



Regole per la scrittura di VHDL Sintetizzabile

Fabio Campi

Corso di Elettronica dei Sistemi Digitali LS
AA 2003-2004

VHDL Sintetizzabile

- **Obiettivo:** Mappare su celle Hardware di libreria, riferite alla tecnologia di integrazione scelta, le funzionalita' volute, rispettando una serie di specifiche FISICHE prefissate (frequenza di funzionamento, Area Occupata, Consumo di Potenza).

=> **VHDL viene utilizzato come strumento CAD**

Regole per la Sintetizzabilita'

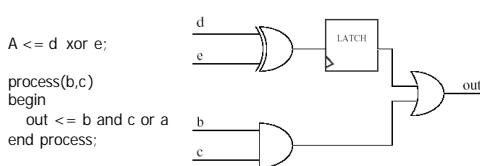
1. I segnali dovrebbero essere di tipo Std_logic_vector, in modo da rappresentare le caratteristiche "elettriche" del segnale (Verificare conflitti "X", valori non definiti "U", three-state "Z" e configurazioni di pull-up "H" e pull-down "W").
2. Usare MASSIMA attenzione nell'uso di dati strutturati quali array, records e tipi enumerativi (mantenere controllo sulle dimensioni dell'hardware istanziato)
3. Tutti I segnali che entrano in un processo combinatorio devono appartenere alla sensitivity list (altrimenti verranno istanziati latches e/o registri "parassiti").
4. I costrutti condizionali (if/case) devono coprire tutte le opzioni possibili
5. I blocchi combinatori debbono essere strettamente separati dalle parti di logica sequenziale, concentrate in espliciti registri.
6. Non utilizzarei LATCHES, gli elementi sequenziali dell'RTL debbono essere F/F
7. Il parallelismo deve essere quanto piu' possibile reso esplicito

Regole per la Sintetizzabilita' (1,2)

Obiettivo: Mantenere corrispondenza tra codice descritto e risorse hardware istanziate

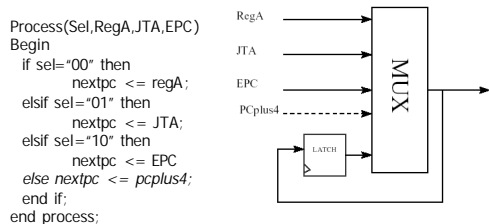
VHDL: Type op is (add,sub,and,or,xor,not);	VHDL sintetizzabile: Entity ALU is Port (a,b in : SIGNED(7 downto 0); op in : SIGNED(3 downto 0); c in : SIGNED(7 downto 0)); End ALU;
Entity ALU is Port (a,b in : Integer; op in : alu_op; c in : Integer); End ALU;	

Regole per la Sintetizzabilita' (3)



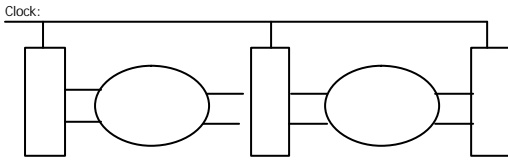
Se uno dei segnali letti da I processo non appartenesse alla sensitivity list il sintetizzatore istanzierebbe un elemento di memorizzazione

Regole per la Sintetizzabilita' (4)



Se uno dei possibili valori del selettore non fosse compreso tra le scelte il sintetizzatore (potrebbe) istanziare un elemento di memorizzazione

Regole per la Sintetizzabilita' (5,6)



Descrizione RTL: La sintesi logica partiziona il codice in blocchi combinatori che analizza indipendentemente. Il CLK deve poter essere considerato una Net IDEALE e NON SI PUO' USARE UNA NET DI SEGNALE come Sincronizzazione. Per questo e' piuttosto difficile usare latches, e se necessario va' fatto con grande attenzione

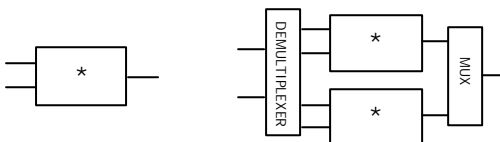
Tecniche per evidenziare parallelismo (definizioni)

- **LATENZA (Latency):** Tempo intercorso tra l'ingresso (il campionamento) in un circuito di un ingresso e la produzione dell'uscita
- **ISSUE DELAY:** Tempo intercorso tra due successivi ingressi (e quindi tra due successive uscite).

L' *Issue Delay* e' un parametro molto piu' rilevante perche' determina il throughput, ossia la quantita' di dati che e' possibile elaborare in un dato tempo.

Unfolding

- **Unfolding:** Tecnica basata sulla ripetizione di blocchi critici di logica in modo da aumentare la "produttivita'" del circuito (viene sacrificata l'area occupata dal circuito per diminuire latenza ed issue delay)

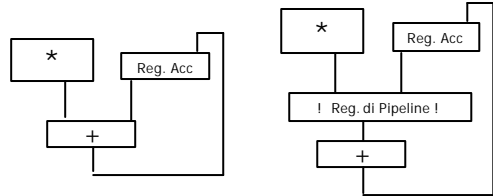


Tecniche per evidenziare parallelismo

- I linguaggi di programmazione imperativi descrivono algoritmi in maniera SEQUENZIALE. Obiettivo del programmatore e' quindi il minimizzare il numero di istruzioni consecutive con cui descrivere il proprio algoritmo
- I linguaggi di descrizione hardware non descrivono ISTRUZIONI, ma RISORSE di CALCOLO (-> transistor o meglio porte CMOS). Obiettivo del progettista e' PARALLELIZZARE il piu' possibile la esecuzione in modo da minimizzare il tempo di calcolo

Pipelining

- **Pipelining:** Tecnica basata sul partizionamento del percorso critico su successivi periodo di clock: (viene sacrificata una maggiore latenza per diminuire issue delay)



Unfolding (2)

