

Esercizio 1

Data la seguente mappa:

x_1x_0 \ x_3x_2	x_3x_2			
	00	01	11	10
00	-	0	0	1
01	1	1	-	-
11	0	-	0	0
10	-	1	0	1

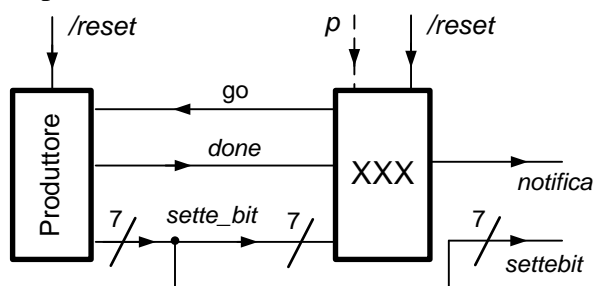
z

1. indicare e classificare gli implicant principali;
2. trovare tutte le possibili liste di copertura non ridondanti, ed individuare quelle cui corrispondono forme di tipo SP di costo minimo secondo il criterio di costo a porte;
3. **(solo per chi ha seguito fino al 2017-18 e non deve svolgere la prova pratica)** per ognuna delle liste di copertura non ridondanti di costo minimo trovate nel punto 2, individuare e classificare le eventuali alee del primo ordine presenti, e modificare la corrispondente lista in modo da eliminare tali alee;

Specificare le espressioni utilizzando esclusivamente le variabili e l'ordinamento della mappa.

Esercizio 2**Descrivere** il circuito XXX che si evolve all'infinito come segue:

- 1) preleva *sette_bit* dal Produttore, sostenendo un handshake *go*, *done* (vedi sotto);
- 2) se *sette_bit* è la codifica ASCII di H, I, ..., N, O, (in bit: 1001000, 1001001, ..., 1001111) notifica ciò tenendo *notifica* a 1 per un ciclo di clock, altrimenti torna immediatamente al punto 1.

Per verificare se la codifica è o non è quella da notificate, si introduca una variabile *test* in accordo al seguente schema:

```
wire test; assign test=... ;
```

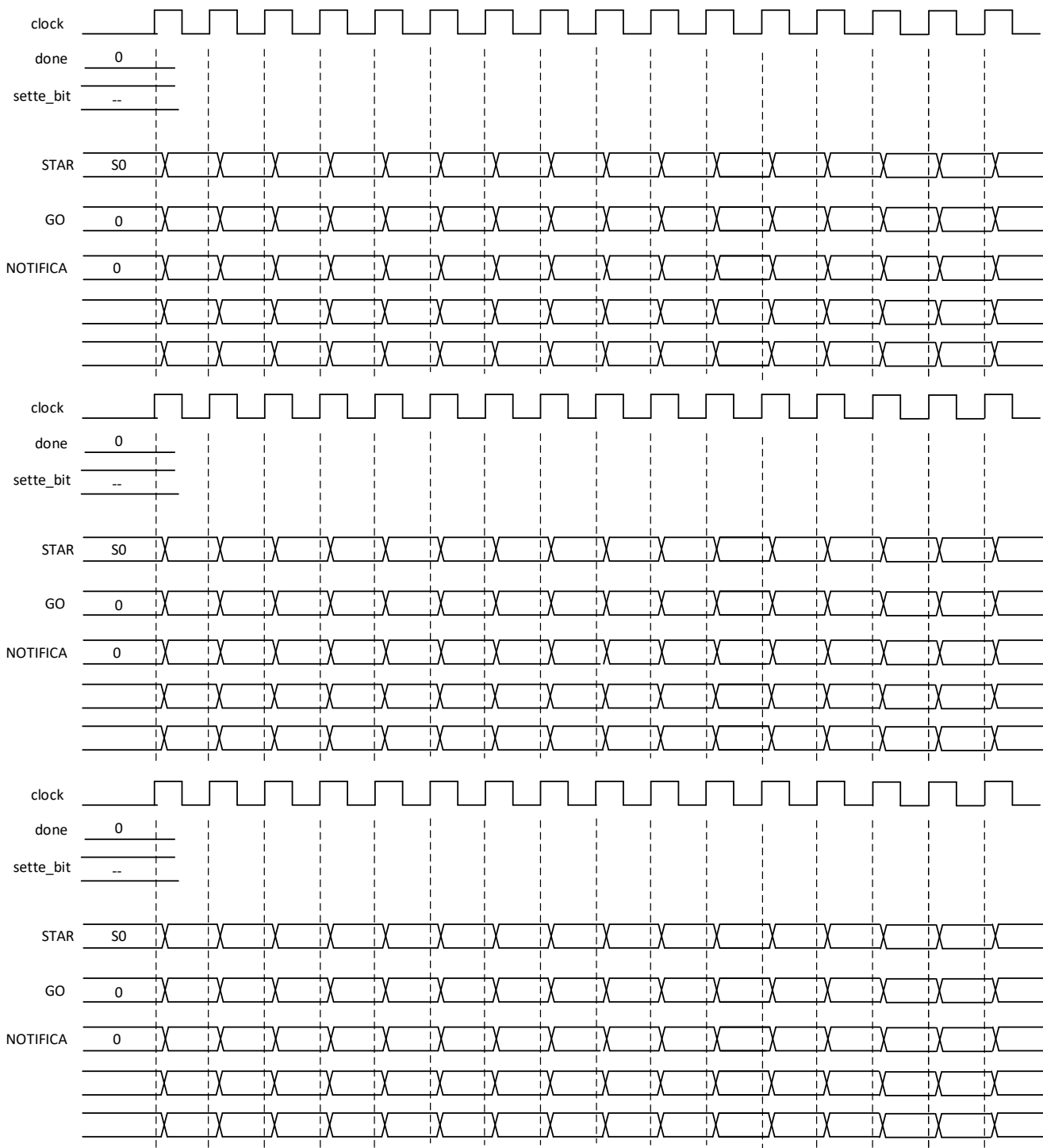
che vale 1 se il test ha successo, 0 altrimenti.

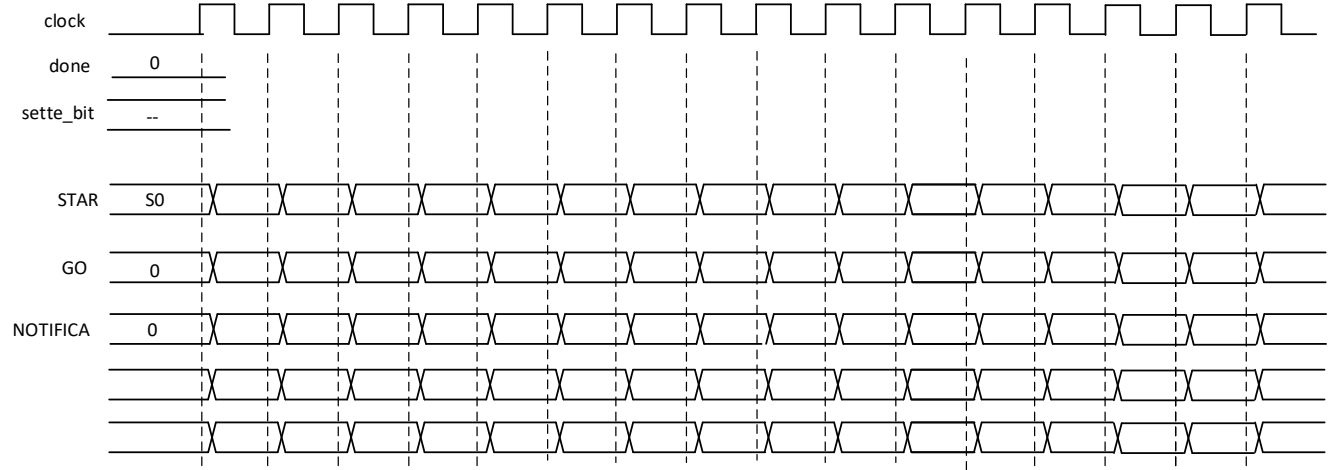
Fare un diagramma temporale che illustri due cicli completi di evoluzione di XXX, supponendo che il Produttore presenti 'B1000111 al primo ciclo e 'B1001000 al secondo ciclo. Affinché il diagramma sia di dimensioni accettabili, supporre anche che la risposta del Produttore, una volta iniziato l'handshake, sia abbastanza veloce (tra uno e due cicli del clock di XXX).

Sintetizzare XXX fornendo l'equazione algebrica di *test*

Handshake: partendo da una condizione in cui *go* e *done* sono a 0, XXX inverte il valore di *go* per chiedere al Produttore un nuovo dato; il Produttore produce ed emette il dato e poi lo notifica invertendo il valore di *done*, e così via all'infinito

Si può usare il retro del testo per i diagrammi temporali (nel caso, ricordarsi di scrivere nome, cognome e matricola nel testo e di consegnarlo).

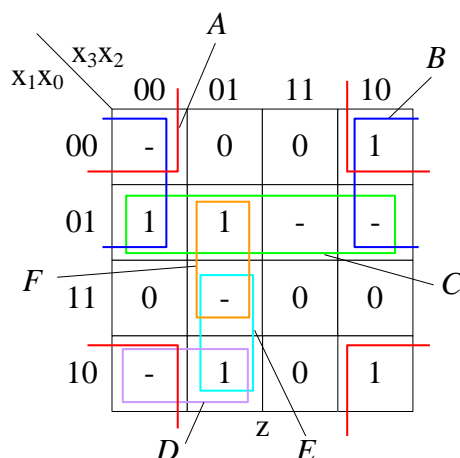




Esercizio 1 – Soluzione

Gli implicanti principali sono:

$$A = \bar{x}_2 \cdot \bar{x}_0, \quad B = \bar{x}_2 \cdot \bar{x}_1, \quad C = \bar{x}_1 \cdot x_0, \quad D = \bar{x}_3 \cdot x_1 \cdot \bar{x}_0, \quad E = \bar{x}_3 \cdot x_2 \cdot x_1, \quad F = \bar{x}_3 \cdot x_2 \cdot x_0.$$



Dalla mappa, in cui sono evidenziati i sottocubi corrispondenti, si può desumere che

- l'implicante A è essenziale,
- i rimanenti implicanti B, C, D, E , ed F sono semplicemente eliminabili.

Le liste di copertura irridondanti principali sono $\{A, C, D\}$, $\{A, C, E\}$, $\{A, B, F, D\}$ e $\{A, B, F, E\}$, cui corrispondono rispettivamente le seguenti forme SP:

1. $z = \bar{x}_2 \cdot \bar{x}_0 + \bar{x}_1 \cdot x_0 + \bar{x}_3 \cdot x_1 \cdot \bar{x}_0$, costo a porte: 4
2. $z = \bar{x}_2 \cdot \bar{x}_0 + \bar{x}_1 \cdot x_0 + \bar{x}_3 \cdot x_2 \cdot x_1$, costo a porte: 4
3. $z = \bar{x}_2 \cdot \bar{x}_0 + \bar{x}_2 \cdot \bar{x}_1 + \bar{x}_3 \cdot x_2 \cdot x_0 + \bar{x}_3 \cdot x_1 \cdot \bar{x}_0$, costo a porte: 5
4. $z = \bar{x}_2 \cdot \bar{x}_0 + \bar{x}_2 \cdot \bar{x}_1 + \bar{x}_3 \cdot x_2 \cdot x_0 + \bar{x}_3 \cdot x_2 \cdot x_1$, costo a porte: 5

Le forme 1 e 2 sono entrambe di costo minimo.

La forma 1 presenta un'alea statica del primo ordine sul livello uno nel passaggio dallo stato d'ingresso 'B?000 a 'B?001 (e viceversa), che può essere eliminata aggiungendo l'implicante B alla lista di copertura, che diventa $\{A, B, C, D\}$

La forma 2 è affetta dalle stesse alee della forma 1, e presenta inoltre un'alea statica del primo ordine sul livello uno nel passaggio dallo stato d'ingresso 'B0111 a 'B0101 (e viceversa), ed un'alea dello stesso tipo nel passaggio dallo stato d'ingresso 'B0010 a 'B0110 (e viceversa). L'eliminazione delle suddette alee richiede l'aggiunta alla lista dei rimanenti implicanti B, D, F .

Esercizio 2 - UNA SOLUZIONE

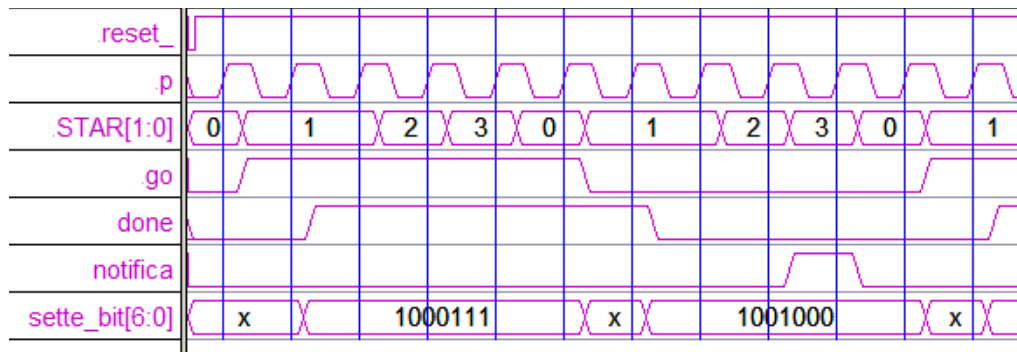
```

module XXX (sette_bit,go,done, notifica, clock,reset_);
input      clock,reset_;
input      done;
output     go,notifica;
input  [6:0] sette_bit;

reg        GO;          assign go=GO;
reg        NOTIFICA;    assign notifica=NOTIFICA;
reg        OLD_DONE;

reg [1:0] STAR; parameter S0=0,S1=1,S2=2,S3=3;
wire test;  assign test=(sette_bit[6:3]=='B1001)?1:0;
always @(reset_==0) #1 begin GO<=0; OLD_DONE<=0; NOTIFICA<=0; STAR=S0; end
always @(posedge clock) if (reset_==1) #3
casex(STAR)
  S0:  begin GO<=~GO; STAR<=S1; end
  S1:  begin STAR<=(done==OLD_DONE)?S1:S2; end
  S2:  begin OLD_DONE<=done; NOTIFICA<=test; STAR<=S3; end
  S3:  begin NOTIFICA<=0; STAR<=S0; end
endcase
endmodule

```



```
test=sette_bit[6]&(~sette_bit[5])&(~sette_bit[4])& sette_bit[3]
```