**,**START12**,**START13**,**START14**,**START15**,**

START16**,**START17**,**START18**,**START19**,**START20**,**START21**,**START22**,**START23**,**

START24**,**START25**,**START26**,**START27**,**START28**,**START29**,**START30**,**START31**:** STD\_LOGIC**;**

**BEGIN**

B0ARS**<=** std\_logic\_vector**(**B0AR**);**

B1ARS**<=** std\_logic\_vector**(**B1AR**);**

B2ARS**<=** std\_logic\_vector**(**B2AR**);**

B3ARS**<=** std\_logic\_vector**(**B3AR**);**

B4ARS**<=** std\_logic\_vector**(**B4AR**);**

B5ARS**<=** std\_logic\_vector**(**B5AR**);**

B6ARS**<=** std\_logic\_vector**(**B6AR**);**

B7ARS**<=** std\_logic\_vector**(**B7AR**);**

B0AIS**<=** std\_logic\_vector**(**B0AI**);**

B1AIS**<=** std\_logic\_vector**(**B1AI**);**

B2AIS**<=** std\_logic\_vector**(**B2AI**);**

B3AIS**<=** std\_logic\_vector**(**B3AI**);**

B4AIS**<=** std\_logic\_vector**(**B4AI**);**

B5AIS**<=** std\_logic\_vector**(**B5AI**);**

B6AIS**<=** std\_logic\_vector**(**B6AI**);**

B7AIS**<=** std\_logic\_vector**(**B7AI**);**

B0BIS**<=** std\_logic\_vector**(**B0BI**);**

B1BIS**<=** std\_logic\_vector**(**B1BI**);**

B2BIS**<=** std\_logic\_vector**(**B2BI**);**

B3BIS**<=** std\_logic\_vector**(**B3BI**);**

B4BIS**<=** std\_logic\_vector**(**B4BI**);**

B5BIS**<=** std\_logic\_vector**(**B5BI**);**

B6BIS**<=** std\_logic\_vector**(**B6BI**);**

B7BIS**<=** std\_logic\_vector**(**B7BI**);**

B0BRS**<=** std\_logic\_vector**(**B0BR**);**

B1BRS**<=** std\_logic\_vector**(**B1BR**);**

B2BRS**<=** std\_logic\_vector**(**B2BR**);**

B3BRS**<=** std\_logic\_vector**(**B3BR**);**

B4BRS**<=** std\_logic\_vector**(**B4BR**);**

B5BRS**<=** std\_logic\_vector**(**B5BR**);**

B6BRS**<=** std\_logic\_vector**(**B6BR**);**

B7BRS**<=** std\_logic\_vector**(**B7BR**);**

START0**<=** B0ARS**(**19**)** **OR** B0ARS**(**18**)** **OR** B0ARS**(**17**)** **OR** B0ARS**(**16**)** **OR** B0ARS**(**15**)** **OR** B0ARS**(**14**)** **OR** B0ARS**(**13**)** **OR** B0ARS**(**12**)** **OR** B0ARS**(**11**)** **OR** B0ARS**(**10**)** **OR** B0ARS**(**9**)** **OR** B0ARS**(**8**)** **OR** B0ARS**(**7**)** **OR** B0ARS**(**6**)** **OR** B0ARS**(**5**)** **OR** B0ARS**(**4**)** **OR** B0ARS**(**3**)** **OR** B0ARS**(**2**)** **OR** B0ARS**(**1**)** **OR** B0ARS**(**0**);**

START1**<=** B1ARS**(**19**)** **OR** B1ARS**(**18**)** **OR** B1ARS**(**17**)** **OR** B1ARS**(**16**)** **OR** B1ARS**(**15**)** **OR** B1ARS**(**14**)** **OR** B1ARS**(**13**)** **OR** B1ARS**(**12**)** **OR** B1ARS**(**11**)** **OR** B1ARS**(**10**)** **OR** B1ARS**(**9**)** **OR** B1ARS**(**8**)** **OR** B1ARS**(**7**)** **OR** B1ARS**(**6**)** **OR** B1ARS**(**5**)** **OR** B1ARS**(**4**)** **OR** B1ARS**(**3**)** **OR** B1ARS**(**2**)** **OR** B1ARS**(**1**)** **OR** B1ARS**(**0**);**

START2**<=** B2ARS**(**19**)** **OR** B2ARS**(**18**)** **OR** B2ARS**(**17**)** **OR** B2ARS**(**16**)** **OR** B2ARS**(**15**)** **OR** B2ARS**(**14**)** **OR** B2ARS**(**13**)** **OR** B2ARS**(**12**)** **OR** B2ARS**(**11**)** **OR** B2ARS**(**10**)** **OR** B2ARS**(**9**)** **OR** B2ARS**(**8**)** **OR** B2ARS**(**7**)** **OR** B2ARS**(**6**)** **OR** B2ARS**(**5**)** **OR** B2ARS**(**4**)** **OR** B2ARS**(**3**)** **OR** B2ARS**(**2**)** **OR** B2ARS**(**1**)** **OR** B2ARS**(**0**);**

START3**<=** B3ARS**(**19**)** **OR** B3ARS**(**18**)** **OR** B3ARS**(**17**)** **OR** B3ARS**(**16**)** **OR** B3ARS**(**15**)** **OR** B3ARS**(**14**)** **OR** B3ARS**(**13**)** **OR** B3ARS**(**12**)** **OR** B3ARS**(**11**)** **OR** B3ARS**(**10**)** **OR** B3ARS**(**9**)** **OR** B3ARS**(**8**)** **OR** B3ARS**(**7**)** **OR** B3ARS**(**6**)** **OR** B3ARS**(**5**)** **OR** B3ARS**(**4**)** **OR** B3ARS**(**3**)** **OR** B3ARS**(**2**)** **OR** B3ARS**(**1**)** **OR** B3ARS**(**0**);**

START4**<=** B4ARS**(**19**)** **OR** B4ARS**(**18**)** **OR** B4ARS**(**17**)** **OR** B4ARS**(**16**)** **OR** B4ARS**(**15**)** **OR** B4ARS**(**14**)** **OR** B4ARS**(**13**)** **OR** B4ARS**(**12**)** **OR** B4ARS**(**11**)** **OR** B4ARS**(**10**)** **OR** B4ARS**(**9**)** **OR** B4ARS**(**8**)** **OR** B4ARS**(**7**)** **OR** B4ARS**(**6**)** **OR** B4ARS**(**5**)** **OR** B4ARS**(**4**)** **OR** B4ARS**(**3**)** **OR** B4ARS**(**2**)** **OR** B4ARS**(**1**)** **OR** B4ARS**(**0**);**

START5**<=** B5ARS**(**19**)** **OR** B5ARS**(**18**)** **OR** B5ARS**(**17**)** **OR** B5ARS**(**16**)** **OR** B5ARS**(**15**)** **OR** B5ARS**(**14**)** **OR** B5ARS**(**13**)** **OR** B5ARS**(**12**)** **OR** B5ARS**(**11**)** **OR** B5ARS**(**10**)** **OR** B5ARS**(**9**)** **OR** B5ARS**(**8**)** **OR** B5ARS**(**7**)** **OR** B5ARS**(**6**)** **OR** B5ARS**(**5**)** **OR** B5ARS**(**4**)** **OR** B5ARS**(**3**)** **OR** B5ARS**(**2**)** **OR** B5ARS**(**1**)** **OR** B5ARS**(**0**);**

START6**<=** B6ARS**(**19**)** **OR** B6ARS**(**18**)** **OR** B6ARS**(**17**)** **OR** B6ARS**(**16**)** **OR** B6ARS**(**15**)** **OR** B6ARS**(**14**)** **OR** B6ARS**(**13**)** **OR** B6ARS**(**12**)** **OR** B6ARS**(**11**)** **OR** B6ARS**(**10**)** **OR** B6ARS**(**9**)** **OR** B6ARS**(**8**)** **OR** B6ARS**(**7**)** **OR** B6ARS**(**6**)** **OR** B6ARS**(**5**)** **OR** B6ARS**(**4**)** **OR** B6ARS**(**3**)** **OR** B6ARS**(**2**)** **OR** B6ARS**(**1**)** **OR** B6ARS**(**0**);**

START7**<=** B7ARS**(**19**)** **OR** B7ARS**(**18**)** **OR** B7ARS**(**17**)** **OR** B7ARS**(**16**)** **OR** B7ARS**(**15**)** **OR** B7ARS**(**14**)** **OR** B7ARS**(**13**)** **OR** B7ARS**(**12**)** **OR** B7ARS**(**11**)** **OR** B7ARS**(**10**)** **OR** B7ARS**(**9**)** **OR** B7ARS**(**8**)** **OR** B7ARS**(**7**)** **OR** B7ARS**(**6**)** **OR** B7ARS**(**5**)** **OR** B7ARS**(**4**)** **OR** B7ARS**(**3**)** **OR** B7ARS**(**2**)** **OR** B7ARS**(**1**)** **OR** B7ARS**(**0**);**

--AIS

START8**<=** B0AIS**(**19**)** **OR** B0AIS**(**18**)** **OR** B0AIS**(**17**)** **OR** B0AIS**(**16**)** **OR** B0AIS**(**15**)** **OR** B0AIS**(**14**)** **OR** B0AIS**(**13**)** **OR** B0AIS**(**12**)** **OR** B0AIS**(**11**)** **OR** B0AIS**(**10**)** **OR** B0AIS**(**9**)** **OR** B0AIS**(**8**)** **OR** B0AIS**(**7**)** **OR** B0AIS**(**6**)** **OR** B0AIS**(**5**)** **OR** B0AIS**(**4**)** **OR** B0AIS**(**3**)** **OR** B0AIS**(**2**)** **OR** B0AIS**(**1**)** **OR** B0AIS**(**0**);**

START9**<=** B1AIS**(**19**)** **OR** B1AIS**(**18**)** **OR** B1AIS**(**17**)** **OR** B1AIS**(**16**)** **OR** B1AIS**(**15**)** **OR** B1AIS**(**14**)** **OR** B1AIS**(**13**)** **OR** B1AIS**(**12**)** **OR** B1AIS**(**11**)** **OR** B1AIS**(**10**)** **OR** B1AIS**(**9**)** **OR** B1AIS**(**8**)** **OR** B1AIS**(**7**)** **OR** B1AIS**(**6**)** **OR** B1AIS**(**5**)** **OR** B1AIS**(**4**)** **OR** B1AIS**(**3**)** **OR** B1AIS**(**2**)** **OR** B1AIS**(**1**)** **OR** B1AIS**(**0**);**

START10 **<=** B2AIS**(**19**)** **OR** B2AIS**(**18**)** **OR** B2AIS**(**17**)** **OR** B2AIS**(**16**)** **OR** B2AIS**(**15**)** **OR** B2AIS**(**14**)** **OR** B2AIS**(**13**)** **OR** B2AIS**(**12**)** **OR** B2AIS**(**11**)** **OR** B2AIS**(**11**)** **OR** B2AIS**(**9**)** **OR** B2AIS**(**8**)** **OR** B2AIS**(**7**)** **OR** B2AIS**(**6**)** **OR** B2AIS**(**5**)** **OR** B2AIS**(**4**)** **OR** B2AIS**(**3**)** **OR** B2AIS**(**2**)** **OR** B2AIS**(**1**)** **OR** B2AIS**(**0**);**

START11**<=** B3AIS**(**19**)** **OR** B3AIS**(**18**)** **OR** B3AIS**(**17**)** **OR** B3AIS**(**16**)** **OR** B3AIS**(**15**)** **OR** B3AIS**(**14**)** **OR** B3AIS**(**13**)** **OR** B3AIS**(**12**)** **OR** B3AIS**(**11**)** **OR** B3AIS**(**10**)** **OR** B3AIS**(**9**)** **OR** B3AIS**(**8**)** **OR** B3AIS**(**7**)** **OR** B3AIS**(**6**)** **OR** B3AIS**(**5**)** **OR** B3AIS**(**4**)** **OR** B3AIS**(**3**)** **OR** B3AIS**(**2**)** **OR** B3AIS**(**1**)** **OR** B3AIS**(**0**);**

START12**<=** B4AIS**(**19**)** **OR** B4AIS**(**18**)** **OR** B4AIS**(**17**)** **OR** B4AIS**(**16**)** **OR** B4AIS**(**15**)** **OR** B4AIS**(**14**)** **OR** B4AIS**(**13**)** **OR** B4AIS**(**12**)** **OR** B4AIS**(**11**)** **OR** B4AIS**(**10**)** **OR** B4AIS**(**9**)** **OR** B4AIS**(**8**)** **OR** B4AIS**(**7**)** **OR** B4AIS**(**6**)** **OR** B4AIS**(**5**)** **OR** B4AIS**(**4**)** **OR** B4AIS**(**3**)** **OR** B4AIS**(**2**)** **OR** B4AIS**(**1**)** **OR** B4AIS**(**0**);**

START13**<=** B5AIS**(**19**)** **OR** B5AIS**(**18**)** **OR** B5AIS**(**17**)** **OR** B5AIS**(**16**)** **OR** B5AIS**(**15**)** **OR** B5AIS**(**14**)** **OR** B5AIS**(**13**)** **OR** B5AIS**(**12**)** **OR** B5AIS**(**11**)** **OR** B5AIS**(**10**)** **OR** B5AIS**(**9**)** **OR** B5AIS**(**8**)** **OR** B5AIS**(**7**)** **OR** B5AIS**(**6**)** **OR** B5AIS**(**5**)** **OR** B5AIS**(**4**)** **OR** B5AIS**(**3**)** **OR** B5AIS**(**2**)** **OR** B5AIS**(**1**)** **OR** B5AIS**(**0**);**

START14**<=** B6AIS**(**19**)** **OR** B6AIS**(**18**)** **OR** B6AIS**(**17**)** **OR** B6AIS**(**16**)** **OR** B6AIS**(**15**)** **OR** B6AIS**(**14**)** **OR** B6AIS**(**13**)** **OR** B6AIS**(**12**)** **OR** B6AIS**(**11**)** **OR** B6AIS**(**10**)** **OR** B6AIS**(**9**)** **OR** B6AIS**(**8**)** **OR** B6AIS**(**7**)** **OR** B6AIS**(**6**)** **OR** B6AIS**(**5**)** **OR** B6AIS**(**4**)** **OR** B6AIS**(**3**)** **OR** B6AIS**(**2**)** **OR** B6AIS**(**1**)** **OR** B6AIS**(**0**);**

START15 **<=** B7AIS**(**19**)** **OR** B7AIS**(**18**)** **OR** B7AIS**(**17**)** **OR** B7AIS**(**16**)** **OR** B7AIS**(**15**)** **OR** B7AIS**(**14**)** **OR** B7AIS**(**13**)** **OR** B7AIS**(**12**)** **OR** B7AIS**(**11**)** **OR** B7AIS**(**10**)** **OR** B7AIS**(**9**)** **OR** B7AIS**(**8**)** **OR** B7AIS**(**7**)** **OR** B7AIS**(**6**)** **OR** B7AIS**(**5**)** **OR** B7AIS**(**4**)** **OR** B7AIS**(**3**)** **OR** B7AIS**(**2**)** **OR** B7AIS**(**1**)** **OR** B7AIS**(**0**);**

-- BIS

START16 **<=** B0BIS**(**19**)** **OR** B0BIS**(**18**)** **OR** B0BIS**(**17**)** **OR** B0BIS**(**16**)** **OR** B0BIS**(**15**)** **OR** B0BIS**(**14**)** **OR** B0BIS**(**13**)** **OR** B0BIS**(**12**)** **OR** B0BIS**(**11**)** **OR** B0BIS**(**10**)** **OR** B0BIS**(**9**)** **OR** B0BIS**(**8**)** **OR** B0BIS**(**7**)** **OR** B0BIS**(**6**)** **OR** B0BIS**(**5**)** **OR** B0BIS**(**4**)** **OR** B0BIS**(**3**)** **OR** B0BIS**(**2**)** **OR** B0BIS**(**1**)** **OR** B0BIS**(**0**);**

START17**<=** B1BIS**(**19**)** **OR** B1BIS**(**18**)** **OR** B1BIS**(**17**)** **OR** B1BIS**(**16**)** **OR** B1BIS**(**15**)** **OR** B1BIS**(**14**)** **OR** B1BIS**(**13**)** **OR** B1BIS**(**12**)** **OR** B1BIS**(**11**)** **OR** B1BIS**(**10**)** **OR** B1BIS**(**9**)** **OR** B1BIS**(**8**)** **OR** B1BIS**(**7**)** **OR** B1BIS**(**6**)** **OR** B1BIS**(**5**)** **OR** B1BIS**(**4**)** **OR** B1BIS**(**3**)** **OR** B1BIS**(**2**)** **OR** B1BIS**(**1**)** **OR** B1BIS**(**0**);**

START18**<=** B2BIS**(**19**)** **OR** B2BIS**(**18**)** **OR** B2BIS**(**17**)** **OR** B2BIS**(**16**)** **OR** B2BIS**(**15**)** **OR** B2BIS**(**14**)** **OR** B2BIS**(**13**)** **OR** B2BIS**(**12**)** **OR** B2BIS**(**11**)** **OR** B2BIS**(**10**)** **OR** B2BIS**(**9**)** **OR** B2BIS**(**8**)** **OR** B2BIS**(**7**)** **OR** B2BIS**(**6**)** **OR** B2BIS**(**5**)** **OR** B2BIS**(**4**)** **OR** B2BIS**(**3**)** **OR** B2BIS**(**2**)** **OR** B2BIS**(**1**)** **OR** B2BIS**(**0**);**

START19**<=** B3BIS**(**19**)** **OR** B3BIS**(**18**)** **OR** B3BIS**(**17**)** **OR** B3BIS**(**16**)** **OR** B3BIS**(**15**)** **OR** B3BIS**(**14**)** **OR** B3BIS**(**13**)** **OR** B3BIS**(**12**)** **OR** B3BIS**(**11**)** **OR** B3BIS**(**10**)** **OR** B3BIS**(**9**)** **OR** B3BIS**(**8**)** **OR** B3BIS**(**7**)** **OR** B3BIS**(**6**)** **OR** B3BIS**(**5**)** **OR** B3BIS**(**4**)** **OR** B3BIS**(**3**)** **OR** B3BIS**(**2**)** **OR** B3BIS**(**1**)** **OR** B3BIS**(**0**);**

START20**<=** B4BIS**(**19**)** **OR** B4BIS**(**18**)** **OR** B4BIS**(**17**)** **OR** B4BIS**(**16**)** **OR** B4BIS**(**15**)** **OR** B4BIS**(**14**)** **OR** B4BIS**(**13**)** **OR** B4BIS**(**12**)** **OR** B4BIS**(**11**)** **OR** B4BIS**(**10**)** **OR** B4BIS**(**9**)** **OR** B4BIS**(**8**)** **OR** B4BIS**(**7**)** **OR** B4BIS**(**6**)** **OR** B4BIS**(**5**)** **OR** B4BIS**(**4**)** **OR** B4BIS**(**3**)** **OR** B4BIS**(**2**)** **OR** B4BIS**(**1**)** **OR** B4BIS**(**0**);**

START21**<=** B5BIS**(**19**)** **OR** B5BIS**(**18**)** **OR** B5BIS**(**17**)** **OR** B5BIS**(**16**)** **OR** B5BIS**(**15**)** **OR** B5BIS**(**14**)** **OR** B5BIS**(**13**)** **OR** B5BIS**(**12**)** **OR** B5BIS**(**11**)** **OR** B5BIS**(**10**)** **OR** B5BIS**(**9**)** **OR** B5BIS**(**8**)** **OR** B5BIS**(**7**)** **OR** B5BIS**(**6**)** **OR** B5BIS**(**5**)** **OR** B5BIS**(**4**)** **OR** B5BIS**(**3**)** **OR** B5BIS**(**2**)** **OR** B5BIS**(**1**)** **OR** B5BIS**(**0**);**

START22**<=** B6BIS**(**19**)** **OR** B6BIS**(**18**)** **OR** B6BIS**(**17**)** **OR** B6BIS**(**16**)** **OR** B6BIS**(**15**)** **OR** B6BIS**(**14**)** **OR** B6BIS**(**13**)** **OR** B6BIS**(**12**)** **OR** B6BIS**(**11**)** **OR** B6BIS**(**10**)** **OR** B6BIS**(**9**)** **OR** B6BIS**(**8**)** **OR** B6BIS**(**7**)** **OR** B6BIS**(**6**)** **OR** B6BIS**(**5**)** **OR** B6BIS**(**4**)** **OR** B6BIS**(**3**)** **OR** B6BIS**(**2**)** **OR** B6BIS**(**1**)** **OR** B6BIS**(**0**);**

START23**<=** B7BIS**(**19**)** **OR** B7BIS**(**18**)** **OR** B7BIS**(**17**)** **OR** B7BIS**(**16**)** **OR** B7BIS**(**15**)** **OR** B7BIS**(**14**)** **OR** B7BIS**(**13**)** **OR** B7BIS**(**12**)** **OR** B7BIS**(**11**)** **OR** B7BIS**(**10**)** **OR** B7BIS**(**9**)** **OR** B7BIS**(**8**)** **OR** B7BIS**(**7**)** **OR** B7BIS**(**6**)** **OR** B7BIS**(**5**)** **OR** B7BIS**(**4**)** **OR** B7BIS**(**3**)** **OR** B7BIS**(**2**)** **OR** B7BIS**(**1**)** **OR** B7BIS**(**0**);**

--BRS

START24**<=** B0BRS**(**19**)** **OR** B0BRS**(**18**)** **OR** B0BRS**(**17**)** **OR** B0BRS**(**16**)** **OR** B0BRS**(**15**)** **OR** B0BRS**(**14**)** **OR** B0BRS**(**13**)** **OR** B0BRS**(**12**)** **OR** B0BRS**(**11**)** **OR** B0BRS**(**10**)** **OR** B0BRS**(**9**)** **OR** B0BRS**(**8**)** **OR** B0BRS**(**7**)** **OR** B0BRS**(**6**)** **OR** B0BRS**(**5**)** **OR** B0BRS**(**4**)** **OR** B0BRS**(**3**)** **OR** B0BRS**(**2**)** **OR** B0BRS**(**1**)** **OR** B0BRS**(**0**);**

START25**<=** B1BRS**(**19**)** **OR** B1BRS**(**18**)** **OR** B1BRS**(**17**)** **OR** B1BRS**(**16**)** **OR** B1BRS**(**15**)** **OR** B1BRS**(**14**)** **OR** B1BRS**(**13**)** **OR** B1BRS**(**12**)** **OR** B1BRS**(**11**)** **OR** B1BRS**(**10**)** **OR** B1BRS**(**9**)** **OR** B1BRS**(**8**)** **OR** B1BRS**(**7**)** **OR** B1BRS**(**6**)** **OR** B1BRS**(**5**)** **OR** B1BRS**(**4**)** **OR** B1BRS**(**3**)** **OR** B1BRS**(**2**)** **OR** B1BRS**(**1**)** **OR** B1BRS**(**0**);**

START26**<=** B2BRS**(**19**)** **OR** B2BRS**(**18**)** **OR** B2BRS**(**17**)** **OR** B2BRS**(**16**)** **OR** B2BRS**(**15**)** **OR** B2BRS**(**14**)** **OR** B2BRS**(**13**)** **OR** B2BRS**(**12**)** **OR** B2BRS**(**11**)** **OR** B2BRS**(**10**)** **OR** B2BRS**(**9**)** **OR** B2BRS**(**8**)** **OR** B2BRS**(**7**)** **OR** B2BRS**(**6**)** **OR** B2BRS**(**5**)** **OR** B2BRS**(**4**)** **OR** B2BRS**(**3**)** **OR** B2BRS**(**2**)** **OR** B2BRS**(**1**)** **OR** B2BRS**(**0**);**

START27**<=** B3BRS**(**19**)** **OR** B3BRS**(**18**)** **OR** B3BRS**(**17**)** **OR** B3BRS**(**16**)** **OR** B3BRS**(**15**)** **OR** B3BRS**(**14**)** **OR** B3BRS**(**13**)** **OR** B3BRS**(**12**)** **OR** B3BRS**(**11**)** **OR** B3BRS**(**10**)** **OR** B3BRS**(**9**)** **OR** B3BRS**(**8**)** **OR** B3BRS**(**7**)** **OR** B3BRS**(**6**)** **OR** B3BRS**(**5**)** **OR** B3BRS**(**4**)** **OR** B3BRS**(**3**)** **OR** B3BRS**(**2**)** **OR** B3BRS**(**1**)** **OR** B3BRS**(**0**);**

START28**<=** B4BRS**(**19**)** **OR** B4BRS**(**18**)** **OR** B4BRS**(**17**)** **OR** B4BRS**(**16**)** **OR** B4BRS**(**15**)** **OR** B4BRS**(**14**)** **OR** B4BRS**(**13**)** **OR** B4BRS**(**12**)** **OR** B4BRS**(**11**)** **OR** B4BRS**(**10**)** **OR** B4BRS**(**9**)** **OR** B4BRS**(**8**)** **OR** B4BRS**(**7**)** **OR** B4BRS**(**6**)** **OR** B4BRS**(**5**)** **OR** B4BRS**(**4**)** **OR** B4BRS**(**3**)** **OR** B4BRS**(**2**)** **OR** B4BRS**(**1**)** **OR** B4BRS**(**0**);**

START29**<=** B5BRS**(**19**)** **OR** B5BRS**(**18**)** **OR** B5BRS**(**17**)** **OR** B5BRS**(**16**)** **OR** B5BRS**(**15**)** **OR** B5BRS**(**14**)** **OR** B5BRS**(**13**)** **OR** B5BRS**(**12**)** **OR** B5BRS**(**11**)** **OR** B5BRS**(**10**)** **OR** B5BRS**(**9**)** **OR** B5BRS**(**8**)** **OR** B5BRS**(**7**)** **OR** B5BRS**(**6**)** **OR** B5BRS**(**5**)** **OR** B5BRS**(**4**)** **OR** B5BRS**(**3**)** **OR** B5BRS**(**2**)** **OR** B5BRS**(**1**)** **OR** B5BRS**(**0**);**

START30**<=** B6BRS**(**19**)** **OR** B6BRS**(**18**)** **OR** B6BRS**(**17**)** **OR** B6BRS**(**16**)** **OR** B6BRS**(**15**)** **OR** B6BRS**(**14**)** **OR** B6BRS**(**13**)** **OR** B6BRS**(**12**)** **OR** B6BRS**(**11**)** **OR** B6BRS**(**10**)** **OR** B6BRS**(**9**)** **OR** B6BRS**(**8**)** **OR** B6BRS**(**7**)** **OR** B6BRS**(**6**)** **OR** B6BRS**(**5**)** **OR** B6BRS**(**4**)** **OR** B6BRS**(**3**)** **OR** B6BRS**(**2**)** **OR** B6BRS**(**1**)** **OR** B6BRS**(**0**);**

START31**<=** B7BRS**(**19**)** **OR** B7BRS**(**18**)** **OR** B7BRS**(**17**)** **OR** B7BRS**(**16**)** **OR** B7BRS**(**15**)** **OR** B7BRS**(**14**)** **OR** B7BRS**(**13**)** **OR** B7BRS**(**12**)** **OR** B7BRS**(**11**)** **OR** B7BRS**(**10**)** **OR** B7BRS**(**9**)** **OR** B7BRS**(**8**)** **OR** B7BRS**(**7**)** **OR** B7BRS**(**6**)** **OR** B7BRS**(**5**)** **OR** B7BRS**(**4**)** **OR** B7BRS**(**3**)** **OR** B7BRS**(**2**)** **OR** B7BRS**(**1**)** **OR** B7BRS**(**0**);**

START**<=** START0 **OR** START1 **OR** START2 **OR** START3 **OR** START4 **OR** START5 **OR** START6 **OR** START7 **OR** START8 **OR**

START9 **OR** START10 **OR** START11 **OR** START12 **OR** START13 **OR** START14 **OR** START15 **OR**

START16 **OR** START17 **OR** START18 **OR** START19 **OR** START20 **OR** START21 **OR** START22 **OR** START23 **OR**

START24 **OR** START25 **OR** START26 **OR** START27 **OR** START28 **OR** START29 **OR** START30 **OR** START31**;**

**end** behav**;**

**FLIPFLOP\_STATUS.VHD**

**LIBRARY** ieee**;**

**USE** ieee**.**std\_logic\_1164**.all;**

**USE** ieee**.**numeric\_std**.all;**

**ENTITY** FlipFlop\_status **IS**

**PORT** **(**D**,** CLK**,**set**,**enable**,**RESET **:** **IN** std\_logic**;**

Q**:** **OUT** std\_logic**);**

**END** FlipFlop\_status**;**

**ARCHITECTURE** behav **OF** FlipFlop\_status **IS**

**signal** new\_d **:** std\_logic**;**

**BEGIN**

new\_d**<=** **NOT(**D**);**

**PROCESS(**CLK**,**set**,**enable**,**RESET**)**

**BEGIN**

**IF** **(**CLK'**EVENT** **AND** CLK **=** '1'**)** **THEN**

**IF** **(**enable **=** '1'**)** **THEN**

Q**<=** new\_d**;**

**ELSE** **IF** **(**set**=**'1'**)** **then**

Q**<=**'1'**;**

**ELSE** **IF** **(**RESET **=** '1'**)** **THEN**

Q **<=** '0'**;**

**END** **IF;**

**END** **IF;**

**END** **IF;**

**END** **IF;**

**END** **PROCESS;**

**END** behav**;**

**REG\_STATUS.VHD**

**LIBRARY** ieee**;**

**USE** ieee**.**std\_logic\_1164**.all;**

**USE** ieee**.**numeric\_std**.all;**

**ENTITY** REG\_STATUS **IS**

**PORT** **(** CLK**,**set**,**enable0**,**enable1**,**enable2**,**enable3**,**RESET **:** **IN** std\_logic**;**

Q\_OUT**:** **OUT** std\_logic\_vector**(**3 **downto** 0**));**

**END** REG\_STATUS**;**

**ARCHITECTURE** BEHAV **OF** REG\_STATUS **IS**

**COMPONENT** FlipFlop\_status **IS**

**PORT** **(**D**,** CLK**,**set**,**enable**,**RESET **:** **IN** std\_logic**;**

Q**:** **OUT** std\_logic**);**

**END** **COMPONENT;**

**BEGIN**

ff0**:** FlipFlop\_status **PORT** **MAP(**CLK**=>**CLK**,**set**=>**set**,**enable**=>**enable0**,**Q**=>**Q\_OUT**(**0**),**D**=>** enable0**,**RESET**=>**RESET**);**

ff1**:** FlipFlop\_status **PORT** **MAP(**CLK**=>**CLK**,**set**=>**set**,**enable**=>**enable1**,**Q**=>**Q\_OUT**(**1**),**D**=>** enable1**,**RESET**=>**RESET**);**

ff2**:** FlipFlop\_status **PORT** **MAP(**CLK**=>**CLK**,**set**=>**set**,**enable**=>**enable2**,**Q**=>**Q\_OUT**(**2**),**D**=>** enable2**,**RESET**=>**RESET**);**

ff3**:** FlipFlop\_status **PORT** **MAP(**CLK**=>**CLK**,**set**=>**set**,**enable**=>**enable3**,**Q**=>**Q\_OUT**(**3**),**D**=>** enable3**,**RESET**=>**RESET**);**

**end** behav**;**

**MUX\_4\_RF.VHD**

**LIBRARY** ieee**;**

**USE** ieee**.**std\_logic\_1164**.all;**

**USE** ieee**.**numeric\_std**.all;**

**ENTITY** MUX\_4\_RF **IS**

**PORT(** AR**,**AI**,**BR**,**BI**:** **IN** SIGNED**(**19 **downto** 0**);**

SEL **:** **IN** std\_logic\_vector**(**1 **downto** 0**);**

OUTPUT**:** **OUT** SIGNED**(**19 **downto** 0**));**

**END** MUX\_4\_RF**;**

**ARCHITECTURE** behav **OF** MUX\_4\_RF **IS**

**BEGIN**

**with** SEL **select**

OUTPUT **<=** AR **when** "00"**,**

AI **when** "01"**,**

BR **when** "10"**,**

BI **when** "11"**,**

"00000000000000000000" **when** **others;**

**end** Behav**;**

**MUX\_20TO1\_RF.VHD**

**LIBRARY** ieee**;**

**USE** ieee**.**std\_logic\_1164**.all;**

**USE** ieee**.**numeric\_std**.all;**

**ENTITY** MUX\_20to1\_RF **IS**

**PORT(** A**,**B**:** **IN** SIGNED**(**19 **downto** 0**);**

SEL **:** **IN** std\_logic**;**

OUTPUT**:** **OUT** SIGNED**(**19 **downto** 0**));**

**END** MUX\_20to1\_RF**;**

**ARCHITECTURE** behav **OF** MUX\_20to1\_RF **IS**

**BEGIN**

**with** SEL **select**

OUTPUT **<=** A **when** '0'**,**

B **when** '1'**,**

"00000000000000000000" **when** **others;**

**end** Behav**;**

**REG\_BF\_AB.VHD**

**LIBRARY** ieee**;**

**USE** ieee**.**std\_logic\_1164**.all;**

**USE** ieee**.**numeric\_std**.all;**

**ENTITY** REG\_BF\_AB **IS**

**PORT** **(** CLK**,** ENABLE**:** **IN** std\_logic**;**

D**:** **IN** SIGNED**(**19 **downto** 0**);** -- va cambiato il formato e conversioni

Q**:** **OUT** SIGNED**(**19 **downto** 0**));**

**END** REG\_BF\_AB**;**

**ARCHITECTURE** behav **OF** REG\_BF\_AB **IS**

**BEGIN**

**PROCESS(**CLK**)**

**BEGIN**

**IF** **(**CLK'**EVENT** **AND** CLK **=** '1'**)** **THEN**

**IF** **(** ENABLE **=** '1'**)** **THEN**

Q**<=** D**;**

**END** **IF;**

**END** **IF;**

**END** **PROCESS;**

**END** behav**;**

**REG\_BF\_W.VHD**

**LIBRARY** ieee**;**

**USE** ieee**.**std\_logic\_1164**.all;**

**USE** ieee**.**numeric\_std**.all;**

**ENTITY** REG\_BF\_W **IS**

**PORT** **(** CLK**:** **IN** std\_logic**;**

D**:** **IN** SIGNED**(**19 **downto** 0**);** -- va cambiato il formato e conversioni

Q**:** **OUT** SIGNED**(**19 **downto** 0**));**

**END** REG\_BF\_W**;**

**ARCHITECTURE** behav **OF** REG\_BF\_W **IS**

**BEGIN**

**PROCESS(**CLK**)**

**BEGIN**

**IF** **(**CLK'**EVENT** **AND** CLK **=** '1'**)** **THEN**

Q**<=** D**;**

**END** **IF;**

**END** **PROCESS;**

**END** behav**;**

**RF.VHD**

**LIBRARY** ieee**;**

**USE** ieee**.**std\_logic\_1164**.all;**

**USE** ieee**.**numeric\_std**.all;**

**ENTITY** RF **IS**

**PORT** **(** CLK**,** ENABLE**:** **IN** std\_logic**;**

AI**,**AR**,**BI**,**BR**,**WR**,**WI**:** **IN** SIGNED**(**19 **downto** 0**);** -- va cambiato il formato e conversioni

SEL3**,**SEL1 **:** **IN** std\_logic**;**

SEL2 **:** **IN** std\_logic\_vector**(**1 **downto** 0**);**

OUT\_MUX1**,**OUT\_MUX2**,**OUT\_MUX3**,**AI\_CTRL**,**AR\_CTRL**,**BI\_CTRL**,**BR\_CTRL**:** **OUT** SIGNED**(**19 **downto** 0**));**

**END** RF**;**

**ARCHITECTURE** BEHAV **OF** RF **IS**

**COMPONENT** REG\_BF\_W **IS**

**PORT** **(** CLK**:** **IN** std\_logic**;**

D**:** **IN** SIGNED**(**19 **downto** 0**);** -- va cambiato il formato e conversioni

Q**:** **OUT** SIGNED**(**19 **downto** 0**));**

**END** **COMPONENT;**

**COMPONENT** REG\_BF\_AB **IS**

**PORT** **(** CLK**,** ENABLE**:** **IN** std\_logic**;**

D**:** **IN** SIGNED**(**19 **downto** 0**);** -- va cambiato il formato e conversioni

Q**:** **OUT** SIGNED**(**19 **downto** 0**));**

**END** **COMPONENT;**

**COMPONENT** MUX\_20to1\_RF **IS**

**PORT(** A**,**B**:** **IN** SIGNED**(**19 **downto** 0**);**

SEL **:** **IN** std\_logic**;**

OUTPUT**:** **OUT** SIGNED**(**19 **downto** 0**));**

**END** **COMPONENT;**

**COMPONENT** MUX\_4\_RF **IS**

**PORT(** AR**,**AI**,**BR**,**BI**:** **IN** SIGNED**(**19 **downto** 0**);**

SEL **:** **IN** std\_logic\_vector**(**1 **downto** 0**);**

OUTPUT**:** **OUT** SIGNED**(**19 **downto** 0**));**

**END** **COMPONENT;**

**SIGNAL** ARQ**,**AIQ**,**BRQ**,**BIQ**,**WRQ**,**WIQ **:** SIGNED**(**19 **downto** 0**);**

**BEGIN**

MUX\_2BA **:** MUX\_4\_RF **PORT** **MAP(**AR**=>**ARQ**,**AI **=>**AIQ**,**BR**=>**BRQ**,**BI**=>**BIQ**,**SEL**=>**SEL2**,**OUTPUT**=>**OUT\_MUX2**);**

MUX\_3W **:** MUX\_20to1\_RF **PORT** **MAP** **(**A**=>**WRQ**,**B**=>**WIQ**,**SEL**=>**SEL3**,**OUTPUT**=>**OUT\_MUX3**);**

MUX\_1A **:** MUX\_20to1\_RF **PORT** **MAP** **(**A**=>**ARQ**,**B**=>**AIQ**,**SEL**=>**SEL1**,**OUTPUT**=>**OUT\_MUX1**);**

R\_AR **:** REG\_BF\_AB **PORT** **MAP** **(**CLK**=>**CLK**,**ENABLE**=>**ENABLE**,**D**=>**AR**,**Q**=>**ARQ**);**

R\_AI **:** REG\_BF\_AB **PORT** **MAP** **(**CLK**=>**CLK**,**ENABLE**=>**ENABLE**,**D**=>**AI**,**Q**=>**AIQ**);**

R\_BR **:** REG\_BF\_AB **PORT** **MAP** **(**CLK**=>**CLK**,**ENABLE**=>**ENABLE**,**D**=>**BR**,**Q**=>**BRQ**);**

R\_BI **:** REG\_BF\_AB **PORT** **MAP** **(**CLK**=>**CLK**,**ENABLE**=>**ENABLE**,**D**=>**BI**,**Q**=>**BIQ**);**

R\_WR **:** REG\_BF\_W **PORT** **MAP** **(**CLK**=>**CLK**,**D**=>**WR**,**Q**=>**WRQ**);**

R\_WI **:** REG\_BF\_W **PORT** **MAP** **(**CLK**=>**CLK**,**D**=>**WI**,**Q**=>**WIQ**);**

AI\_CTRL**<=** AIQ**;**

AR\_CTRL**<=** ARQ**;**

BI\_CTRL**<=** BIQ**;**

BR\_CTRL**<=** BRQ**;**

**END** BEHAV**;**

**START\_SENSE.VHD**

**LIBRARY** ieee**;**

**USE** ieee**.**std\_logic\_1164**.all;**

**USE** ieee**.**numeric\_std**.all;**

**ENTITY** start\_sense **IS**

**PORT** **(**D**,** CLK**,**set**,**enable **:** **IN** std\_logic**;**

SENSE **:** **OUT** std\_logic**);**

**END** start\_sense**;**

**ARCHITECTURE** behav **OF** start\_sense **IS**

**signal** new\_d **:** std\_logic**;**

**BEGIN**

new\_d**<=not(**D**);**

**PROCESS(**CLK**,**set**,**enable**)**

**BEGIN**

**IF** **(**CLK'**EVENT** **AND** CLK **=** '1'**)** **THEN**

**IF** **(**enable **=** '1'**)** **THEN**

SENSE **<=** new\_d**;**

**ELSE** **IF** **(**set**=**'1'**)** **then**

SENSE **<=**'1'**;**

**END** **IF;**

**END** **IF;**

**END** **IF;**

**END** **PROCESS;**

**END** behav**;**

**REG\_SUM.VHD**

**LIBRARY** ieee**;**

**USE** ieee**.**std\_logic\_1164**.all;**

**USE** ieee**.**numeric\_std**.all;**

**ENTITY** REG\_SUM **IS**

**PORT** **(** CLK**:** **IN** std\_logic**;**

D**:** **IN** SIGNED**(**39 **downto** 0**);**

Q**:** **OUT** SIGNED **(**39 **downto** 0**));**

**END** REG\_SUM**;**

**ARCHITECTURE** behav **OF** REG\_SUM **IS**

**BEGIN**

**PROCESS(**CLK**)**

**BEGIN**

**IF** **(**CLK'**EVENT** **AND** CLK **=** '1'**)** **THEN**

Q**<=** D**;**

**END** **IF;**

**END** **PROCESS;**

**END** behav**;**

**SUB\_ADDER.VHD**

**LIBRARY** ieee**;**

**USE** ieee**.**std\_logic\_1164**.all;**

**USE** ieee**.**numeric\_std**.all;**

**ENTITY** SUB\_ADDER **IS**

**PORT** **(** A\_S**:** **IN** std\_logic**;**

T1**,**T2**:** **IN** SIGNED**(**38 **downto** 0**);**

RESULT**:** **OUT** SIGNED**(**39 **downto** 0**));**

**END** SUB\_ADDER**;**

**ARCHITECTURE** behav **OF** SUB\_ADDER **IS**

**SIGNAL** sign\_ext\_t1**,**sign\_ext\_t2**:** SIGNED**(**39 **DOWNTO** 0**);**

**BEGIN**

sign\_ext\_t1**(**39**)** **<=** T1**(**38**);** --passo nel formato Q2.38

sign\_ext\_t1**(**38 **DOWNTO** 0**)** **<=** T1**(**38 **DOWNTO** 0**);**

sign\_ext\_t2**(**39**)** **<=** T2**(**38**);** --passo nel formato Q2.38

sign\_ext\_t2**(**38 **DOWNTO** 0**)** **<=** T2**(**38 **DOWNTO** 0**);**

**PROCESS(**A\_S**,**sign\_ext\_t1**,**sign\_ext\_t2**)**

**BEGIN**

**IF** **(**A\_S **=** '0'**)** **THEN**

RESULT**(**39 **DOWNTO** 0**)** **<=** sign\_ext\_t1**+**sign\_ext\_t2**;**

**END** **IF;**

**IF** **(**A\_S **=**'1'**)** **THEN**

RESULT**(**39 **DOWNTO** 0**)** **<=** sign\_ext\_t1**-(**sign\_ext\_t2**);**

**END** **IF;**

**END** **PROCESS;**

**END** behav**;**

**SUM.VHD**

**LIBRARY** ieee**;**

**USE** ieee**.**std\_logic\_1164**.all;**

**USE** ieee**.**numeric\_std**.all;**

**ENTITY** SUM **IS**

**PORT** **(** A\_S**,**CLK**:** **IN** std\_logic**;**

T1**,**T2**:** **IN** SIGNED**(**38 **downto** 0**);**

RESULT**:** **OUT** SIGNED**(**39 **downto** 0**));**

**END** SUM**;**

**ARCHITECTURE** behav **OF** SUM **IS**

**COMPONENT** REG\_SUM **IS**

**PORT** **(** CLK**:** **IN** std\_logic**;**

D**:** **IN** SIGNED**(**39 **downto** 0**);**

Q**:** **OUT** SIGNED **(**39 **downto** 0**));**

**END** **COMPONENT;**

**COMPONENT** SUB\_ADDER **IS**

**PORT** **(** A\_S**:** **IN** std\_logic**;**

T1**,**T2**:** **IN** SIGNED**(**38 **downto** 0**);**

RESULT**:** **OUT** SIGNED**(**39 **downto** 0**));**

**END** **COMPONENT;**

**SIGNAL** partial **:** SIGNED**(**39 **downto** 0**);**

**BEGIN**

SOMMATORE**:** SUB\_ADDER **PORT** **MAP(**A\_S**=>**A\_S**,**T1**=>**T1**,**T2**=>**T2**,**RESULT**=>**PARTIAL**);**

REG**:** REG\_SUM **PORT** **MAP** **(**CLK**=>**CLK**,**D**=>**partial**,**Q**=>**RESULT**);**

**END** behav**;**

**OR\_32.VHD**

**LIBRARY** ieee**;**

**USE** ieee**.**std\_logic\_1164**.all;**

**USE** ieee**.**numeric\_std**.all;**

**ENTITY** OR\_32 **IS**

**PORT** **(** INPUT**:** **IN** std\_logic\_vector**(**31 **downto** 0**);**

OUT\_XOR\_CHECK**:** **OUT** std\_logic**);**

**END** OR\_32**;**

**ARCHITECTURE** behav **OF** OR\_32 **IS**

**BEGIN**

OUT\_XOR\_CHECK **<=** INPUT**(**31**)** **OR** INPUT**(**30**)** **OR** INPUT**(**29**)** **OR** INPUT**(**28**)** **OR** INPUT**(**27**)** **OR** INPUT**(**26**)** **OR** INPUT**(**25**)** **OR** INPUT**(**24**)** **OR** INPUT**(**23**)** **OR** INPUT**(**22**)** **OR** INPUT**(**21**)** **OR** INPUT**(**20**)** **OR** INPUT**(**19**)** **OR** INPUT**(**18**)** **OR** INPUT**(**17**)** **OR** INPUT**(**16**)** **OR** INPUT**(**15**)** **OR** INPUT**(**14**)** **OR**

INPUT**(**13**)** **OR** INPUT**(**12**)** **OR** INPUT**(**11**)** **OR** INPUT**(**10**)** **OR** INPUT**(**9**)** **OR** INPUT**(**8**)** **OR** INPUT**(**7**)** **OR** INPUT**(**6**)** **OR** INPUT**(**5**)** **OR** INPUT**(**4**)** **OR** INPUT**(**3**)** **OR** INPUT**(**2**)** **OR** INPUT**(**1**)** **OR** INPUT**(**0**);**

**END** behav**;**

**XOR\_CHECK.VHD**

**LIBRARY** ieee**;**

**USE** ieee**.**std\_logic\_1164**.all;**

**USE** ieee**.**numeric\_std**.all;**

**ENTITY** XOR\_CHECK **IS**

**PORT** **(**

B0AR**,**B1AR**,**B2AR**,**B3AR**,**B4AR**,**B5AR**,**B6AR**,**B7AR**:** **IN** SIGNED**(**19 **downto** 0**);**

B0AI**,**B1AI**,**B2AI**,**B3AI**,**B4AI**,**B5AI**,**B6AI**,**B7AI**:** **IN** SIGNED**(**19 **downto** 0**);**

B0BR**,**B1BR**,**B2BR**,**B3BR**,**B4BR**,**B5BR**,**B6BR**,**B7BR**:** **IN** SIGNED**(**19 **downto** 0**);**

B0BI**,**B1BI**,**B2BI**,**B3BI**,**B4BI**,**B5BI**,**B6BI**,**B7BI**:** **IN** SIGNED**(**19 **downto** 0**);**

B0AR\_OLD**,**B1AR\_OLD**,**B2AR\_OLD**,**B3AR\_OLD**,**B4AR\_OLD**,**B5AR\_OLD**,**B6AR\_OLD**,**B7AR\_OLD**:**

**IN** SIGNED**(**19 **downto** 0**);**

B0AI\_OLD**,**B1AI\_OLD**,**B2AI\_OLD**,**B3AI\_OLD**,**B4AI\_OLD**,**B5AI\_OLD**,**B6AI\_OLD**,**B7AI\_OLD**:**

**IN** SIGNED**(**19 **downto** 0**);**

B0BR\_OLD**,**B1BR\_OLD**,**B2BR\_OLD**,**B3BR\_OLD**,**B4BR\_OLD**,**B5BR\_OLD**,**B6BR\_OLD**,**B7BR\_OLD**:**

**IN** SIGNED**(**19 **downto** 0**);**

B0BI\_OLD**,**B1BI\_OLD**,**B2BI\_OLD**,**B3BI\_OLD**,**B4BI\_OLD**,**B5BI\_OLD**,**B6BI\_OLD**,**B7BI\_OLD**:**

**IN** SIGNED**(**19 **downto** 0**);**

OUT\_XOR\_CHECK**:** **OUT** std\_logic**);**

**END** XOR\_CHECK**;**

**ARCHITECTURE** behav **OF** XOR\_CHECK **IS**

**COMPONENT** XOR\_PORT **IS**

**PORT** **(** IN\_NEW**,**IN\_OLD**:** **IN** SIGNED**(**19 **downto** 0**);**

OUT\_XOR**:** **OUT** std\_logic**);**

**END** **COMPONENT;**

**COMPONENT** OR\_32 **IS**

**PORT** **(** INPUT**:** **IN** std\_logic\_vector**(**31 **downto** 0**);**

OUT\_XOR\_CHECK**:** **OUT** std\_logic**);**

**END** **COMPONENT;**

**SIGNAL** INPUT\_S**:** std\_logic\_vector**(**31 **downto** 0**);**

**BEGIN**

AR\_0**:** XOR\_PORT **PORT** **MAP(**IN\_NEW**=>**B0AR**,**IN\_OLD**=>**B0AR\_OLD**,**OUT\_XOR**=>**INPUT\_S**(**31**));**

AR\_1**:** XOR\_PORT **PORT** **MAP(**IN\_NEW**=>**B1AR**,**IN\_OLD**=>**B1AR\_OLD**,**OUT\_XOR**=>**INPUT\_S**(**30**));**

AR\_2**:** XOR\_PORT **PORT** **MAP(**IN\_NEW**=>**B2AR**,**IN\_OLD**=>**B2AR\_OLD**,**OUT\_XOR**=>**INPUT\_S**(**29**));**

AR\_3**:** XOR\_PORT **PORT** **MAP(**IN\_NEW**=>**B3AR**,**IN\_OLD**=>**B3AR\_OLD**,**OUT\_XOR**=>**INPUT\_S**(**28**));**

AR\_4**:** XOR\_PORT **PORT** **MAP(**IN\_NEW**=>**B4AR**,**IN\_OLD**=>**B4AR\_OLD**,**OUT\_XOR**=>**INPUT\_S**(**27**));**

AR\_5**:** XOR\_PORT **PORT** **MAP(**IN\_NEW**=>**B5AR**,**IN\_OLD**=>**B5AR\_OLD**,**OUT\_XOR**=>**INPUT\_S**(**26**));**

AR\_6**:** XOR\_PORT **PORT** **MAP(**IN\_NEW**=>**B6AR**,**IN\_OLD**=>**B6AR\_OLD**,**OUT\_XOR**=>**INPUT\_S**(**25**));**

AR\_7**:** XOR\_PORT **PORT** **MAP(**IN\_NEW**=>**B7AR**,**IN\_OLD**=>**B7AR\_OLD**,**OUT\_XOR**=>**INPUT\_S**(**24**));**

AI\_0**:** XOR\_PORT **PORT** **MAP(**IN\_NEW**=>**B0AI**,**IN\_OLD**=>**B0AI\_OLD**,**OUT\_XOR**=>**INPUT\_S**(**23**));**

AI\_1**:** XOR\_PORT **PORT** **MAP(**IN\_NEW**=>**B1AI**,**IN\_OLD**=>**B1AI\_OLD**,**OUT\_XOR**=>**INPUT\_S**(**22**));**

AI\_2**:** XOR\_PORT **PORT** **MAP(**IN\_NEW**=>**B2AI**,**IN\_OLD**=>**B2AI\_OLD**,**OUT\_XOR**=>**INPUT\_S**(**21**));**

AI\_3**:** XOR\_PORT **PORT** **MAP(**IN\_NEW**=>**B3AI**,**IN\_OLD**=>**B3AI\_OLD**,**OUT\_XOR**=>**INPUT\_S**(**20**));**

AI\_4**:** XOR\_PORT **PORT** **MAP(**IN\_NEW**=>**B4AI**,**IN\_OLD**=>**B4AI\_OLD**,**OUT\_XOR**=>**INPUT\_S**(**19**));**

AI\_5**:** XOR\_PORT **PORT** **MAP(**IN\_NEW**=>**B5AI**,**IN\_OLD**=>**B5AI\_OLD**,**OUT\_XOR**=>**INPUT\_S**(**18**));**

AI\_6**:** XOR\_PORT **PORT** **MAP(**IN\_NEW**=>**B6AI**,**IN\_OLD**=>**B6AI\_OLD**,**OUT\_XOR**=>**INPUT\_S**(**17**));**

AI\_7**:** XOR\_PORT **PORT** **MAP(**IN\_NEW**=>**B7AI**,**IN\_OLD**=>**B7AI\_OLD**,**OUT\_XOR**=>**INPUT\_S**(**16**));**

BR\_0**:** XOR\_PORT **PORT** **MAP(**IN\_NEW**=>**B0BR**,**IN\_OLD**=>**B0BR\_OLD**,**OUT\_XOR**=>**INPUT\_S**(**15**));**

BR\_1**:** XOR\_PORT **PORT** **MAP(**IN\_NEW**=>**B1BR**,**IN\_OLD**=>**B1BR\_OLD**,**OUT\_XOR**=>**INPUT\_S**(**14**));**

BR\_2**:** XOR\_PORT **PORT** **MAP(**IN\_NEW**=>**B2BR**,**IN\_OLD**=>**B2BR\_OLD**,**OUT\_XOR**=>**INPUT\_S**(**13**));**

BR\_3**:** XOR\_PORT **PORT** **MAP(**IN\_NEW**=>**B3BR**,**IN\_OLD**=>**B3BR\_OLD**,**OUT\_XOR**=>**INPUT\_S**(**12**));**

BR\_4**:** XOR\_PORT **PORT** **MAP(**IN\_NEW**=>**B4BR**,**IN\_OLD**=>**B4BR\_OLD**,**OUT\_XOR**=>**INPUT\_S**(**11**));**

BR\_5**:** XOR\_PORT **PORT** **MAP(**IN\_NEW**=>**B5BR**,**IN\_OLD**=>**B5BR\_OLD**,**OUT\_XOR**=>**INPUT\_S**(**10**));**

BR\_6**:** XOR\_PORT **PORT** **MAP(**IN\_NEW**=>**B6BR**,**IN\_OLD**=>**B6BR\_OLD**,**OUT\_XOR**=>**INPUT\_S**(**9**));**

BR\_7**:** XOR\_PORT **PORT** **MAP(**IN\_NEW**=>**B7BR**,**IN\_OLD**=>**B7BR\_OLD**,**OUT\_XOR**=>**INPUT\_S**(**8**));**

BI\_0**:** XOR\_PORT **PORT** **MAP(**IN\_NEW**=>**B0BI**,**IN\_OLD**=>**B0BI\_OLD**,**OUT\_XOR**=>**INPUT\_S**(**7**));**

BI\_1**:** XOR\_PORT **PORT** **MAP(**IN\_NEW**=>**B1BI**,**IN\_OLD**=>**B1BI\_OLD**,**OUT\_XOR**=>**INPUT\_S**(**6**));**

BI\_2**:** XOR\_PORT **PORT** **MAP(**IN\_NEW**=>**B2BI**,**IN\_OLD**=>**B2BI\_OLD**,**OUT\_XOR**=>**INPUT\_S**(**5**));**

BI\_3**:** XOR\_PORT **PORT** **MAP(**IN\_NEW**=>**B3BI**,**IN\_OLD**=>**B3BI\_OLD**,**OUT\_XOR**=>**INPUT\_S**(**4**));**

BI\_4**:** XOR\_PORT **PORT** **MAP(**IN\_NEW**=>**B4BI**,**IN\_OLD**=>**B4BI\_OLD**,**OUT\_XOR**=>**INPUT\_S**(**3**));**

BI\_5**:** XOR\_PORT **PORT** **MAP(**IN\_NEW**=>**B5BI**,**IN\_OLD**=>**B5BI\_OLD**,**OUT\_XOR**=>**INPUT\_S**(**2**));**

BI\_6**:** XOR\_PORT **PORT** **MAP(**IN\_NEW**=>**B6BI**,**IN\_OLD**=>**B6BI\_OLD**,**OUT\_XOR**=>**INPUT\_S**(**1**));**

BI\_7**:** XOR\_PORT **PORT** **MAP(**IN\_NEW**=>**B7BI**,**IN\_OLD**=>**B7BI\_OLD**,**OUT\_XOR**=>**INPUT\_S**(**0**));**

OR\_LEVEL**:** OR\_32 **PORT** **MAP(**INPUT**=>**INPUT\_S**,**OUT\_XOR\_CHECK**=>**OUT\_XOR\_CHECK**);**

**END** BEHAV**;**

**XOR\_PORT.VHD**

**LIBRARY** ieee**;**

**USE** ieee**.**std\_logic\_1164**.all;**

**USE** ieee**.**numeric\_std**.all;**

**ENTITY** XOR\_PORT **IS**

**PORT** **(** IN\_NEW**,**IN\_OLD**:** **IN** SIGNED**(**19 **downto** 0**);**

OUT\_XOR**:** **OUT** std\_logic**);**

**END** XOR\_PORT**;**

**ARCHITECTURE** behav **OF** XOR\_PORT **IS**

**SIGNAL** NEW\_S**,**OLD\_S **:** std\_logic\_vector**(**19 **downto** 0**);**

**SIGNAL** OUT\_19 **:** std\_logic\_vector**(**19 **downto** 0**);**

**BEGIN**

NEW\_S**<=** std\_logic\_vector**(**IN\_NEW**);**

OLD\_S**<=** std\_logic\_vector**(**IN\_OLD**);**

OUT\_19**(**19**)** **<=** NEW\_S**(**19**)** **XOR** OLD\_S**(**19**);**

OUT\_19**(**18**)** **<=** NEW\_S**(**18**)** **XOR** OLD\_S**(**18**);**

OUT\_19**(**17**)** **<=** NEW\_S**(**17**)** **XOR** OLD\_S**(**17**);**

OUT\_19**(**16**)** **<=** NEW\_S**(**16**)** **XOR** OLD\_S**(**16**);**

OUT\_19**(**15**)** **<=** NEW\_S**(**15**)** **XOR** OLD\_S**(**15**);**

OUT\_19**(**14**)** **<=** NEW\_S**(**14**)** **XOR** OLD\_S**(**14**);**

OUT\_19**(**13**)** **<=** NEW\_S**(**13**)** **XOR** OLD\_S**(**13**);**

OUT\_19**(**12**)** **<=** NEW\_S**(**12**)** **XOR** OLD\_S**(**12**);**

OUT\_19**(**11**)** **<=** NEW\_S**(**11**)** **XOR** OLD\_S**(**11**);**

OUT\_19**(**10**)** **<=** NEW\_S**(**10**)** **XOR** OLD\_S**(**10**);**

OUT\_19**(**9**)** **<=** NEW\_S**(**9**)** **XOR** OLD\_S**(**9**);**

OUT\_19**(**8**)** **<=** NEW\_S**(**8**)** **XOR** OLD\_S**(**8**);**

OUT\_19**(**7**)** **<=** NEW\_S**(**7**)** **XOR** OLD\_S**(**7**);**

OUT\_19**(**6**)** **<=** NEW\_S**(**6**)** **XOR** OLD\_S**(**6**);**

OUT\_19**(**5**)** **<=** NEW\_S**(**5**)** **XOR** OLD\_S**(**5**);**

OUT\_19**(**4**)** **<=** NEW\_S**(**4**)** **XOR** OLD\_S**(**4**);**

OUT\_19**(**3**)** **<=** NEW\_S**(**3**)** **XOR** OLD\_S**(**3**);**

OUT\_19**(**2**)** **<=** NEW\_S**(**2**)** **XOR** OLD\_S**(**2**);**

OUT\_19**(**1**)** **<=** NEW\_S**(**1**)** **XOR** OLD\_S**(**1**);**

OUT\_19**(**0**)** **<=** NEW\_S**(**0**)** **XOR** OLD\_S**(**0**);**

OUT\_XOR**<=** OUT\_19**(**19**)** **OR** OUT\_19**(**18**)** **OR** OUT\_19**(**17**)** **OR** OUT\_19**(**16**)** **OR** OUT\_19**(**15**)** **OR** OUT\_19**(**14**)** **OR** OUT\_19**(**13**)** **OR** OUT\_19**(**12**)** **OR** OUT\_19**(**11**)** **OR** OUT\_19**(**10**)** **OR** OUT\_19**(**9**)** **OR** OUT\_19**(**8**)** **OR**

OUT\_19**(**7**)** **OR** OUT\_19**(**6**)** **OR** OUT\_19**(**5**)** **OR** OUT\_19**(**4**)** **OR** OUT\_19**(**3**)** **OR** OUT\_19**(**2**)** **OR**

OUT\_19**(**1**)** **OR** OUT\_19**(**0**);**

**END** behav**;**

**FF\_SEQ.VHD**

**LIBRARY** ieee**;**

**USE** ieee**.**std\_logic\_1164**.all;**

**USE** ieee**.**numeric\_std**.all;**

**ENTITY** FF\_SEQ **IS**

**PORT** **(**CLK**,**D**,**SET**,**RESET**,**ENABLE**:** **IN** STD\_LOGIC**;**

Q**:** **OUT** STD\_LOGIC**);**

**END** FF\_SEQ**;**

**ARCHITECTURE** behav **OF** FF\_SEQ **IS**

**BEGIN**

**PROCESS(**CLK**,**SET**,**RESET**,**ENABLE**)**

**BEGIN**

**IF** **(**CLK'**EVENT** **AND** CLK **=** '1'**)** **THEN**

**IF** **(**ENABLE **=** '1'**)** **THEN**

Q **<=** D**;**

**ELSE** **IF** **(**SET **=** '1'**)** **THEN**

Q **<=** '1'**;**

**ELSE** **IF** **(**RESET **=** '1'**)** **THEN**

Q **<=** '0'**;**

**END** **IF;**

**END** **IF;**

**END** **IF;**

**END** **IF;**

**END** **PROCESS;**

**END** behav**;**

**GUARD.VHD**

**LIBRARY** ieee**;**

**USE** ieee**.**std\_logic\_1164**.all;**

**USE** ieee**.**numeric\_std**.all;**

**ENTITY** GUARD **IS**

**PORT** **(** C0AR**,**C1AR**,**C2AR**,**C3AR**,**C4AR**,**C5AR**,**C6AR**,**C7AR**:** **IN** SIGNED**(**19 **downto** 0**);**

C0AI**,**C1AI**,**C2AI**,**C3AI**,**C4AI**,**C5AI**,**C6AI**,**C7AI**:** **IN** SIGNED**(**19 **downto** 0**);**

C0BR**,**C1BR**,**C2BR**,**C3BR**,**C4BR**,**C5BR**,**C6BR**,**C7BR**:** **IN** SIGNED**(**19 **downto** 0**);**

C0BI**,**C1BI**,**C2BI**,**C3BI**,**C4BI**,**C5BI**,**C6BI**,**C7BI**:** **IN** SIGNED**(**19 **downto** 0**);**

C0AR\_GUARD**,**C1AR\_GUARD**,**C2AR\_GUARD**,**C3AR\_GUARD**,**C4AR\_GUARD**,**C5AR\_GUARD**,**C6AR\_GUARD**,**C7AR\_GUARD**:**

**OUT** SIGNED**(**19 **downto** 0**);**

C0AI\_GUARD**,**C1AI\_GUARD**,**C2AI\_GUARD**,**C3AI\_GUARD**,**C4AI\_GUARD**,**C5AI\_GUARD**,**C6AI\_GUARD**,**C7AI\_GUARD**:**

**OUT** SIGNED**(**19 **downto** 0**);**

C0BR\_GUARD**,**C1BR\_GUARD**,**C2BR\_GUARD**,**C3BR\_GUARD**,**C4BR\_GUARD**,**C5BR\_GUARD**,**C6BR\_GUARD**,**C7BR\_GUARD**:**

**OUT** SIGNED**(**19 **downto** 0**);**

C0BI\_GUARD**,**C1BI\_GUARD**,**C2BI\_GUARD**,**C3BI\_GUARD**,**C4BI\_GUARD**,**C5BI\_GUARD**,**C6BI\_GUARD**,**C7BI\_GUARD**:**

**OUT** SIGNED**(**19 **downto** 0**));**

**END** GUARD**;**

**ARCHITECTURE** behav **OF** GUARD **IS**

**BEGIN**

C0AR\_GUARD**(**19**)** **<=** C0AR**(**19**);** C0AR\_GUARD**(**18 **DOWNTO** 0**)** **<=** C0AR**(**19 **DOWNTO** 1**);**

C1AR\_GUARD**(**19**)** **<=** C1AR**(**19**);** C1AR\_GUARD**(**18 **DOWNTO** 0**)** **<=** C1AR**(**19 **DOWNTO** 1**);**

C2AR\_GUARD**(**19**)** **<=** C2AR**(**19**);** C2AR\_GUARD**(**18 **DOWNTO** 0**)** **<=** C2AR**(**19 **DOWNTO** 1**);**

C3AR\_GUARD**(**19**)** **<=** C3AR**(**19**);** C3AR\_GUARD**(**18 **DOWNTO** 0**)** **<=** C3AR**(**19 **DOWNTO** 1**);**

C4AR\_GUARD**(**19**)** **<=** C4AR**(**19**);** C4AR\_GUARD**(**18 **DOWNTO** 0**)** **<=** C4AR**(**19 **DOWNTO** 1**);**

C5AR\_GUARD**(**19**)** **<=** C5AR**(**19**);** C5AR\_GUARD**(**18 **DOWNTO** 0**)** **<=** C5AR**(**19 **DOWNTO** 1**);**

C6AR\_GUARD**(**19**)** **<=** C6AR**(**19**);** C6AR\_GUARD**(**18 **DOWNTO** 0**)** **<=** C6AR**(**19 **DOWNTO** 1**);**

C7AR\_GUARD**(**19**)** **<=** C7AR**(**19**);** C7AR\_GUARD**(**18 **DOWNTO** 0**)** **<=** C7AR**(**19 **DOWNTO** 1**);**

C0AI\_GUARD**(**19**)** **<=** C0AI**(**19**);** C0AI\_GUARD**(**18 **DOWNTO** 0**)** **<=** C0AI**(**19 **DOWNTO** 1**);**

C1AI\_GUARD**(**19**)** **<=** C1AI**(**19**);** C1AI\_GUARD**(**18 **DOWNTO** 0**)** **<=** C1AI**(**19 **DOWNTO** 1**);**

C2AI\_GUARD**(**19**)** **<=** C2AI**(**19**);** C2AI\_GUARD**(**18 **DOWNTO** 0**)** **<=** C2AI**(**19 **DOWNTO** 1**);**

C3AI\_GUARD**(**19**)** **<=** C3AI**(**19**);** C3AI\_GUARD**(**18 **DOWNTO** 0**)** **<=** C3AI**(**19 **DOWNTO** 1**);**

C4AI\_GUARD**(**19**)** **<=** C4AI**(**19**);** C4AI\_GUARD**(**18 **DOWNTO** 0**)** **<=** C4AI**(**19 **DOWNTO** 1**);**

C5AI\_GUARD**(**19**)** **<=** C5AI**(**19**);** C5AI\_GUARD**(**18 **DOWNTO** 0**)** **<=** C5AI**(**19 **DOWNTO** 1**);**

C6AI\_GUARD**(**19**)** **<=** C6AI**(**19**);** C6AI\_GUARD**(**18 **DOWNTO** 0**)** **<=** C6AI**(**19 **DOWNTO** 1**);**

C7AI\_GUARD**(**19**)** **<=** C7AI**(**19**);** C7AI\_GUARD**(**18 **DOWNTO** 0**)** **<=** C7AI**(**19 **DOWNTO** 1**);**

C0BR\_GUARD**(**19**)** **<=** C0BR**(**19**);** C0BR\_GUARD**(**18 **DOWNTO** 0**)** **<=** C0BR**(**19 **DOWNTO** 1**);**

C1BR\_GUARD**(**19**)** **<=** C1BR**(**19**);** C1BR\_GUARD**(**18 **DOWNTO** 0**)** **<=** C1BR**(**19 **DOWNTO** 1**);**

C2BR\_GUARD**(**19**)** **<=** C2BR**(**19**);** C2BR\_GUARD**(**18 **DOWNTO** 0**)** **<=** C2BR**(**19 **DOWNTO** 1**);**

C3BR\_GUARD**(**19**)** **<=** C3BR**(**19**);** C3BR\_GUARD**(**18 **DOWNTO** 0**)** **<=** C3BR**(**19 **DOWNTO** 1**);**

C4BR\_GUARD**(**19**)** **<=** C4BR**(**19**);** C4BR\_GUARD**(**18 **DOWNTO** 0**)** **<=** C4BR**(**19 **DOWNTO** 1**);**

C5BR\_GUARD**(**19**)** **<=** C5BR**(**19**);** C5BR\_GUARD**(**18 **DOWNTO** 0**)** **<=** C5BR**(**19 **DOWNTO** 1**);**

C6BR\_GUARD**(**19**)** **<=** C6BR**(**19**);** C6BR\_GUARD**(**18 **DOWNTO** 0**)** **<=** C6BR**(**19 **DOWNTO** 1**);**

C7BR\_GUARD**(**19**)** **<=** C7BR**(**19**);** C7BR\_GUARD**(**18 **DOWNTO** 0**)** **<=** C7BR**(**19 **DOWNTO** 1**);**

C0BI\_GUARD**(**19**)** **<=** C0BI**(**19**);** C0BI\_GUARD**(**18 **DOWNTO** 0**)** **<=** C0BI**(**19 **DOWNTO** 1**);**

C1BI\_GUARD**(**19**)** **<=** C1BI**(**19**);** C1BI\_GUARD**(**18 **DOWNTO** 0**)** **<=** C1BI**(**19 **DOWNTO** 1**);**

C2BI\_GUARD**(**19**)** **<=** C2BI**(**19**);** C2BI\_GUARD**(**18 **DOWNTO** 0**)** **<=** C2BI**(**19 **DOWNTO** 1**);**

C3BI\_GUARD**(**19**)** **<=** C3BI**(**19**);** C3BI\_GUARD**(**18 **DOWNTO** 0**)** **<=** C3BI**(**19 **DOWNTO** 1**);**

C4BI\_GUARD**(**19**)** **<=** C4BI**(**19**);** C4BI\_GUARD**(**18 **DOWNTO** 0**)** **<=** C4BI**(**19 **DOWNTO** 1**);**

C5BI\_GUARD**(**19**)** **<=** C5BI**(**19**);** C5BI\_GUARD**(**18 **DOWNTO** 0**)** **<=** C5BI**(**19 **DOWNTO** 1**);**

C6BI\_GUARD**(**19**)** **<=** C6BI**(**19**);** C6BI\_GUARD**(**18 **DOWNTO** 0**)** **<=** C6BI**(**19 **DOWNTO** 1**);**

C7BI\_GUARD**(**19**)** **<=** C7BI**(**19**);** C7BI\_GUARD**(**18 **DOWNTO** 0**)** **<=** C7BI**(**19 **DOWNTO** 1**);**

**END** behav**;**

**R\_STONE\_0.VHD**

**LIBRARY** ieee**;**

**USE** ieee**.**std\_logic\_1164**.all;**

**USE** ieee**.**numeric\_std**.all;**

**ENTITY** R\_STONE\_0 **IS**

**PORT(**

OUTPUT**:** **OUT** SIGNED**(**19 **DOWNTO** 0**));**

**END** R\_STONE\_0**;**

**ARCHITECTURE** BEHAV **OF** R\_STONE\_0 **IS**

**SIGNAL** VALORE **:**SIGNED **(**19 **DOWNTO** 0**);**

**BEGIN**

VALORE**<=** "00000000000000000000"**;**

OUTPUT**<=**VALORE**;**

**END** BEHAV**;**

**R\_STONE\_1.VHD**

**LIBRARY** ieee**;**

**USE** ieee**.**std\_logic\_1164**.all;**

**USE** ieee**.**numeric\_std**.all;**

**ENTITY** R\_STONE\_1 **IS**

**PORT(**

OUTPUT\_P**,**OUTPUT\_N**:** **OUT** SIGNED**(**19 **DOWNTO** 0**));**

**END** R\_STONE\_1**;**

**ARCHITECTURE** BEHAV **OF** R\_STONE\_1 **IS**

**SIGNAL** VALORE **:**SIGNED **(**19 **DOWNTO** 0**);**

**BEGIN**

VALORE**<=** "01111111111111111111"**;**

OUTPUT\_P**<=**VALORE**;**

OUTPUT\_N**<=** **NOT** VALORE**;**

**END** BEHAV**;**

**R\_STONE\_3.VHD**

**LIBRARY** ieee**;**

**USE** ieee**.**std\_logic\_1164**.all;**

**USE** ieee**.**numeric\_std**.all;**

**ENTITY** R\_STONE\_3 **IS**

**PORT(**

OUTPUT\_P**,**OUTPUT\_N**:** **OUT** SIGNED**(**19 **DOWNTO** 0**));**

**END** R\_STONE\_3**;**

**ARCHITECTURE** BEHAV **OF** R\_STONE\_3 **IS**

**SIGNAL** VALORE **:**SIGNED **(**19 **DOWNTO** 0**);**

**BEGIN**

--VALORE<= "00110000111100000000";

VALORE**<=** "00110000111110111100"**;**

OUTPUT\_P**<=**VALORE**;**

OUTPUT\_N**(**19 **downto** 3**)<=** **NOT** VALORE**(**19 **downto** 3**);**

OUTPUT\_N**(**2 **downto** 0**)<=** VALORE**(**2 **DOWNTO** 0**);**

**END** BEHAV**;**

**R\_STONE\_7.VHD**

**LIBRARY** ieee**;**

**USE** ieee**.**std\_logic\_1164**.all;**

**USE** ieee**.**numeric\_std**.all;**

**ENTITY** R\_STONE\_7 **IS**

**PORT(**

OUTPUT\_P**,**OUTPUT\_N**:** **OUT** SIGNED**(**19 **DOWNTO** 0**));**

**END** R\_STONE\_7**;**

**ARCHITECTURE** BEHAV **OF** R\_STONE\_7 **IS**

**SIGNAL** VALORE **:**SIGNED **(**19 **DOWNTO** 0**);**

**BEGIN** --

--VALORE<= "01011010100000000000";

VALORE**<=** "01011010100000101000"**;**

OUTPUT\_P**<=**VALORE**;**

OUTPUT\_N**(**19 **downto** 4**)<=** **NOT** VALORE**(**19 **downto** 4**);**

OUTPUT\_N**(**3 **downto** 0**)<=** VALORE**(**3 **DOWNTO** 0**);**

**END** BEHAV**;**

**R\_STONE\_9.VHD**

**LIBRARY** ieee**;**

**USE** ieee**.**std\_logic\_1164**.all;**

**USE** ieee**.**numeric\_std**.all;**

**ENTITY** R\_STONE\_9 **IS**

**PORT(**

OUTPUT\_P**,**OUTPUT\_N**:** **OUT** SIGNED**(**19 **DOWNTO** 0**));**

**END** R\_STONE\_9**;**

**ARCHITECTURE** BEHAV **OF** R\_STONE\_9 **IS**

**SIGNAL** VALORE **:**SIGNED **(**19 **DOWNTO** 0**);**

**BEGIN**

--VALORE<= "01110110010000000000";

VALORE**<=** "01110110010000011011"**;**

OUTPUT\_P**<=**VALORE**;**

OUTPUT\_N**(**19 **downto** 1**)<=** **NOT** VALORE**(**19 **downto** 1**);**

OUTPUT\_N**(**0**)<=** VALORE**(**0**);**

**END** BEHAV**;**

**REG\_SEQ.VHD**

**LIBRARY** ieee**;**

**USE** ieee**.**std\_logic\_1164**.all;**

**USE** ieee**.**numeric\_std**.all;**

**ENTITY** REG\_SEQ **IS**

**PORT** **(**CLK**,**D0**,**D1**,**D2**,**D3**,**SET**,**RESET**,**EN0**,**EN1**,**EN2**,**EN3**:** **IN** STD\_LOGIC**;**

Q0**,**Q1**,**Q2**,**Q3**:** **OUT** STD\_LOGIC**);**

**END** REG\_SEQ**;**

**ARCHITECTURE** behav **OF** REG\_SEQ **IS**

**COMPONENT** FF\_SEQ **IS**

**PORT** **(**CLK**,**D**,**SET**,**RESET**,**ENABLE**:** **IN** STD\_LOGIC**;**

Q**:** **OUT** STD\_LOGIC**);**

**END** **COMPONENT;**

**BEGIN**

FF0**:** FF\_SEQ **PORT** **MAP(**CLK**=>**CLK**,**D**=>**D0**,**SET**=>**SET**,**RESET**=>**RESET**,**ENABLE**=>**EN0**,**Q**=>**Q0**);**

FF1**:** FF\_SEQ **PORT** **MAP(**CLK**=>**CLK**,**D**=>**D1**,**SET**=>**SET**,**RESET**=>**RESET**,**ENABLE**=>**EN1**,**Q**=>**Q1**);**

FF2**:** FF\_SEQ **PORT** **MAP(**CLK**=>**CLK**,**D**=>**D2**,**SET**=>**SET**,**RESET**=>**RESET**,**ENABLE**=>**EN2**,**Q**=>**Q2**);**

FF3**:** FF\_SEQ **PORT** **MAP(**CLK**=>**CLK**,**D**=>**D3**,**SET**=>**SET**,**RESET**=>**RESET**,**ENABLE**=>**EN3**,**Q**=>**Q3**);**

**END** behav**;**

**FFT\_16.VHD**

**LIBRARY** ieee**;**

**USE** ieee**.**std\_logic\_1164**.all;**

**USE** ieee**.**numeric\_std**.all;**

**ENTITY** FFT\_16 **IS**

**PORT** **(** CLK**,**RESET**:** **IN** STD\_LOGIC**;**

DONE**,**READY**:** **OUT** STD\_LOGIC**;**

C0AR**,**C1AR**,**C2AR**,**C3AR**,**C4AR**,**C5AR**,**C6AR**,**C7AR**:** **IN** SIGNED**(**19 **downto** 0**);**

C0AI**,**C1AI**,**C2AI**,**C3AI**,**C4AI**,**C5AI**,**C6AI**,**C7AI**:** **IN** SIGNED**(**19 **downto** 0**);**

C0BR**,**C1BR**,**C2BR**,**C3BR**,**C4BR**,**C5BR**,**C6BR**,**C7BR**:** **IN** SIGNED**(**19 **downto** 0**);**

C0BI**,**C1BI**,**C2BI**,**C3BI**,**C4BI**,**C5BI**,**C6BI**,**C7BI**:** **IN** SIGNED**(**19 **downto** 0**);**

C0AR\_OUT**,**C1AR\_OUT**,**C2AR\_OUT**,**C3AR\_OUT**,**C4AR\_OUT**,**C5AR\_OUT**,**C6AR\_OUT**,**C7AR\_OUT**:**

**OUT** SIGNED**(**19 **downto** 0**);**

C0AI\_OUT**,**C1AI\_OUT**,**C2AI\_OUT**,**C3AI\_OUT**,**C4AI\_OUT**,**C5AI\_OUT**,**C6AI\_OUT**,**C7AI\_OUT**:**

**OUT** SIGNED**(**19 **downto** 0**);**

C0BR\_OUT**,**C1BR\_OUT**,**C2BR\_OUT**,**C3BR\_OUT**,**C4BR\_OUT**,**C5BR\_OUT**,**C6BR\_OUT**,**C7BR\_OUT**:**

**OUT** SIGNED**(**19 **downto** 0**);**

C0BI\_OUT**,**C1BI\_OUT**,**C2BI\_OUT**,**C3BI\_OUT**,**C4BI\_OUT**,**C5BI\_OUT**,**C6BI\_OUT**,**C7BI\_OUT**:**

**OUT** SIGNED**(**19 **downto** 0**));**

**END** FFT\_16**;**

**ARCHITECTURE** behav **OF** FFT\_16 **IS**

**COMPONENT** BF **IS**

**PORT** **(** CLK**,** ENABLE**:** **IN** std\_logic**;**

AI**,**AR**,**BI**,**BR**,**WR**,**WI**:** **IN** SIGNED**(**19 **downto** 0**);** -- va cambiato il formato e conversioni

SEL\_INV**,**SEL3**,**SEL1**,**SELSUM**,**C**,**A\_S**,**EN\_REGR **:** **IN** std\_logic**;**

SEL2**:** **IN** std\_logic\_vector**(**1 **downto** 0**);**

EN\_REGO**:** **IN** std\_logic\_vector**(**2 **downto** 0**);**

A1R**,**A1I**,**B1R**,**B1I**,**AI\_CTRL**,**AR\_CTRL**,**BI\_CTRL**,**BR\_CTRL**:** **OUT** SIGNED**(**19 **downto** 0**));**

**END** **COMPONENT;**

**COMPONENT** OR\_PORT **IS**

**PORT** **(** B0AR**,**B1AR**,**B2AR**,**B3AR**,**B4AR**,**B5AR**,**B6AR**,**B7AR**:** **IN** SIGNED**(**19 **downto** 0**);**

B0AI**,**B1AI**,**B2AI**,**B3AI**,**B4AI**,**B5AI**,**B6AI**,**B7AI**:** **IN** SIGNED**(**19 **downto** 0**);**

B0BR**,**B1BR**,**B2BR**,**B3BR**,**B4BR**,**B5BR**,**B6BR**,**B7BR**:** **IN** SIGNED**(**19 **downto** 0**);**

B0BI**,**B1BI**,**B2BI**,**B3BI**,**B4BI**,**B5BI**,**B6BI**,**B7BI**:** **IN** SIGNED**(**19 **downto** 0**);**

START**:** **OUT** std\_logic**);**

**END** **COMPONENT;**

**COMPONENT** XOR\_CHECK **IS**

**PORT** **(** B0AR**,**B1AR**,**B2AR**,**B3AR**,**B4AR**,**B5AR**,**B6AR**,**B7AR**:** **IN** SIGNED**(**19 **downto** 0**);**

B0AI**,**B1AI**,**B2AI**,**B3AI**,**B4AI**,**B5AI**,**B6AI**,**B7AI**:** **IN** SIGNED**(**19 **downto** 0**);**

B0BR**,**B1BR**,**B2BR**,**B3BR**,**B4BR**,**B5BR**,**B6BR**,**B7BR**:** **IN** SIGNED**(**19 **downto** 0**);**

B0BI**,**B1BI**,**B2BI**,**B3BI**,**B4BI**,**B5BI**,**B6BI**,**B7BI**:** **IN** SIGNED**(**19 **downto** 0**);**

B0AR\_OLD**,**B1AR\_OLD**,**B2AR\_OLD**,**B3AR\_OLD**,**B4AR\_OLD**,**B5AR\_OLD**,**B6AR\_OLD**,**B7AR\_OLD**:**

**IN** SIGNED**(**19 **downto** 0**);**

B0AI\_OLD**,**B1AI\_OLD**,**B2AI\_OLD**,**B3AI\_OLD**,**B4AI\_OLD**,**B5AI\_OLD**,**B6AI\_OLD**,**B7AI\_OLD**:**

**IN** SIGNED**(**19 **downto** 0**);**

B0BR\_OLD**,**B1BR\_OLD**,**B2BR\_OLD**,**B3BR\_OLD**,**B4BR\_OLD**,**B5BR\_OLD**,**B6BR\_OLD**,**B7BR\_OLD**:**

**IN** SIGNED**(**19 **downto** 0**);**

B0BI\_OLD**,**B1BI\_OLD**,**B2BI\_OLD**,**B3BI\_OLD**,**B4BI\_OLD**,**B5BI\_OLD**,**B6BI\_OLD**,**B7BI\_OLD**:**

**IN** SIGNED**(**19 **downto** 0**);**

OUT\_XOR\_CHECK**:** **OUT** std\_logic**);**

**END** **COMPONENT;**

**COMPONENT** R\_STONE\_0 **IS**

**PORT(**

OUTPUT**:** **OUT** SIGNED**(**19 **DOWNTO** 0**));**

**END** **COMPONENT;**

**COMPONENT** R\_STONE\_1 **IS**

**PORT(**

OUTPUT\_P**,**OUTPUT\_N**:** **OUT** SIGNED**(**19 **DOWNTO** 0**));**

**END** **COMPONENT;**

**COMPONENT** R\_STONE\_3 **IS**

**PORT(**

OUTPUT\_P**,**OUTPUT\_N**:** **OUT** SIGNED**(**19 **DOWNTO** 0**));**

**END** **COMPONENT;**

**COMPONENT** R\_STONE\_7 **IS**

**PORT(**

OUTPUT\_P**,**OUTPUT\_N**:** **OUT** SIGNED**(**19 **DOWNTO** 0**));**

**END** **COMPONENT;**

**COMPONENT** R\_STONE\_9 **IS**

**PORT(**

OUTPUT\_P**,**OUTPUT\_N**:** **OUT** SIGNED**(**19 **DOWNTO** 0**));**

**END** **COMPONENT;**

**COMPONENT** CU\_BF **IS**

**PORT(** LOAD**,**SEQ**,**CLK**,**RESET**:** **IN** std\_logic**;**

CTRL\_OUT**:** **OUT** std\_logic\_vector**(**14 **downto** 0**));**

**END** **COMPONENT;**

**COMPONENT** CU\_TOP **IS**

**PORT(** START**,**PROGRESS**,**FREE\_M**,**END\_BF**,**SEQ**,**CLK**,**RESET**:** **IN** std\_logic**;**

CTRL\_TOP\_OUT**:** **OUT** std\_logic\_vector**(**5 **downto** 0**));**

**END** **COMPONENT;**

**COMPONENT** start\_sense **IS**

**PORT** **(**D**,** CLK**,**set**,**enable **:** **IN** std\_logic**;**

SENSE **:** **OUT** std\_logic**);**

**END** **COMPONENT;**

**COMPONENT** REG\_STATUS **IS**

**PORT** **(** CLK**,**set**,**enable0**,**enable1**,**enable2**,**enable3**,**RESET **:** **IN** std\_logic**;**

Q\_OUT**:** **OUT** std\_logic\_vector **(**3 **downto** 0**));**

**END** **COMPONENT;**

**COMPONENT** REG\_SEQ **IS**

**PORT** **(**CLK**,**D0**,**D1**,**D2**,**D3**,**SET**,**RESET**,**EN0**,**EN1**,**EN2**,**EN3**:** **IN** STD\_LOGIC**;**

Q0**,**Q1**,**Q2**,**Q3**:** **OUT** STD\_LOGIC**);**

**END** **COMPONENT;**

**COMPONENT** GUARD **IS**

**PORT** **(** C0AR**,**C1AR**,**C2AR**,**C3AR**,**C4AR**,**C5AR**,**C6AR**,**C7AR**:** **IN** SIGNED**(**19 **downto** 0**);**

C0AI**,**C1AI**,**C2AI**,**C3AI**,**C4AI**,**C5AI**,**C6AI**,**C7AI**:** **IN** SIGNED**(**19 **downto** 0**);**

C0BR**,**C1BR**,**C2BR**,**C3BR**,**C4BR**,**C5BR**,**C6BR**,**C7BR**:** **IN** SIGNED**(**19 **downto** 0**);**

C0BI**,**C1BI**,**C2BI**,**C3BI**,**C4BI**,**C5BI**,**C6BI**,**C7BI**:** **IN** SIGNED**(**19 **downto** 0**);**

C0AR\_GUARD**,**C1AR\_GUARD**,**C2AR\_GUARD**,**C3AR\_GUARD**,**C4AR\_GUARD**,**C5AR\_GUARD**,**C6AR\_GUARD**,**C7AR\_GUARD**:**

**OUT** SIGNED**(**19 **downto** 0**);**

C0AI\_GUARD**,**C1AI\_GUARD**,**C2AI\_GUARD**,**C3AI\_GUARD**,**C4AI\_GUARD**,**C5AI\_GUARD**,**C6AI\_GUARD**,**C7AI\_GUARD**:**

**OUT** SIGNED**(**19 **downto** 0**);**

C0BR\_GUARD**,**C1BR\_GUARD**,**C2BR\_GUARD**,**C3BR\_GUARD**,**C4BR\_GUARD**,**C5BR\_GUARD**,**C6BR\_GUARD**,**C7BR\_GUARD**:**

**OUT** SIGNED**(**19 **downto** 0**);**

C0BI\_GUARD**,**C1BI\_GUARD**,**C2BI\_GUARD**,**C3BI\_GUARD**,**C4BI\_GUARD**,**C5BI\_GUARD**,**C6BI\_GUARD**,**C7BI\_GUARD**:**

**OUT** SIGNED**(**19 **downto** 0**));**

**END** **COMPONENT;**

-- SEGNALI CTRL TOP OUT

**SIGNAL** FF\_VALUE\_S**,**EN\_FF\_S**:** STD\_LOGIC**;**

**SIGNAL** RESET\_OUT**:** STD\_LOGIC**;**

**SIGNAL** PROGRESS\_S**:** STD\_LOGIC\_VECTOR **(**3 **downto** 0**);**

**SIGNAL** EN\_REGS0**,**EN\_REGS1**,**EN\_REGS2**,**EN\_REGS3**,**FREE\_M\_S**:** STD\_LOGIC**;**

**SIGNAL** NEW\_PROGRESS\_S **:** STD\_LOGIC**;**

-- SEGNALI PER LA CU DELLE BF USCITA

**SIGNAL** SEL3\_S0**,**SEL1\_S0**,**SELSUM\_S0**,**C\_S0**,**A\_S\_S0**,**EN\_REGR\_S0**,**ENABLE\_S0**,**SEL\_INV\_S0 **:** STD\_LOGIC**;**

**SIGNAL** SEL3\_S1**,**SEL1\_S1**,**SELSUM\_S1**,**C\_S1**,**A\_S\_S1**,**EN\_REGR\_S1**,**ENABLE\_S1**,**SEL\_INV\_S1 **:** STD\_LOGIC**;**

**SIGNAL** SEL3\_S2**,**SEL1\_S2**,**SELSUM\_S2**,**C\_S2**,**A\_S\_S2**,**EN\_REGR\_S2**,**ENABLE\_S2**,**SEL\_INV\_S2 **:** STD\_LOGIC**;**

**SIGNAL** SEL3\_S3**,**SEL1\_S3**,**SELSUM\_S3**,**C\_S3**,**A\_S\_S3**,**EN\_REGR\_S3**,**ENABLE\_S3**,**SEL\_INV\_S3 **:** STD\_LOGIC**;**

**SIGNAL** SEL2\_S0**:** STD\_LOGIC\_VECTOR**(**1 **DOWNTO** 0**);**

**SIGNAL** SEL2\_S1**:** STD\_LOGIC\_VECTOR**(**1 **DOWNTO** 0**);**

**SIGNAL** SEL2\_S2**:** STD\_LOGIC\_VECTOR**(**1 **DOWNTO** 0**);**

**SIGNAL** SEL2\_S3**:** STD\_LOGIC\_VECTOR**(**1 **DOWNTO** 0**);**

**SIGNAL** EN\_REGO\_S0**:** STD\_LOGIC\_VECTOR**(**2 **DOWNTO** 0**);**

**SIGNAL** EN\_REGO\_S1**:** STD\_LOGIC\_VECTOR**(**2 **DOWNTO** 0**);**

**SIGNAL** EN\_REGO\_S2**:** STD\_LOGIC\_VECTOR**(**2 **DOWNTO** 0**);**

**SIGNAL** EN\_REGO\_S3**:** STD\_LOGIC\_VECTOR**(**2 **DOWNTO** 0**);**

--SEGNALI PER LA CU TOP INGRESSO

**SIGNAL** START\_S**,**START\_S\_TOP**,** OUT\_XOR\_CHECK\_S**,**NEW\_OUT\_XOR\_CHECK\_S **:** STD\_LOGIC**;**

**SIGNAL** END\_BF\_TOP**,**FREE\_M\_TOP **:** STD\_LOGIC**;**

-- SONO I SEGNALI DA MANDARE AI REGISTRI WR WI

**SIGNAL** P\_0**,**P\_1**,**P\_3**,**P\_7**,**P\_9 **:** SIGNED **(**19 **DOWNTO** 0**);**

**SIGNAL** N\_1**,**N\_3**,**N\_7**,**N\_9 **:**SIGNED **(**19 **DOWNTO** 0**);**

-- SEGNALI PER CONTROLLO XOR (da mettere tra step 0 e XOR\_CHECK

**SIGNAL** AI0\_CTRL**,**AI1\_CTRL**,**AI2\_CTRL**,**AI3\_CTRL**,**AI4\_CTRL**,**AI5\_CTRL**,**AI6\_CTRL**,**AI7\_CTRL **:** SIGNED **(**19 **DOWNTO** 0**);**

**SIGNAL** AR0\_CTRL**,**AR1\_CTRL**,**AR2\_CTRL**,**AR3\_CTRL**,**AR4\_CTRL**,**AR5\_CTRL**,**AR6\_CTRL**,**AR7\_CTRL **:** SIGNED **(**19 **DOWNTO** 0**);**

**SIGNAL** BR0\_CTRL**,**BR1\_CTRL**,**BR2\_CTRL**,**BR3\_CTRL**,**BR4\_CTRL**,**BR5\_CTRL**,**BR6\_CTRL**,**BR7\_CTRL **:** SIGNED **(**19 **DOWNTO** 0**);**

**SIGNAL** BI0\_CTRL**,**BI1\_CTRL**,**BI2\_CTRL**,**BI3\_CTRL**,**BI4\_CTRL**,**BI5\_CTRL**,**BI6\_CTRL**,**BI7\_CTRL **:** SIGNED **(**19 **DOWNTO** 0**);**

-- SEGNALI PER COLLEGARE UNA BF ALLA SUCCESSIVA

**SIGNAL** S01\_0\_AR**,**S01\_1\_AR**,** S01\_2\_AR**,** S01\_3\_AR**,** S01\_4\_AR**,** S01\_5\_AR**,** S01\_6\_AR**,** S01\_7\_AR**:**SIGNED**(**19 **DOWNTO** 0**);**

**SIGNAL** S01\_0\_AI**,**S01\_1\_AI**,** S01\_2\_AI**,** S01\_3\_AI**,** S01\_4\_AI**,** S01\_5\_AI**,** S01\_6\_AI**,** S01\_7\_AI**:**SIGNED**(**19 **DOWNTO** 0**);**

**SIGNAL** S01\_0\_BR**,**S01\_1\_BR**,** S01\_2\_BR**,** S01\_3\_BR**,** S01\_4\_BR**,** S01\_5\_BR**,** S01\_6\_BR**,** S01\_7\_BR**:**SIGNED**(**19 **DOWNTO** 0**);**

**SIGNAL** S01\_0\_BI**,**S01\_1\_BI**,** S01\_2\_BI**,** S01\_3\_BI**,** S01\_4\_BI**,** S01\_5\_BI**,** S01\_6\_BI**,** S01\_7\_BI**:**SIGNED**(**19 **DOWNTO** 0**);**

**SIGNAL** S12\_0\_AR**,**S12\_1\_AR**,** S12\_2\_AR**,** S12\_3\_AR**,** S12\_4\_AR**,** S12\_5\_AR**,** S12\_6\_AR**,** S12\_7\_AR**:**SIGNED**(**19 **DOWNTO** 0**);**

**SIGNAL** S12\_0\_AI**,**S12\_1\_AI**,** S12\_2\_AI**,** S12\_3\_AI**,** S12\_4\_AI**,** S12\_5\_AI**,** S12\_6\_AI**,** S12\_7\_AI**:**SIGNED**(**19 **DOWNTO** 0**);**

**SIGNAL** S12\_0\_BR**,**S12\_1\_BR**,** S12\_2\_BR**,** S12\_3\_BR**,** S12\_4\_BR**,** S12\_5\_BR**,** S12\_6\_BR**,** S12\_7\_BR**:**SIGNED**(**19 **DOWNTO** 0**);**

**SIGNAL** S12\_0\_BI**,**S12\_1\_BI**,** S12\_2\_BI**,** S12\_3\_BI**,** S12\_4\_BI**,** S12\_5\_BI**,** S12\_6\_BI**,** S12\_7\_BI**:**SIGNED**(**19 **DOWNTO** 0**);**

**SIGNAL** S23\_0\_AR**,**S23\_1\_AR**,** S23\_2\_AR**,** S23\_3\_AR**,** S23\_4\_AR**,** S23\_5\_AR**,** S23\_6\_AR**,** S23\_7\_AR**:**SIGNED**(**19 **DOWNTO** 0**);**

**SIGNAL** S23\_0\_AI**,**S23\_1\_AI**,** S23\_2\_AI**,** S23\_3\_AI**,** S23\_4\_AI**,** S23\_5\_AI**,** S23\_6\_AI**,** S23\_7\_AI**:**SIGNED**(**19 **DOWNTO** 0**);**

**SIGNAL** S23\_0\_BR**,**S23\_1\_BR**,** S23\_2\_BR**,** S23\_3\_BR**,** S23\_4\_BR**,** S23\_5\_BR**,** S23\_6\_BR**,** S23\_7\_BR**:**SIGNED**(**19 **DOWNTO** 0**);**

**SIGNAL** S23\_0\_BI**,**S23\_1\_BI**,** S23\_2\_BI**,** S23\_3\_BI**,** S23\_4\_BI**,** S23\_5\_BI**,** S23\_6\_BI**,** S23\_7\_BI**:**SIGNED**(**19 **DOWNTO** 0**);**

-- GUARD BIT

**SIGNAL** C0AR\_GUARD\_S**,**C1AR\_GUARD\_S**,**C2AR\_GUARD\_S**,**C3AR\_GUARD\_S**,**C4AR\_GUARD\_S**,**C5AR\_GUARD\_S**,**C6AR\_GUARD\_S**,**C7AR\_GUARD\_S **:** SIGNED**(**19 **DOWNTO** 0**);**

**SIGNAL** C0AI\_GUARD\_S**,**C1AI\_GUARD\_S**,**C2AI\_GUARD\_S**,**C3AI\_GUARD\_S**,**C4AI\_GUARD\_S**,**C5AI\_GUARD\_S**,**C6AI\_GUARD\_S**,**C7AI\_GUARD\_S **:** SIGNED**(**19 **DOWNTO** 0**);**

**SIGNAL** C0BR\_GUARD\_S**,**C1BR\_GUARD\_S**,**C2BR\_GUARD\_S**,**C3BR\_GUARD\_S**,**C4BR\_GUARD\_S**,**C5BR\_GUARD\_S**,**C6BR\_GUARD\_S**,**C7BR\_GUARD\_S **:** SIGNED**(**19 **DOWNTO** 0**);**

**SIGNAL** C0BI\_GUARD\_S**,**C1BI\_GUARD\_S**,**C2BI\_GUARD\_S**,**C3BI\_GUARD\_S**,**C4BI\_GUARD\_S**,**C5BI\_GUARD\_S**,**C6BI\_GUARD\_S**,**C7BI\_GUARD\_S **:** SIGNED**(**19 **DOWNTO** 0**);**

**SIGNAL** SENSE\_S**:** STD\_LOGIC**;**

**SIGNAL** SEQ\_CU0**,**SEQ\_CU1**,**SEQ\_CU2**,**SEQ\_CU3**:** STD\_LOGIC**;**

**SIGNAL** SET\_FALSO**:** STD\_LOGIC**;**

**SIGNAL** SET\_REG\_SEQ**:** STD\_LOGIC**;**

**BEGIN**

SET\_REG\_SEQ **<=** FF\_VALUE\_S **AND** NEW\_OUT\_XOR\_CHECK\_S**;**

NEW\_OUT\_XOR\_CHECK\_S **<=** OUT\_XOR\_CHECK\_S **AND** START\_S**;**

SET\_FALSO **<=** RESET\_OUT **AND** FF\_VALUE\_S**;**

START\_S\_TOP **<=** SENSE\_S **AND** START\_S**;**

NEW\_PROGRESS\_S **<=** PROGRESS\_S**(**3**)** **OR** PROGRESS\_S**(**2**)** **OR** PROGRESS\_S**(**1**)** **OR** PROGRESS\_S**(**0**);**

-- BF DELLO STEP 0 COLLEGATE DIRETTAMENTI AGLI INGRESSI DELLA FFT 16

BF\_0\_0**:** BF **PORT** **MAP(**AI**=>**C0AI\_GUARD\_S**,**AR**=>**C0AR\_GUARD\_S**,**BR**=>**C0BR\_GUARD\_S**,**BI**=>**C0BI\_GUARD\_S**,** A1R**=>**S01\_0\_AR**,**A1I**=>**S01\_0\_AI**,**B1R**=>**S01\_0\_BR**,**B1I**=>** S01\_0\_BI**,**SEL3**=>** SEL3\_S0**,**SEL1**=>**SEL1\_S0**,**SELSUM**=>**SELSUM\_S0**,**C**=>**C\_S0**,**A\_S**=>**A\_S\_S0**,**EN\_REGR**=>** EN\_REGR\_S0**,**ENABLE**=>**ENABLE\_S0**,**SEL2**=>** SEL2\_S0**,**EN\_REGO**=>**EN\_REGO\_S0**,**WR**=>**P\_1**,**WI**=>**P\_0**,**CLK**=>**CLK**,**AI\_CTRL**=>**AI1\_CTRL**,**AR\_CTRL**=>**AR0\_CTRL**,**BI\_CTRL**=>**BI0\_CTRL**,**BR\_CTRL**=>**BR0\_CTRL**,**SEL\_INV**=>**SEL\_INV\_S0**);**

BF\_0\_1**:** BF **PORT** **MAP(**AI**=>**C1AI\_GUARD\_S**,**AR**=>**C1AR\_GUARD\_S**,**BR**=>**C1BR\_GUARD\_S**,**BI**=>**C1BI\_GUARD\_S**,** A1R**=>**S01\_1\_AR**,**A1I**=>**S01\_1\_AI**,**B1R**=>**S01\_1\_BR**,**B1I**=>** S01\_1\_BI**,** SEL3**=>** SEL3\_S0**,**SEL1**=>**SEL1\_S0**,**SELSUM**=>**SELSUM\_S0**,**C**=>**C\_S0**,**A\_S**=>**A\_S\_S0**,**EN\_REGR**=>** EN\_REGR\_S0**,**ENABLE**=>**ENABLE\_S0**,**SEL2**=>** SEL2\_S0**,**EN\_REGO**=>**EN\_REGO\_S0**,**WR**=>**P\_1**,**WI**=>**P\_0**,** CLK**=>**CLK**,**AI\_CTRL**=>**AI0\_CTRL**,**AR\_CTRL**=>**AR1\_CTRL**,**BI\_CTRL**=>**BI1\_CTRL**,**BR\_CTRL**=>**BR1\_CTRL**,**SEL\_INV**=>**SEL\_INV\_S0**);**

BF\_0\_2**:** BF **PORT** **MAP(**AI**=>**C2AI\_GUARD\_S**,**AR**=>**C2AR\_GUARD\_S**,**BR**=>**C2BR\_GUARD\_S**,**BI**=>**C2BI\_GUARD\_S**,** A1R**=>**S01\_2\_AR**,**A1I**=>**S01\_2\_AI**,**B1R**=>**S01\_2\_BR**,**B1I**=>** S01\_2\_BI**,**SEL3**=>** SEL3\_S0**,**SEL1**=>**SEL1\_S0**,**SELSUM**=>**SELSUM\_S0**,**C**=>**C\_S0**,**A\_S**=>**A\_S\_S0**,** EN\_REGR**=>**EN\_REGR\_S0**,**ENABLE**=>**ENABLE\_S0**,**SEL2**=>** SEL2\_S0**,**EN\_REGO**=>**EN\_REGO\_S0**,**WR**=>**P\_1**,**WI**=>**P\_0**,**CLK**=>**CLK**,**AI\_CTRL**=>**AI2\_CTRL**,**AR\_CTRL**=>**AR2\_CTRL**,**BI\_CTRL**=>**BI2\_CTRL**,**BR\_CTRL**=>**BR2\_CTRL**,**SEL\_INV**=>**SEL\_INV\_S0**);**

BF\_0\_3**:** BF **PORT** **MAP(**AI**=>**C3AI\_GUARD\_S**,**AR**=>**C3AR\_GUARD\_S**,**BR**=>**C3BR\_GUARD\_S**,**BI**=>**C3BI\_GUARD\_S**,** A1R**=>**S01\_3\_AR**,**A1I**=>**S01\_3\_AI**,**B1R**=>**S01\_3\_BR**,**B1I**=>** S01\_3\_BI**,**SEL3**=>** SEL3\_S0**,**SEL1**=>**SEL1\_S0**,**SELSUM**=>**SELSUM\_S0**,**C**=>**C\_S0**,**A\_S**=>**A\_S\_S0**,**EN\_REGR**=>** EN\_REGR\_S0**,**ENABLE**=>**ENABLE\_S0**,**SEL2**=>** SEL2\_S0**,**EN\_REGO**=>**EN\_REGO\_S0**,**WR**=>**P\_1**,**WI**=>**P\_0**,** CLK**=>**CLK**,**AI\_CTRL**=>**AI3\_CTRL**,**AR\_CTRL**=>**AR3\_CTRL**,**BI\_CTRL**=>**BI3\_CTRL**,**BR\_CTRL**=>**BR3\_CTRL**,**SEL\_INV**=>**SEL\_INV\_S0**);**

BF\_0\_4**:** BF **PORT** **MAP(**AI**=>**C4AI\_GUARD\_S**,**AR**=>**C4AR\_GUARD\_S**,**BR**=>**C4BR\_GUARD\_S**,**BI**=>**C4BI\_GUARD\_S**,** A1R**=>**S01\_4\_AR**,**A1I**=>**S01\_4\_AI**,**B1R**=>**S01\_4\_BR**,**B1I**=>** S01\_4\_BI**,** SEL3**=>** SEL3\_S0**,**SEL1**=>**SEL1\_S0**,**SELSUM**=>**SELSUM\_S0**,**C**=>**C\_S0**,**A\_S**=>**A\_S\_S0**,**EN\_REGR**=>** EN\_REGR\_S0**,**ENABLE**=>**ENABLE\_S0**,**SEL2**=>** SEL2\_S0**,**EN\_REGO**=>**EN\_REGO\_S0**,**WR**=>**P\_1**,**WI**=>**P\_0**,** CLK**=>**CLK**,**AI\_CTRL**=>**AI4\_CTRL**,**AR\_CTRL**=>**AR4\_CTRL**,**BI\_CTRL**=>**BI4\_CTRL**,**BR\_CTRL**=>**BR4\_CTRL**,**SEL\_INV**=>**SEL\_INV\_S0**);**

BF\_0\_5**:** BF **PORT** **MAP(**AI**=>**C5AI\_GUARD\_S**,**AR**=>**C5AR\_GUARD\_S**,**BR**=>**C5BR\_GUARD\_S**,**BI**=>**C5BI\_GUARD\_S**,** A1R**=>**S01\_5\_AR**,**A1I**=>**S01\_5\_AI**,**B1R**=>**S01\_5\_BR**,**B1I**=>** S01\_5\_BI**,** SEL3**=>** SEL3\_S0**,**SEL1**=>**SEL1\_S0**,**SELSUM**=>**SELSUM\_S0**,**C**=>**C\_S0**,**A\_S**=>**A\_S\_S0**,** EN\_REGR**=>**EN\_REGR\_S0**,**ENABLE**=>**ENABLE\_S0**,**SEL2**=>** SEL2\_S0**,**EN\_REGO**=>**EN\_REGO\_S0**,**WR**=>**P\_1**,**WI**=>**P\_0**,**CLK**=>**CLK**,**AI\_CTRL**=>**AI5\_CTRL**,**AR\_CTRL**=>**AR5\_CTRL**,**BI\_CTRL**=>**BI5\_CTRL**,**BR\_CTRL**=>**BR5\_CTRL**,**SEL\_INV**=>**SEL\_INV\_S0**);**

BF\_0\_6**:** BF **PORT** **MAP(**AI**=>**C6AI\_GUARD\_S**,**AR**=>**C6AR\_GUARD\_S**,**BR**=>**C6BR\_GUARD\_S**,**BI**=>**C6BI\_GUARD\_S**,** A1R**=>**S01\_6\_AR**,**A1I**=>**S01\_6\_AI**,**B1R**=>**S01\_6\_BR**,**B1I**=>** S01\_6\_BI**,** SEL3**=>** SEL3\_S0**,**SEL1**=>**SEL1\_S0**,**SELSUM**=>**SELSUM\_S0**,**C**=>**C\_S0**,**A\_S**=>**A\_S\_S0**,**EN\_REGR**=>** EN\_REGR\_S0**,**ENABLE**=>**ENABLE\_S0**,**SEL2**=>** SEL2\_S0**,**EN\_REGO**=>**EN\_REGO\_S0**,**WR**=>**P\_1**,**WI**=>**P\_0**,** CLK**=>**CLK**,**AI\_CTRL**=>**AI6\_CTRL**,**AR\_CTRL**=>**AR6\_CTRL**,**BI\_CTRL**=>**BI6\_CTRL**,**BR\_CTRL**=>**BR6\_CTRL**,**SEL\_INV**=>**SEL\_INV\_S0**);**

BF\_0\_7**:** BF **PORT** **MAP(**AI**=>**C7AI\_GUARD\_S**,**AR**=>**C7AR\_GUARD\_S**,**BR**=>**C7BR\_GUARD\_S**,**BI**=>**C7BI\_GUARD\_S**,** A1R**=>**S01\_7\_AR**,**A1I**=>**S01\_7\_AI**,**B1R**=>**S01\_7\_BR**,**B1I**=>** S01\_7\_BI**,** SEL3**=>** SEL3\_S0**,**SEL1**=>**SEL1\_S0**,**SELSUM**=>**SELSUM\_S0**,**C**=>**C\_S0**,**A\_S**=>**A\_S\_S0**,**EN\_REGR**=>** EN\_REGR\_S0**,**ENABLE**=>**ENABLE\_S0**,**SEL2**=>** SEL2\_S0**,**EN\_REGO**=>**EN\_REGO\_S0**,**WR**=>**P\_1**,**WI**=>**P\_0**,** CLK**=>**CLK**,**AI\_CTRL**=>**AI7\_CTRL**,**AR\_CTRL**=>**AR7\_CTRL**,**BI\_CTRL**=>**BI7\_CTRL**,**BR\_CTRL**=>**BR7\_CTRL**,**SEL\_INV**=>**SEL\_INV\_S0**);**

-- BF STEP 1

BF\_1\_0**:** BF **PORT** **MAP(**AI**=>**S01\_0\_AI**,**AR**=>**S01\_0\_AR**,**BR**=>**S01\_4\_AR**,**BI**=>**S01\_4\_AI**,** A1R**=>**S12\_0\_AR**,**A1I**=>**S12\_0\_AI**,**B1R**=>** S12\_0\_BR**,**B1I**=>**S12\_0\_BI**,**SEL3**=>**SEL3\_S1**,**SEL1**=>**SEL1\_S1**,**SELSUM**=>**SELSUM\_S1**,**C**=>**C\_S1**,**A\_S**=>** A\_S\_S1**,**EN\_REGR**=>**EN\_REGR\_S1**,**ENABLE**=>**ENABLE\_S1**,**SEL2**=>** SEL2\_S1**,**EN\_REGO**=>**EN\_REGO\_S1**,**WR**=>**P\_1**,**WI**=>**P\_0**,** CLK**=>**CLK**,**SEL\_INV**=>**SEL\_INV\_S1**);**

BF\_1\_1**:** BF **PORT** **MAP(**AI**=>**S01\_1\_AI**,**AR**=>**S01\_1\_AR**,**BR**=>**S01\_5\_AR**,**BI**=>**S01\_5\_AI**,** A1R**=>**S12\_1\_AR**,**A1I**=>**S12\_1\_AI**,**B1R**=>** S12\_1\_BR**,**B1I**=>**S12\_1\_BI**,**SEL3**=>**SEL3\_S1**,**SEL1**=>** SEL1\_S1**,**SELSUM**=>**SELSUM\_S1**,**C**=>**C\_S1**,**A\_S **=>** A\_S\_S1**,**EN\_REGR**=>**EN\_REGR\_S1**,**ENABLE**=>**ENABLE\_S1**,**SEL2**=>** SEL2\_S1**,**EN\_REGO**=>**EN\_REGO\_S1**,**WR**=>**P\_1**,**WI**=>**P\_0**,** CLK**=>**CLK**,**SEL\_INV**=>**SEL\_INV\_S1**);**

BF\_1\_2**:** BF **PORT** **MAP(**AI**=>**S01\_2\_AI**,**AR**=>**S01\_2\_AR**,**BR**=>**S01\_6\_AR**,**BI**=>**S01\_6\_AI**,** A1R**=>**S12\_2\_AR**,**A1I**=>**S12\_2\_AI**,**B1R**=>** S12\_2\_BR**,**B1I**=>**S12\_2\_BI**,**SEL3**=>**SEL3\_S1**,**SEL1**=>**SEL1\_S1**,**SELSUM**=>**SELSUM\_S1**,**C**=>**C\_S1**,**A\_S**=>** A\_S\_S1**,**EN\_REGR**=>**EN\_REGR\_S1**,**ENABLE**=>**ENABLE\_S1**,**SEL2**=>** SEL2\_S1**,**EN\_REGO**=>**EN\_REGO\_S1**,**WR**=>**P\_1**,**WI**=>**P\_0**,** CLK**=>**CLK**,**SEL\_INV**=>**SEL\_INV\_S1**);**

BF\_1\_3**:** BF **PORT** **MAP(**AI**=>**S01\_3\_AI**,**AR**=>**S01\_3\_AR**,**BR**=>**S01\_7\_AR**,**BI**=>**S01\_7\_AI**,** A1R**=>**S12\_3\_AR**,**A1I**=>**S12\_3\_AI**,**B1R**=>** S12\_3\_BR**,**B1I**=>**S12\_3\_BI**,**SEL3**=>**SEL3\_S1**,**SEL1**=>**SEL1\_S1**,**SELSUM**=>**SELSUM\_S1**,**C**=>**C\_S1**,**A\_S**=>** A\_S\_S1**,**EN\_REGR**=>**EN\_REGR\_S1**,**ENABLE**=>**ENABLE\_S1**,**SEL2**=>** SEL2\_S1**,**EN\_REGO**=>**EN\_REGO\_S1**,**WR**=>**P\_1**,**WI**=>**P\_0**,** CLK**=>**CLK**,**SEL\_INV**=>**SEL\_INV\_S1**);**

BF\_1\_4**:** BF **PORT** **MAP(**AI**=>**S01\_0\_BI**,**AR**=>**S01\_0\_BR**,**BR**=>**S01\_4\_BR**,**BI**=>**S01\_4\_BI**,** A1R**=>**S12\_4\_AR**,**A1I**=>**S12\_4\_AI**,**B1R**=>** S12\_4\_BR**,**B1I**=>**S12\_4\_BI**,**SEL3**=>**SEL3\_S1**,**SEL1**=>** SEL1\_S1**,**SELSUM**=>**SELSUM\_S1**,**C**=>**C\_S1**,**A\_S**=>** A\_S\_S1**,**EN\_REGR**=>**EN\_REGR\_S1**,**ENABLE**=>**ENABLE\_S1**,**SEL2**=>** SEL2\_S1**,**EN\_REGO**=>**EN\_REGO\_S1**,**WR**=>**P\_0**,**WI**=>**N\_1**,** CLK**=>**CLK**,**SEL\_INV**=>**SEL\_INV\_S1**);**

BF\_1\_5**:** BF **PORT** **MAP(**AI**=>**S01\_1\_BI**,**AR**=>**S01\_1\_BR**,**BR**=>**S01\_5\_BR**,**BI**=>**S01\_5\_BI**,** A1R**=>**S12\_5\_AR**,**A1I**=>**S12\_5\_AI**,**B1R**=>** S12\_5\_BR**,**B1I**=>**S12\_5\_BI**,**SEL3**=>**SEL3\_S1**,**SEL1**=>**SEL1\_S1**,**SELSUM**=>**SELSUM\_S1**,**C**=>**C\_S1**,**A\_S**=>** A\_S\_S1**,**EN\_REGR**=>**EN\_REGR\_S1**,**ENABLE**=>**ENABLE\_S1**,**SEL2**=>** SEL2\_S1**,**EN\_REGO**=>**EN\_REGO\_S1**,**WR**=>**P\_0**,**WI**=>**N\_1**,** CLK**=>**CLK**,**SEL\_INV**=>**SEL\_INV\_S1**);**

BF\_1\_6**:** BF **PORT** **MAP(**AI**=>**S01\_2\_BI**,**AR**=>**S01\_2\_BR**,**BR**=>**S01\_6\_BR**,**BI**=>**S01\_6\_BI**,** A1R**=>**S12\_6\_AR**,**A1I**=>**S12\_6\_AI**,**B1R**=>** S12\_6\_BR**,**B1I**=>**S12\_6\_BI**,**SEL3**=>**SEL3\_S1**,**SEL1**=>**SEL1\_S1**,**SELSUM**=>**SELSUM\_S1**,**C**=>**C\_S1**,**A\_S**=>** A\_S\_S1**,**EN\_REGR**=>**EN\_REGR\_S1**,**ENABLE**=>**ENABLE\_S1**,**SEL2**=>** SEL2\_S1**,**EN\_REGO**=>**EN\_REGO\_S1**,**WR**=>**P\_0**,**WI**=>**N\_1**,** CLK**=>**CLK**,**SEL\_INV**=>**SEL\_INV\_S1**);**

BF\_1\_7**:** BF **PORT** **MAP(**AI**=>**S01\_3\_BI**,**AR**=>**S01\_3\_BR**,**BR**=>**S01\_7\_BR**,**BI**=>**S01\_7\_BI**,** A1R**=>**S12\_7\_AR**,**A1I**=>**S12\_7\_AI**,**B1R**=>** S12\_7\_BR**,**B1I**=>**S12\_7\_BI**,**SEL3**=>**SEL3\_S1**,**SEL1**=>**SEL1\_S1**,**SELSUM**=>**SELSUM\_S1**,**C**=>**C\_S1**,**A\_S**=>** A\_S\_S1**,**EN\_REGR**=>**EN\_REGR\_S1**,**ENABLE**=>**ENABLE\_S1**,**SEL2**=>** SEL2\_S1**,**EN\_REGO**=>**EN\_REGO\_S1**,**WR**=>**P\_0**,**WI**=>**N\_1**,** CLK**=>**CLK**,**SEL\_INV**=>**SEL\_INV\_S1**);**

-- BF STEP 2

BF\_2\_0**:** BF **PORT** **MAP(**AI**=>**S12\_0\_AI**,**AR**=>**S12\_0\_AR**,**BR**=>**S12\_2\_AR**,**BI**=>**S12\_2\_AI**,** A1R**=>**S23\_0\_AR**,**A1I**=>**S23\_0\_AI**,**B1R**=>** S23\_0\_BR**,**B1I**=>**S23\_0\_BI**,**SEL3**=>**SEL3\_S2**,**SEL1**=>**SEL1\_S2**,**SELSUM**=>**SELSUM\_S2**,**C**=>**C\_S2**,**A\_S**=>** A\_S\_S2**,**EN\_REGR**=>**EN\_REGR\_S2**,**ENABLE**=>**ENABLE\_S2**,**SEL2**=>** SEL2\_S2**,**EN\_REGO**=>**EN\_REGO\_S2**,**WR**=>**P\_1**,**WI**=>**P\_0**,** CLK**=>**CLK**,**SEL\_INV**=>**SEL\_INV\_S2**);**

BF\_2\_1**:** BF **PORT** **MAP(**AI**=>**S12\_1\_AI**,**AR**=>**S12\_1\_AR**,**BR**=>**S12\_3\_AR**,**BI**=>**S12\_3\_AI**,** A1R**=>**S23\_1\_AR**,**A1I**=>**S23\_1\_AI**,**B1R**=>** S23\_1\_BR**,**B1I**=>**S23\_1\_BI**,**SEL3**=>**SEL3\_S2**,**SEL1**=>**SEL1\_S2**,**SELSUM**=>**SELSUM\_S2**,**C**=>**C\_S2**,**A\_S**=>** A\_S\_S2**,**EN\_REGR**=>**EN\_REGR\_S2**,**ENABLE**=>**ENABLE\_S2**,**SEL2**=>** SEL2\_S2**,**EN\_REGO**=>**EN\_REGO\_S2**,**WR**=>**P\_1**,**WI**=>**P\_0**,** CLK**=>**CLK**,**SEL\_INV**=>**SEL\_INV\_S2**);**

BF\_2\_2**:** BF **PORT** **MAP(**AI**=>**S12\_0\_BI**,**AR**=>**S12\_0\_BR**,**BR**=>**S12\_2\_BR**,**BI**=>**S12\_2\_BI**,** A1R**=>**S23\_2\_AR**,**A1I**=>**S23\_2\_AI**,**B1R**=>** S23\_2\_BR**,**B1I**=>**S23\_2\_BI**,**SEL3**=>**SEL3\_S2**,**SEL1**=>**SEL1\_S2**,**SELSUM**=>**SELSUM\_S2**,**C**=>**C\_S2**,**A\_S**=>** A\_S\_S2**,**EN\_REGR**=>**EN\_REGR\_S2**,**ENABLE**=>**ENABLE\_S2**,**SEL2**=>** SEL2\_S2**,**EN\_REGO**=>**EN\_REGO\_S2**,**WR**=>**P\_0**,**WI**=>**N\_1**,** CLK**=>**CLK**,**SEL\_INV**=>**SEL\_INV\_S2**);**

BF\_2\_3**:** BF **PORT** **MAP(**AI**=>**S12\_1\_BI**,**AR**=>**S12\_1\_BR**,**BR**=>**S12\_3\_BR**,**BI**=>**S12\_3\_BI**,** A1R**=>**S23\_3\_AR**,**A1I**=>**S23\_3\_AI**,**B1R**=>** S23\_3\_BR**,**B1I**=>**S23\_3\_BI**,**SEL3**=>**SEL3\_S2**,**SEL1**=>**SEL1\_S2**,**SELSUM**=>**SELSUM\_S2**,**C**=>**C\_S2**,**A\_S**=>** A\_S\_S2**,**EN\_REGR**=>**EN\_REGR\_S2**,**ENABLE**=>**ENABLE\_S2**,**SEL2**=>** SEL2\_S2**,**EN\_REGO**=>**EN\_REGO\_S2**,**WR**=>**P\_0**,**WI**=>**N\_1**,** CLK**=>**CLK**,**SEL\_INV**=>**SEL\_INV\_S2**);**

BF\_2\_4**:** BF **PORT** **MAP(**AI**=>**S12\_4\_AI**,**AR**=>**S12\_4\_AR**,**BR**=>**S12\_6\_AR**,**BI**=>**S12\_6\_AI**,** A1R**=>**S23\_4\_AR**,**A1I**=>**S23\_4\_AI**,**B1R**=>** S23\_4\_BR**,**B1I**=>**S23\_4\_BI**,**SEL3**=>**SEL3\_S2**,**SEL1**=>**SEL1\_S2**,**SELSUM**=>**SELSUM\_S2**,**C**=>**C\_S2**,**A\_S**=>** A\_S\_S2**,**EN\_REGR**=>**EN\_REGR\_S2**,**ENABLE**=>**ENABLE\_S2**,**SEL2**=>** SEL2\_S2**,**EN\_REGO**=>**EN\_REGO\_S2**,**WR**=>**P\_7**,**WI**=>**N\_7**,** CLK**=>**CLK**,**SEL\_INV**=>**SEL\_INV\_S2**);**

BF\_2\_5**:** BF **PORT** **MAP(**AI**=>**S12\_5\_AI**,**AR**=>**S12\_5\_AR**,**BR**=>**S12\_7\_AR**,**BI**=>**S12\_7\_AI**,** A1R**=>**S23\_5\_AR**,**A1I**=>**S23\_5\_AI**,**B1R**=>** S23\_5\_BR**,**B1I**=>**S23\_5\_BI**,**SEL3**=>**SEL3\_S2**,**SEL1**=>**SEL1\_S2**,**SELSUM**=>**SELSUM\_S2**,**C**=>**C\_S2**,**A\_S**=>** A\_S\_S2**,**EN\_REGR**=>**EN\_REGR\_S2**,**ENABLE**=>**ENABLE\_S2**,**SEL2**=>** SEL2\_S2**,**EN\_REGO**=>**EN\_REGO\_S2**,**WR**=>**P\_7**,**WI**=>**N\_7**,** CLK**=>**CLK**,**SEL\_INV**=>**SEL\_INV\_S2**);**

BF\_2\_6**:** BF **PORT** **MAP(**AI**=>**S12\_4\_BI**,**AR**=>**S12\_4\_BR**,**BR**=>**S12\_6\_BR**,**BI**=>**S12\_6\_BI**,** A1R**=>**S23\_6\_AR**,**A1I**=>**S23\_6\_AI**,**B1R**=>** S23\_6\_BR**,**B1I**=>**S23\_6\_BI**,**SEL3**=>**SEL3\_S2**,**SEL1**=>**SEL1\_S2**,**SELSUM**=>**SELSUM\_S2**,**C**=>**C\_S2**,**A\_S**=>** A\_S\_S2**,**EN\_REGR**=>**EN\_REGR\_S2**,**ENABLE**=>**ENABLE\_S2**,**SEL2**=>** SEL2\_S2**,**EN\_REGO**=>**EN\_REGO\_S2**,**WR**=>**N\_7**,**WI**=>**N\_7**,** CLK**=>**CLK**,**SEL\_INV**=>**SEL\_INV\_S2**);**

BF\_2\_7**:** BF **PORT** **MAP(**AI**=>**S12\_5\_BI**,**AR**=>**S12\_5\_BR**,**BR**=>**S12\_7\_BR**,**BI**=>**S12\_7\_BI**,** A1R**=>**S23\_7\_AR**,**A1I**=>**S23\_7\_AI**,**B1R**=>** S23\_7\_BR**,**B1I**=>**S23\_7\_BI**,**SEL3**=>**SEL3\_S2**,**SEL1**=>**SEL1\_S2**,**SELSUM**=>**SELSUM\_S2**,**C**=>**C\_S2**,**A\_S**=>** A\_S\_S2**,**EN\_REGR**=>**EN\_REGR\_S2**,**ENABLE**=>**ENABLE\_S2**,**SEL2**=>** SEL2\_S2**,**EN\_REGO**=>**EN\_REGO\_S2**,**WR**=>**N\_7**,**WI**=>**N\_7**,** CLK**=>**CLK**,**SEL\_INV**=>**SEL\_INV\_S2**);**

-- BF STEP 3 (ULTIMO)

BF\_3\_0**:** BF **PORT** **MAP(**AI**=>**S23\_0\_AI**,**AR**=>**S23\_0\_AR**,**BR**=>**S23\_1\_AR**,**BI**=>**S23\_1\_AI**,** A1R**=>**C0AR\_OUT**,**A1I**=>**C0AI\_OUT**,**B1R**=>** C0BR\_OUT**,**B1I**=>**C0BI\_OUT**,**SEL3**=>**SEL3\_S3**,**SEL1**=>**SEL1\_S3**,**SELSUM**=>**SELSUM\_S3**,**C**=>**C\_S3**,**A\_S**=>** A\_S\_S3**,**EN\_REGR**=>**EN\_REGR\_S3**,**ENABLE**=>**ENABLE\_S3**,**SEL2**=>** SEL2\_S3**,**EN\_REGO**=>**EN\_REGO\_S3**,**WR**=>**P\_1**,**WI**=>**P\_0**,** CLK**=>**CLK**,**SEL\_INV**=>**SEL\_INV\_S3**);**

BF\_3\_1**:** BF **PORT** **MAP(**AI**=>**S23\_0\_BI**,**AR**=>**S23\_0\_BR**,**BR**=>**S23\_1\_BR**,**BI**=>**S23\_1\_BI**,** A1R**=>**C1AR\_OUT**,**A1I**=>**C1AI\_OUT**,**B1R**=>** C1BR\_OUT**,**B1I**=>**C1BI\_OUT**,**SEL3**=>**SEL3\_S3**,**SEL1**=>**SEL1\_S3**,**SELSUM**=>**SELSUM\_S3**,**C**=>**C\_S3**,**A\_S**=>** A\_S\_S3**,**EN\_REGR**=>**EN\_REGR\_S3**,**ENABLE**=>**ENABLE\_S3**,**SEL2**=>** SEL2\_S3**,**EN\_REGO**=>**EN\_REGO\_S3**,**WR**=>**P\_0**,**WI**=>**N\_1**,** CLK**=>**CLK**,**SEL\_INV**=>**SEL\_INV\_S3**);**

BF\_3\_2**:** BF **PORT** **MAP(**AI**=>**S23\_2\_AI**,**AR**=>**S23\_2\_AR**,**BR**=>**S23\_3\_AR**,**BI**=>**S23\_3\_AI**,** A1R**=>**C2AR\_OUT**,**A1I**=>**C2AI\_OUT**,**B1R**=>** C2BR\_OUT**,**B1I**=>**C2BI\_OUT**,**SEL3**=>**SEL3\_S3**,**SEL1**=>**SEL1\_S3**,**SELSUM**=>**SELSUM\_S3**,**C**=>**C\_S3**,**A\_S**=>** A\_S\_S3**,**EN\_REGR**=>**EN\_REGR\_S3**,**ENABLE**=>**ENABLE\_S3**,**SEL2**=>** SEL2\_S3**,**EN\_REGO**=>**EN\_REGO\_S3**,**WR**=>**P\_7**,**WI**=>**N\_7**,** CLK**=>**CLK**,**SEL\_INV**=>**SEL\_INV\_S3**);**

BF\_3\_3**:** BF **PORT** **MAP(**AI**=>**S23\_2\_BI**,**AR**=>**S23\_2\_BR**,**BR**=>**S23\_3\_BR**,**BI**=>**S23\_3\_BI**,** A1R**=>**C3AR\_OUT**,**A1I**=>**C3AI\_OUT**,**B1R**=>** C3BR\_OUT**,**B1I**=>**C3BI\_OUT**,**SEL3**=>**SEL3\_S3**,**SEL1**=>**SEL1\_S3**,**SELSUM**=>**SELSUM\_S3**,**C**=>**C\_S3**,**A\_S**=>** A\_S\_S3**,**EN\_REGR**=>**EN\_REGR\_S3**,**ENABLE**=>**ENABLE\_S3**,**SEL2**=>** SEL2\_S3**,**EN\_REGO**=>**EN\_REGO\_S3**,**WR**=>**N\_7**,**WI**=>**N\_7**,** CLK**=>**CLK**,**SEL\_INV**=>**SEL\_INV\_S3**);**

BF\_3\_4**:** BF **PORT** **MAP(**AI**=>**S23\_4\_AI**,**AR**=>**S23\_4\_AR**,**BR**=>**S23\_5\_AR**,**BI**=>**S23\_5\_AI**,** A1R**=>**C4AR\_OUT**,**A1I**=>**C4AI\_OUT**,**B1R**=>** C4BR\_OUT**,**B1I**=>**C4BI\_OUT**,**SEL3**=>**SEL3\_S3**,**SEL1**=>**SEL1\_S3**,**SELSUM**=>**SELSUM\_S3**,**C**=>**C\_S3**,**A\_S**=>** A\_S\_S3**,**EN\_REGR**=>**EN\_REGR\_S3**,**ENABLE**=>**ENABLE\_S3**,**SEL2**=>** SEL2\_S3**,**EN\_REGO**=>**EN\_REGO\_S3**,**WR**=>**P\_9**,**WI**=>**N\_3**,** CLK**=>**CLK**,**SEL\_INV**=>**SEL\_INV\_S3**);**

BF\_3\_5**:** BF **PORT** **MAP(**AI**=>**S23\_4\_BI**,**AR**=>**S23\_4\_BR**,**BR**=>**S23\_5\_BR**,**BI**=>**S23\_5\_BI**,** A1R**=>**C5AR\_OUT**,**A1I**=>**C5AI\_OUT**,**B1R**=>** C5BR\_OUT**,**B1I**=>**C5BI\_OUT**,**SEL3**=>**SEL3\_S3**,**SEL1**=>**SEL1\_S3**,**SELSUM**=>**SELSUM\_S3**,**C**=>**C\_S3**,**A\_S**=>** A\_S\_S3**,**EN\_REGR**=>**EN\_REGR\_S3**,**ENABLE**=>**ENABLE\_S3**,**SEL2**=>** SEL2\_S3**,**EN\_REGO**=>**EN\_REGO\_S3**,**WR**=>**N\_3**,**WI**=>**N\_9**,** CLK**=>**CLK**,**SEL\_INV**=>**SEL\_INV\_S3**);**

BF\_3\_6**:** BF **PORT** **MAP(**AI**=>**S23\_6\_AI**,**AR**=>**S23\_6\_AR**,**BR**=>**S23\_7\_AR**,**BI**=>**S23\_7\_AI**,** A1R**=>**C6AR\_OUT**,**A1I**=>**C6AI\_OUT**,**B1R**=>** C6BR\_OUT**,**B1I**=>**C6BI\_OUT**,**SEL3**=>**SEL3\_S3**,**SEL1**=>**SEL1\_S3**,**SELSUM**=>**SELSUM\_S3**,**C**=>**C\_S3**,**A\_S**=>** A\_S\_S3**,**EN\_REGR**=>**EN\_REGR\_S3**,**ENABLE**=>**ENABLE\_S3**,**SEL2**=>** SEL2\_S3**,**EN\_REGO**=>**EN\_REGO\_S3**,**WR**=>**P\_3**,**WI**=>**N\_9**,** CLK**=>**CLK**,**SEL\_INV**=>**SEL\_INV\_S3**);**

BF\_3\_7**:** BF **PORT** **MAP(**AI**=>**S23\_6\_BI**,**AR**=>**S23\_6\_BR**,**BR**=>**S23\_7\_BR**,**BI**=>**S23\_7\_BI**,** A1R**=>**C7AR\_OUT**,**A1I**=>**C7AI\_OUT**,**B1R**=>** C7BR\_OUT**,**B1I**=>**C7BI\_OUT**,**SEL3**=>**SEL3\_S3**,**SEL1**=>**SEL1\_S3**,**SELSUM**=>**SELSUM\_S3**,**C**=>**C\_S3**,**A\_S**=>** A\_S\_S3**,**EN\_REGR**=>**EN\_REGR\_S3**,**ENABLE**=>**ENABLE\_S3**,**SEL2**=>** SEL2\_S3**,**EN\_REGO**=>**EN\_REGO\_S3**,**WR**=>**N\_9**,**WI**=>**N\_3**,** CLK**=>**CLK**,**SEL\_INV**=>**SEL\_INV\_S3**);**

OR\_LOGIC**:**OR\_PORT **PORT** **MAP(**START**=>**START\_S**,**B0AR**=>**C0AR**,**B1AR**=>**C1AR**,**B2AR**=>**C2AR**,**B3AR**=>**C3AR**,**B4AR**=>**C4AR**,**B5AR**=>**C5AR**,** B6AR**=>**C6AR**,**B7AR**=>**C7AR**,**

B0AI**=>**C0AI**,**B1AI**=>**C1AI**,**B2AI**=>**C2AI**,**B3AI**=>**C3AI**,**B4AI**=>**C4AI**,**B5AI**=>**C5AI**,**B6AI**=>**C6AI**,**B7AI **=>**C7AI**,**B0BR**=>**C0BR**,**B1BR**=>**C1BR**,**B2BR**=>**C2BR**,**B3BR**=>**C3BR**,**B4BR**=>**C4BR**,**B5BR**=>**C5BR**,**B6BR**=>** C6BR**,**B7BR**=>**C7BR**,**B0BI**=>**C0BI**,**B1BI**=>**C1BI**,**B2BI**=>**C2BI**,**B3BI**=>**C3BI**,**B4BI**=>**C4BI**,**B5BI**=>**C5BI**,** B6BI**=>**C6BI**,**B7BI**=>**C7BI**);**

SEQ\_LOGIC**:** XOR\_CHECK **PORT** **MAP(**OUT\_XOR\_CHECK**=>**OUT\_XOR\_CHECK\_S**,**B0AR**=>**C0AR\_GUARD\_S**,**B1AR**=>**C1AR\_GUARD\_S**,**B2AR**=>**C2AR\_GUARD\_S**,**B3AR**=>**C3AR\_GUARD\_S**,** B4AR**=>**C4AR\_GUARD\_S**,**B5AR**=>**C5AR\_GUARD\_S**,**B6AR**=>**C6AR\_GUARD\_S**,**B7AR**=>**C7AR\_GUARD\_S**,**B0AI**=>**C0AI\_GUARD\_S**,**B1AI**=>**C1AI\_GUARD\_S**,**B2AI**=>**C2AI\_GUARD\_S**,**B3AI**=>** C3AI\_GUARD\_S**,**B4AI**=>**C4AI\_GUARD\_S**,**B5AI**=>**C5AI\_GUARD\_S**,**B6AI**=>**C6AI\_GUARD\_S**,**B7AI**=>**C7AI\_GUARD\_S**,**B0BR**=>**C0BR\_GUARD\_S**,**B1BR**=>**C1BR\_GUARD\_S**,**B2BR**=>**C2BR\_GUARD\_S**,** B3BR**=>**C3BR\_GUARD\_S**,**B4BR**=>**C4BR\_GUARD\_S**,**B5BR**=>**C5BR\_GUARD\_S**,**B6BR**=>**C6BR\_GUARD\_S**,**B7BR**=>**C7BR\_GUARD\_S**,**B0BI**=>**C0BI\_GUARD\_S**,**B1BI**=>**C1BI\_GUARD\_S**,**B2BI**=>** C2BI\_GUARD\_S**,**B3BI**=>**C3BI\_GUARD\_S**,**B4BI**=>**C4BI\_GUARD\_S**,**B5BI**=>**C5BI\_GUARD\_S**,**B6BI**=>**C6BI\_GUARD\_S**,**B7BI**=>**C7BI\_GUARD\_S**,**B0AR\_OLD**=>**AR0\_CTRL**,** B1AR\_OLD**=>**AR1\_CTRL**,**B2AR\_OLD**=>**AR2\_CTRL**,**B3AR\_OLD**=>**AR3\_CTRL**,**B4AR\_OLD**=>**AR4\_CTRL**,** B5AR\_OLD**=>**AR5\_CTRL**,**B6AR\_OLD**=>**AR6\_CTRL**,**B7AR\_OLD**=>**AR7\_CTRL**,**B0AI\_OLD**=>**AI0\_CTRL**,**B1AI\_OLD**=>** AI1\_CTRL**,**B2AI\_OLD**=>**AI2\_CTRL**,**B3AI\_OLD**=>**AI3\_CTRL**,**B4AI\_OLD**=>**AI4\_CTRL**,**B5AI\_OLD**=>**AI5\_CTRL**,** B6AI\_OLD**=>**AI6\_CTRL**,**B7AI\_OLD**=>**AI7\_CTRL**,**B0BR\_OLD**=>**BR0\_CTRL**,**B1BR\_OLD**=>**BR1\_CTRL**,**B2BR\_OLD**=>** BR2\_CTRL**,**B3BR\_OLD**=>**BR3\_CTRL**,**B4BR\_OLD**=>**BR4\_CTRL**,**B5BR\_OLD**=>**BR5\_CTRL**,**B6BR\_OLD**=>**BR6\_CTRL**,** B7BR\_OLD**=>**BR7\_CTRL**,**B0BI\_OLD**=>**BI0\_CTRL**,**B1BI\_OLD**=>**BI1\_CTRL**,**B2BI\_OLD**=>**BI2\_CTRL**,**B3BI\_OLD**=>** BI3\_CTRL**,**B4BI\_OLD**=>**BI4\_CTRL**,**B5BI\_OLD**=>**BI5\_CTRL**,**B6BI\_OLD**=>**BI6\_CTRL**,**B7BI\_OLD**=>**BI7\_CTRL**);**

STONE\_0**:** R\_STONE\_0 **PORT** **MAP(**OUTPUT**=>**P\_0**);**

STONE\_1**:** R\_STONE\_1 **PORT** **MAP(**OUTPUT\_P**=>**P\_1**,**OUTPUT\_N**=>**N\_1**);**

STONE\_3**:** R\_STONE\_3 **PORT** **MAP(**OUTPUT\_P**=>**P\_3**,**OUTPUT\_N**=>**N\_3**);**

STONE\_7**:** R\_STONE\_7 **PORT** **MAP(**OUTPUT\_P**=>**P\_7**,**OUTPUT\_N**=>**N\_7**);**

STONE\_9**:** R\_STONE\_9 **PORT** **MAP(**OUTPUT\_P**=>**P\_9**,**OUTPUT\_N**=>**N\_9**);**

-- CU BF

CU0 **:** CU\_BF **PORT** **MAP** **(**LOAD**=>**ENABLE\_S0**,**SEQ**=>**SEQ\_CU0**,**CLK**=>**CLK**,**RESET**=>**RESET**,**CTRL\_OUT**(**14**)=>**SEL\_INV\_S0**,**

CTRL\_OUT**(**13**)=>**SELSUM\_S0**,**CTRL\_OUT**(**12**)=>**SEL1\_S0**,**CTRL\_OUT**(**11 **downto** 10**)=>**SEL2\_S0**,**

CTRL\_OUT**(**9**)=>**SEL3\_S0**,**CTRL\_OUT**(**8**)=>**C\_S0**,**CTRL\_OUT**(**7**)=>**A\_S\_S0**,**CTRL\_OUT**(**6**)=>**EN\_REGS0**,**

CTRL\_OUT**(**5**)=>**EN\_REGR\_S0**,**CTRL\_OUT**(**4 **DOWNTO** 2**)=>**EN\_REGO\_S0**,**CTRL\_OUT**(**1**)=>**FREE\_M\_TOP**,**

CTRL\_OUT**(**0**)=>**ENABLE\_S1**);**

CU1 **:** CU\_BF **PORT** **MAP** **(**LOAD**=>**ENABLE\_S1**,**SEQ**=>**SEQ\_CU1**,**CLK**=>**CLK**,**RESET**=>**RESET**,**CTRL\_OUT**(**14**)=>**SEL\_INV\_S1**,**

CTRL\_OUT**(**13**)=>**SELSUM\_S1**,**CTRL\_OUT**(**12**)=>**SEL1\_S1**,**CTRL\_OUT**(**11 **downto** 10**)=>**SEL2\_S1**,**

CTRL\_OUT**(**9**)=>**SEL3\_S1**,**CTRL\_OUT**(**8**)=>**C\_S1**,**CTRL\_OUT**(**7**)=>**A\_S\_S1**,**CTRL\_OUT**(**6**)=>**EN\_REGS1**,**

CTRL\_OUT**(**5**)=>**EN\_REGR\_S1**,**CTRL\_OUT**(**4 **DOWNTO** 2**)=>**EN\_REGO\_S1**,**CTRL\_OUT**(**1**)=>**FREE\_M\_S**,**

CTRL\_OUT**(**0**)=>**ENABLE\_S2**);**

CU2 **:** CU\_BF **PORT** **MAP** **(**LOAD**=>**ENABLE\_S2**,**SEQ**=>**SEQ\_CU2**,**CLK**=>**CLK**,**RESET**=>**RESET**,**CTRL\_OUT**(**14**)=>**SEL\_INV\_S2**,**

CTRL\_OUT**(**13**)=>**SELSUM\_S2**,**CTRL\_OUT**(**12**)=>**SEL1\_S2**,**CTRL\_OUT**(**11 **downto** 10**)=>**SEL2\_S2**,**

CTRL\_OUT**(**9**)=>**SEL3\_S2**,**CTRL\_OUT**(**8**)=>**C\_S2**,**CTRL\_OUT**(**7**)=>**A\_S\_S2**,**CTRL\_OUT**(**6**)=>**EN\_REGS2**,**

CTRL\_OUT**(**5**)=>**EN\_REGR\_S2**,**CTRL\_OUT**(**4 **DOWNTO** 2**)=>**EN\_REGO\_S2**,**CTRL\_OUT**(**1**)=>**FREE\_M\_S**,**

CTRL\_OUT**(**0**)=>**ENABLE\_S3**);**

CU3 **:** CU\_BF **PORT** **MAP** **(**LOAD**=>**ENABLE\_S3**,**SEQ**=>**SEQ\_CU3**,**CLK**=>**CLK**,**RESET**=>**RESET**,**CTRL\_OUT**(**14**)=>**SEL\_INV\_S3**,**

CTRL\_OUT**(**13**)=>**SELSUM\_S3**,**CTRL\_OUT**(**12**)=>**SEL1\_S3**,**CTRL\_OUT**(**11 **downto** 10**)=>**SEL2\_S3**,**

CTRL\_OUT**(**9**)=>**SEL3\_S3**,**CTRL\_OUT**(**8**)=>**C\_S3**,**CTRL\_OUT**(**7**)=>**A\_S\_S3**,**CTRL\_OUT**(**6**)=>**EN\_REGS3**,**

CTRL\_OUT**(**5**)=>**EN\_REGR\_S3**,**CTRL\_OUT**(**4 **DOWNTO** 2**)=>**EN\_REGO\_S3**,**CTRL\_OUT**(**1**)=>**FREE\_M\_S**,**

CTRL\_OUT**(**0**)=>**END\_BF\_TOP**);**

--CU TOP

TOP **:** CU\_TOP **PORT** **MAP** **(**START**=>**START\_S\_TOP**,**PROGRESS**=>**NEW\_PROGRESS\_S**,**FREE\_M**=>**FREE\_M\_TOP**,**END\_BF**=>**END\_BF\_TOP**,**

SEQ**=>**NEW\_OUT\_XOR\_CHECK\_S**,**CLK**=>**CLK**,**RESET**=>**RESET**,**

CTRL\_TOP\_OUT**(**5**)=>**ENABLE\_S0**,**CTRL\_TOP\_OUT**(**4**)=>**EN\_FF\_S**,**CTRL\_TOP\_OUT**(**3**)=>**FF\_VALUE\_S**,**

CTRL\_TOP\_OUT**(**2**)=>**DONE**,**CTRL\_TOP\_OUT**(**1**)=>**READY**,**CTRL\_TOP\_OUT**(**0**)=>**RESET\_OUT**);**

-- SENSE START

SS **:** start\_sense **PORT** **MAP(**D**=>**EN\_FF\_S**,**CLK**=>**CLK**,**set**=>**RESET\_OUT**,**enable**=>**EN\_FF\_S**,**SENSE**=>**SENSE\_S**);**

-- REG STATUS

RS **:** REG\_STATUS **PORT** **MAP** **(**CLK**=>**CLK**,**set**=>**ENABLE\_S0**,**enable0**=>**EN\_REGS0**,**enable1**=>**EN\_REGS1**,**enable2**=>**EN\_REGS2**,**

enable3**=>**EN\_REGS3**,**Q\_OUT**=>**PROGRESS\_S**,**RESET**=>**RESET\_OUT**);**

-- REG SEQ

RSEQ **:** REG\_SEQ **PORT** **MAP** **(**CLK**=>**CLK**,**D0**=>**NEW\_OUT\_XOR\_CHECK\_S**,**D1**=>**PROGRESS\_S**(**0**),**D2**=>**PROGRESS\_S**(**1**),**D3**=>**PROGRESS\_S**(**2**),**

SET**=>**SET\_REG\_SEQ**,**RESET**=>**RESET\_OUT**,**EN0**=>**ENABLE\_S1**,**EN1**=>**ENABLE\_S2**,**

EN2**=>**ENABLE\_S3**,**EN3**=>**END\_BF\_TOP**,**Q0**=>**SEQ\_CU0**,**Q1**=>**SEQ\_CU1**,**Q2**=>**SEQ\_CU2**,**Q3**=>**SEQ\_CU3**);**

GB **:** GUARD **PORT** **MAP** **(**C0AR**=>**C0AR**,**C1AR**=>**C1AR**,**C2AR**=>**C2AR**,**C3AR**=>**C3AR**,**C4AR**=>**C4AR**,**C5AR**=>**C5AR**,**C6AR**=>**C6AR**,**C7AR**=>**C7AR**,**

C0AI**=>**C0AI**,**C1AI**=>**C1AI**,**C2AI**=>**C2AI**,**C3AI**=>**C3AI**,**C4AI**=>**C4AI**,**C5AI**=>**C5AI**,**C6AI**=>**C6AI**,**C7AI**=>**C7AI**,**

C0BR**=>**C0BR**,**C1BR**=>**C1BR**,**C2BR**=>**C2BR**,**C3BR**=>**C3BR**,**C4BR**=>**C4BR**,**C5BR**=>**C5BR**,**C6BR**=>**C6BR**,**C7BR**=>**C7BR**,**

C0BI**=>**C0BI**,**C1BI**=>**C1BI**,**C2BI**=>**C2BI**,**C3BI**=>**C3BI**,**C4BI**=>**C4BI**,**C5BI**=>**C5BI**,**C6BI**=>**C6BI**,**C7BI**=>**C7BI**,**

C0AR\_GUARD**=>**C0AR\_GUARD\_S**,**C1AR\_GUARD**=>**C1AR\_GUARD\_S**,**C2AR\_GUARD**=>**C2AR\_GUARD\_S**,**C3AR\_GUARD**=>**C3AR\_GUARD\_S**,**

C4AR\_GUARD**=>**C4AR\_GUARD\_S**,**C5AR\_GUARD**=>**C5AR\_GUARD\_S**,**C6AR\_GUARD**=>**C6AR\_GUARD\_S**,**C7AR\_GUARD**=>**C7AR\_GUARD\_S**,**

C0AI\_GUARD**=>**C0AI\_GUARD\_S**,**C1AI\_GUARD**=>**C1AI\_GUARD\_S**,**C2AI\_GUARD**=>**C2AI\_GUARD\_S**,**C3AI\_GUARD**=>**C3AI\_GUARD\_S**,**

C4AI\_GUARD**=>**C4AI\_GUARD\_S**,**C5AI\_GUARD**=>**C5AI\_GUARD\_S**,**C6AI\_GUARD**=>**C6AI\_GUARD\_S**,**C7AI\_GUARD**=>**C7AI\_GUARD\_S**,**

C0BR\_GUARD**=>**C0BR\_GUARD\_S**,**C1BR\_GUARD**=>**C1BR\_GUARD\_S**,**C2BR\_GUARD**=>**C2BR\_GUARD\_S**,**C3BR\_GUARD**=>**C3BR\_GUARD\_S**,**

C4BR\_GUARD**=>**C4BR\_GUARD\_S**,**C5BR\_GUARD**=>**C5BR\_GUARD\_S**,**C6BR\_GUARD**=>**C6BR\_GUARD\_S**,**C7BR\_GUARD**=>**C7BR\_GUARD\_S**,**

C0BI\_GUARD**=>**C0BI\_GUARD\_S**,**C1BI\_GUARD**=>**C1BI\_GUARD\_S**,**C2BI\_GUARD**=>**C2BI\_GUARD\_S**,**C3BI\_GUARD**=>**C3BI\_GUARD\_S**,**

C4BI\_GUARD**=>**C4BI\_GUARD\_S**,**C5BI\_GUARD**=>**C5BI\_GUARD\_S**,**C6BI\_GUARD**=>**C6BI\_GUARD\_S**,**C7BI\_GUARD**=>**C7BI\_GUARD\_S**);**

**END** BEHAV**;**