

Compito di linguaggi di descrizione dell'hardware

Esercizio 1

Si realizzi un modello comportamentale in VHDL di un generatore di segnale di clock. In particolare, se l'ingresso *enable* é a 0, l'uscita *clk* vale 0, mentre se vale 1 lascia che su *clk* venga prodotto un segnale periodico. Il circuito riceve come parametri generic il tempo per cui il clock sta a 1 (*h*) e il periodo di clock (*T*). Il duty cycle del clock é pari quindi a h/T . Il componente deve anche verificare che sia $T > 200ps$, altrimenti l'uscita deve andare a X (pt. 5.0).

Soluzione

<https://www.edaplayground.com/x/ggFc>

Esercizio 2

Si descriva al livello comportamentale in VHDL un componente combinatorio che riceve in ingresso due parole (di tipo `std_logic_vector`) *a* e *b* di 8 bit e un bit *m*. Se $m = 0$, gli ingressi rappresentano interi senza segno, altrimenti interi con segno. Compito della rete é produrre in uscita la somma degli ingressi. Si noti che l'uscita *s* é di tipo `std_logic_vector` su 8 bit e rappresenta un intero consistente con il tipo di ingressi. Si ha poi un bit *x* di uscita che se gli interi sono senza segno rappresenta il bit di maggior peso del risultato e un indicazione di overflow se gli interi hanno il segno (pt. 5.0).

Soluzione

<https://www.edaplayground.com/x/tLD7>

Esercizio 3

Si consideri il seguente algoritmo:

0. $u0 := a + b;$
1. $u1 := c + u0;$
2. $u2 := u1 * d;$
3. $u3 := e + f;$
4. $u4 := g + h;$
5. $u5 := u2 * i;$
6. $u6 := u3 * k;$
7. $u7 := u5 * u6;$
8. $u8 := u4 + u7;$
9. $u9 := u4 * u6;$

Si tracci il DFG dell'algoritmo. Utilizzando l'ipotesi di ciclo singolo, si determinino lo scheduling ASAP e quello ALAP individuando le risorse da allocare in ciascuno dei due casi. Si individui poi una soluzione con latenza minima che minimizzando il numero di risorse utilizzate. Si analizzi il critical path del DFG per vedere se é possibile riformulare l'algoritmo in modo da ridurre la latenza (pt. 5.0).