

### Esercizio 1

Data la seguente mappa:

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

$z$

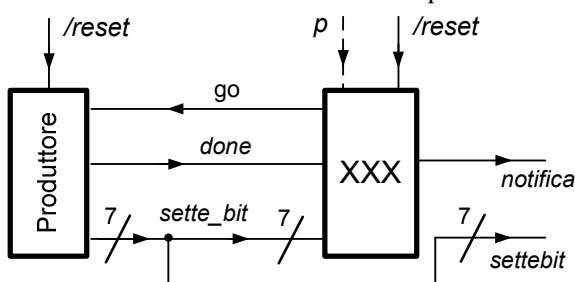
1. indicare e classificare gli implicanti principali;
2. trovare tutte le possibili liste di copertura non ridondanti, ed individuare quelle cui corrispondono forme di tipo SP di costo minimo secondo il criterio di costo a porte;
3. per ognuna delle liste di copertura non ridondanti di costo minimo trovate 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 notificare, 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

