

Cognome e Nome: _____ Matricola _____

Programma: vecchio (senza prova pratica) ☐ nuovo (con prova pratica) ☐

Prima della consegna barrare una delle due caselle sottostanti. L'opzione scelta non può essere modificata dopo la consegna.

Intendo sostenere la prova orale nell'appello di Febbraio. Prendo atto che, a seguito della mia decisione, le altre prove consegnate cesseranno di essere valide al termine di questo appello e non potranno essere usate per l'appello straordinario di Aprile. ☐

Ho diritto a ed intenzione di rimandare la prova orale all'appello straordinario di Aprile. Prendo atto che il mio diritto a rimandare la prova orale sarà oggetto di verifica, e che dovrò ripetere l'intero esame da capo se la verifica darà esiti negativi (per qualunque motivo). ☐

Esercizio 1

Sintetizzare un circuito di *calcolo dell'opposto* per numeri interi in base β su n cifre, rappresentati in traslazione.

- Esprimere la relazione algebrica tra la rappresentazione dell'uscita B e quella dell'ingresso A
- Disegnare uno schema valido per qualunque base, facendo uso soltanto di reti trattate a lezione
- Per il caso $\beta=6$ spingere la sintesi fino al livello delle porte logiche elementari

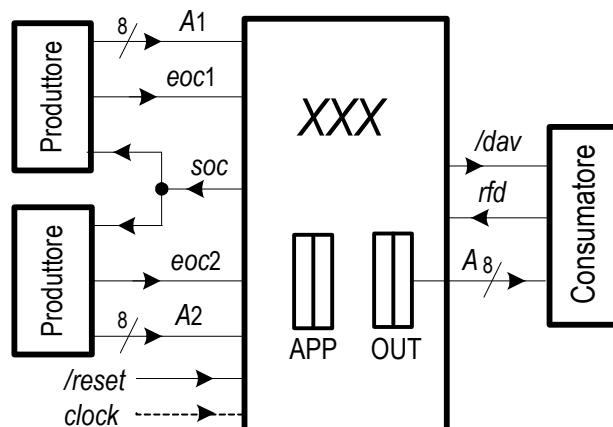
Esercizio 2

Descrivere l'Unità XXX che compie all'infinito cicli di durata pari a 1023 periodi del clock portando avanti, in ciascuno di essi, le seguenti azioni:

“Prelievo dai Produttori di due numeri naturali A_1 e A_2 e invio al Consumatore di A_1 o di A_2 considerando A_1 e A_2 come le rappresentazioni in complemento a due di due numeri interi a_1 e a_2 ed inviando A_1 se **il valore assoluto** di a_1 è maggiore o uguale al **valore assoluto** di a_2 , altrimenti A_2 ”.

NOTA 1: Si definisca, in Verilog, una funzione *mia_funzione*(A_1, A_2) per individuare quale numero, fra A_1 e A_2 , deve essere mandato al Consumatore e quindi si disegni una comprensibile struttura della corrispondente rete combinatoria in termini di sottoreti note.

NOTA 2: Non **aggiungere** né **togliere** variabili di ingresso o di uscita e ricordare che il **Verilog** è **case-sensitive**. I due Produttori **non hanno lo stesso tempo di risposta**. La somma del tempo di risposta del più lento dei Produttori con il tempo di risposta del Consumatore è sufficientemente inferiore ai 1023 periodi di clock da non creare alcun problema di nessun tipo. Al reset iniziale si supponga che tutto avvenga come se XXX avesse ricevuto 'H00 da entrambi i Produttori.

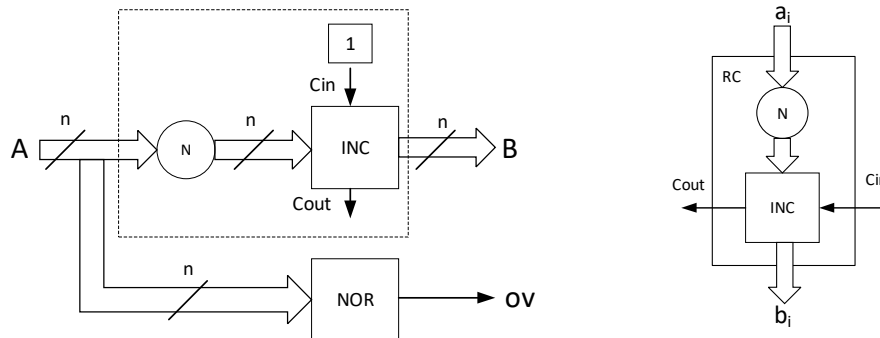


Esercizio 1 - soluzione

In traslazione abbiamo $A = a + \beta^n/2$. Se $a \neq -\beta^n/2$, cioè se $A \neq 0$, l'opposto esiste, ed abbiamo $B = (-a) + \beta^n/2 = (\beta^n/2 - A) + \beta^n/2 = \beta^n - A = (\beta^n - 1 - A) + 1 = \bar{A} + 1$.

Pertanto:

- La parte che rileva la fattibilità deve dare $ov=1$ s.se $A = 0$. Avremo quindi $ov=NOR(A)$, intendendo che vanno portati in ingresso alla NOR tutti i bit delle codifiche delle n cifre che compongono A .
- La parte che produce l'uscita B deve effettuare il complemento e l'incremento della rappresentazione del numero in ingresso, come nella figura a sinistra.



Visto che complemento e somma sono operazioni scomponibili su cifre singole, il circuito racchiuso dalla linea tratteggiata può essere scomposto in circuiti che lavorano su cifre singole, come quello nella figura a destra, collegati tramite catena dei riporti. La tabella di verità per la rete RC in base 6 è scritta sotto, e la sintesi SP corrispondente è riportata a destra.

C_{in}	a_2	a_1	a_0	b_2	b_1	b_0	C_{out}
0	0	0	0	1	0	1	0
0	0	0	1	1	0	0	0
0	0	1	0	0	1	1	0
0	0	1	1	0	1	0	0
0	1	0	0	0	0	1	0
0	1	0	1	0	0	0	0
0	1	1	0	-	-	-	-
0	1	1	1	-	-	-	-
1	0	0	0	0	0	0	1
1	0	0	1	1	0	1	0
1	0	1	0	1	0	0	0
1	0	1	1	0	1	1	0
1	1	0	0	0	1	0	0
1	1	0	1	0	0	1	0
1	1	1	0	-	-	-	-
1	1	1	1	-	-	-	-

		$C_{in} a_2$			
$a_1 a_0$		00	01	11	10
		$b_2 b_1 b_0 C_{out}$			
00		1010	0010	0100	0001
01		1000	0000	0010	1010
11		0100	----	----	0110
10		0110	----	----	1000

$$C_{out} = C_{in} \cdot \bar{a}_2 \cdot \bar{a}_1 \cdot \bar{a}_0$$

$$b_2 = \bar{C}_{in} \cdot \bar{a}_2 \cdot \bar{a}_1 + \bar{a}_2 \cdot \bar{a}_1 \cdot a_0 + C_{in} \cdot a_1 \cdot \bar{a}_0$$

$$b_1 = a_1 \cdot a_0 + \bar{C}_{in} \cdot a_1 + C_{in} \cdot a_2 \cdot \bar{a}_0$$

$$b_0 = \bar{C}_{in} \cdot \bar{a}_0 + C_{in} \cdot a_0$$

Il rilevatore di overflow può essere ottimizzato, osservando che l'unico caso in cui c'è overflow è quello in cui le cifre più significative di A e di B sono entrambe zero, e quindi basta portare in ingresso alla NOR le sole codifiche di a_{n-1} e b_{n-1} .

Soluzione esercizio 2

Una Soluzione con *mia_funzione* che fornisce A_1 o A_2 essendo guidata esclusivamente dal prestito in un sottrattore per numeri naturali.

```

module YYY(soc,eoc1,eoc2,A1,A2, dav_,rfd,A, clock, reset_);
    input  clock, reset_ ;
    output soc; input eoc1,eoc2;
    input[7:0]  A1, A2;
    input rfd;  output dav_;
    output[7:0] A;
    reg SOC,DAV_ ;          assign soc=SOC; assign dav_=DAV_;
    reg [7:0] OUT, APP;  assign A=OUT;
    reg [9:0] COUNT;
    reg [2:0] STAR;      parameter S0=0, S1=1, S2=2, S3=3, S4=4;
    parameter Num_Periodi=16;
    always @(reset_==0) begin COUNT<=Num_Periodi; SOC=0; DAV_=1; APP<='H00;
STAR=S0; end
    always @(posedge clock) if (reset_==1) #3
        casex(STAR)
            S0:  begin COUNT<=COUNT-1; OUT<=APP; SOC<=1;
                    STAR<=({eoc1,eoc2}=='B00)?S1:S0;end
            S1:  begin COUNT<=COUNT-1; SOC<=0; APP<=mia_funzione(A1,A2);
                    STAR<=({eoc1,eoc2}=='B11)?S2:S1; end
            S2:  begin COUNT<=COUNT-1; DAV_<=0; STAR<=(rfd==0)?S3:S2; end
            S3:  begin COUNT<=COUNT-1; DAV_<=1; STAR<=(rfd==1)?S4:S3; end
            S4:  begin COUNT<=(COUNT==1)?Num_Periodi:(COUNT-1); STAR<=(COUNT==1)?S0:S4;
        end
        endcase
        function [7:0] mia_funzione;
            input [7:0]  A1,A2;
            mia_funzione=((ABS_di_integer(A1)-ABS_di_integer(A2))<0)?A2:A1;
        endfunction
        function [7:0] ABS_di_integer;
            input [7:0]  A;
            ABS_di_integer=(A[7]==0)?A:(~A+1);
        endfunction
    endmodule

```