

Cognome

Nome

**ESERCIZIO N°1**

7 punti

Realizzare una subroutine per il microcontrollore AVR XMEGA256A3BU, che valuta il prodotto delle 4 cifre BCD contenute nel registro a 16 bit Y. Il risultato, di cui occorre determinare il range e che deve essere valutato senza errori, deve essere posto nelle locazioni consecutive a partire dalla 0x3ABC (LSByte first)

*Risultato in binario  
cifre non valide → porre tutti 1*

**ESERCIZIO N°2**

5 punti

Realizzare una macchina sequenziale sincrona secondo il modello di Moore con un ingresso e una uscita, in grado di generare la sequenza 101101 nel caso in cui l'ingresso sia 0 e la sequenza 001001 nel caso in cui l'ingresso sia 1. Ogni volta che si ha una commutazione dell'ingresso, la macchina riparte dall'inizio con la sequenza prevista per quell'ingresso. Non è richiesta l'ottimizzazione della codifica degli stati.

**ESERCIZIO N°3**

5 punti

Disegnare lo schema logico di un contatore Johnson modulo 7 con reset, che generi cioè ciclicamente le uscite 000001, 000010, 000100, 0001000, 0010000, 0100000, e 1000000.

## ESERCIZIO N°4

6 punti

- a) Determinare la mappa di Karnaugh di una funzione logica  $Y_{err} = f(X_4, X_3, X_2, X_1, X_0)$  dove  $X_4, X_3, X_2$  e  $X_1$  rappresentano una cifra in codifica BCD, mentre  $X_0$  è un bit di parità (logica parità pari)  $Y_{err}$  vale 1 se la regola di parità non è corretta o se la cifra non rispetta la codifica BCD ( $Y_{err}$  vale 0 altrimenti).
- b) Realizzare con circuito a porte logiche AND, OR, NOT e 2 livelli di logica la funzione del punto a) scegliendo tra diverse possibili soluzioni quella che minimizza numero di porte logiche.
- c) Se porte logiche elementari (AND, OR, NOT) a  $K$  ingressi hanno  $T_{pd} = 0,05 \text{ ns} + 0,25 K \text{ ns}$ , quale è il  $T_{pd}$  massimo del circuito di cui al punto b)? Se ingressi e uscite del circuito combinatorio di cui al punto b) sono registrati con registri aventi  $T_{co} = 0,5 \text{ ns}$ ,  $T_{hold} = 0,3 \text{ ns}$  e  $T_{setup} = 0,5 \text{ ns}$  quale è la massima frequenza di lavoro possibile?

## ESERCIZIO N°5

4 punti

- a) Realizzare la funzione  $Y = f(X_4, X_3, X_2, X_1, X_0)$  di cui all'esercizio 1 tramite multiplexer.
- b) Realizzare la funzione  $Y = f(X_4, X_3, X_2, X_1, X_0)$  di cui all'esercizio 1 tramite decoder.

## ESERCIZIO N°6

6 punti

Dati i numeri  $X = -1128,75$   $Y = 10,125$  e i risultati delle operazioni  $A = -(X + Y)$  e  $B = (X - Y)$

- a) Determinare la loro rappresentazione in virgola fissa e MS, C2, C1, Traslazione e il numero minimo di bit necessario per rappresentarli tutti correttamente.
- b) Se si usa una ALU a 10 bit che opera in C2 si commettono errori di rappresentazione per  $X, Y, A$  e  $B$ ? Se sì, di che entità sono gli errori in valore assoluto e percentuale?
- c) Determinare la rappresentazione di  $X, Y, A$  e  $B$  in virgola mobile formato standard IEEE 754 singola precisione (binary 32). In questo caso, si commettono errori di rappresentazione? Di che entità sono gli eventuali errori in valore assoluto e percentuale?

# 1

Realizzare una subroutine per il microcontrollore AVR XMEGA256A3BU, che valuta il prodotto delle 4 cifre BCD contenute nel registro a 16 bit Y. Il risultato, di cui occorre determinare il range e che deve essere valutato senza errori, deve essere posto nelle locazioni consecutive a partire dalla 0x3ABC (LSByte first). Il risultato deve essere rappresentato in binario. Nel caso di valori non validi come cifre BCD, si pongano nel risultato tutti 1.

```
BCD_product:
  push R0 //salva i registri usati
  push R1
  push R16
  push R17
  push R18

  ser R16
  mov R0,R16
  mov R1,R16 //predispone nel caso ci siano cifre errate

  ldi R16,0x66 //test di validita` delle cifre u e d
  add R16,YL
  brcs end_BCD_product
  brhs end_BCD_product

  ldi R16,0x66 //test di validita` delle cifre c e m
  add R16,YH
  brcs end_BCD_product
  brhs end_BCD_product

  mov R16,YL
  mov R17,YH
  andi R16,0x0F //isola u
  andi R17,0x0F //isola c
  mul R16,R17
  mov R18,R0 //R0=u*c; R1=0

  mov R16,YL
  mov R17,YH
  andi R16,0xF0 //isola 16*d
  andi R17,0xF0 //isola 16*m
  mul R16,R17 //R0=0; R1=m*d

  mul R18,R1 //R1:R0=u*d*c*m

end_BCD_product:
  sts 0x3ABC,R0
  sts (0x3ABC+1),R1 //risultato trasferito in memoria, LSB first

  pop R18 //ripristina i registri usati
  pop R17
  pop R16
  pop R1
  pop R0
  ret
```

2

grafo iniziale

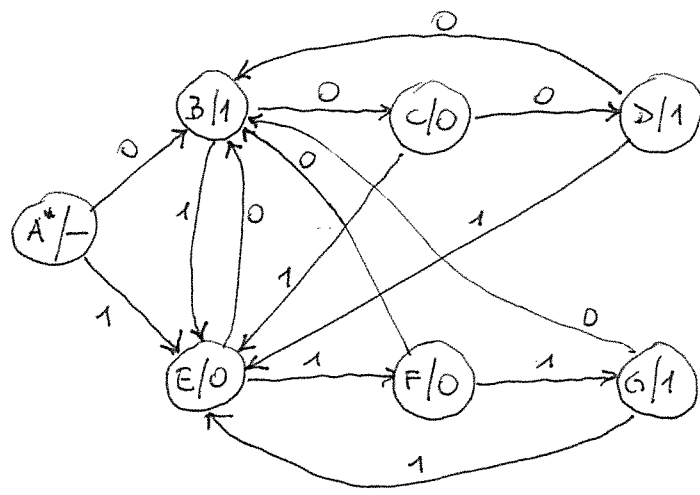
Le sequenze sono le ripetizioni di 2 sequenze di 3 stati.

Sequenze generate

IN  
 0 1 0 1  
 1 0 0 1

A\* stato di reset (000)

uscita don't care



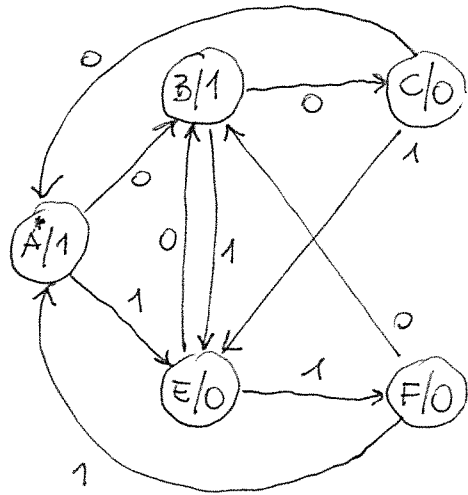
Studio delle equivalenze

B <sub>1</sub>	<del>C B</del> E E					
C <sub>0</sub>	<del>D B</del> E E	<del>C</del>				
D <sub>1</sub>	<del>B B</del> E E	<del>D C</del> E E				
E <sub>0</sub>	<del>B B</del> F E	<del>D D</del> F E	<del>E</del>			
F <sub>0</sub>	<del>B B</del> G G	<del>D D</del> G E	<del>B B</del> G F			
G <sub>1</sub>	<del>B B</del> E E	<del>C</del> E	<del>B B</del> E E	<del>E</del>		
	A <sub>-</sub>	B <sub>1</sub>	C <sub>0</sub>	D <sub>1</sub>	E <sub>0</sub>	F <sub>0</sub>

D ≡ G

A ≡ D ≡ G se l'uscita di A si assume 1

grafo semplificato

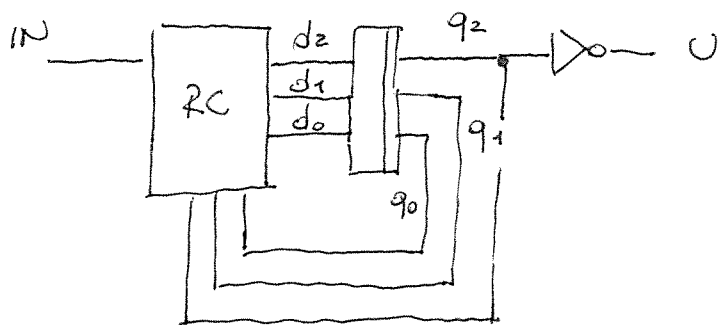


Codifica.

	q <sub>2</sub>	q <sub>1</sub>	q <sub>0</sub>	U
A*	0	0	0	1
B	0	0	1	1
C	1	0	0	0
E	1	1	1	0
F	1	1	0	0

$U = \overline{q_2}$

# Architettura



## Simulazione

		$q_2 \text{ IN}$			
		00	01	11	10
$q_1, q_0$	00	001	111	111	000
	01	100	111	-	-
	11	-	-	110	001
	10	-	-	000	001

0	1	1	0
1	1	-	-
-	-	1	0
-	-	0	0

$$d_2 = IN \bar{q}_1 + \bar{q}_2 q_0 + IN q_0$$

Funzioni effettive

0	1	1	0
1	1	1	0
1	1	1	0
0	0	0	0

0	1	1	0
0	1	-	-
-	-	1	0
-	-	0	0

$$d_1 = IN \bar{q}_1 + IN q_0$$

0	1	1	0
0	1	1	0
0	1	1	0
0	0	0	0

1	1	1	0
0	1	-	-
-	-	0	1
-	-	0	1

$$d_0 = IN \bar{q}_1 + \bar{q}_2 \bar{q}_0 + \bar{IN} q_1$$

1	1	1	0
0	1	1	0
1	0	0	1
1	1	0	1

## Analisi degli stati non specificati del profilo

	codice	stato futuro se $IN=0$	se $IN=1$	
$s_1$	010	001 = B	001 = B	OK
$s_2$	011	101 = $s_3$	110 = F	OK
$s_3$	101	000 = A	111 = E	

Se anche la macchina andasse in uno di questi stati, tornerebbe al normale funzionamento entro i 2 clock successivi

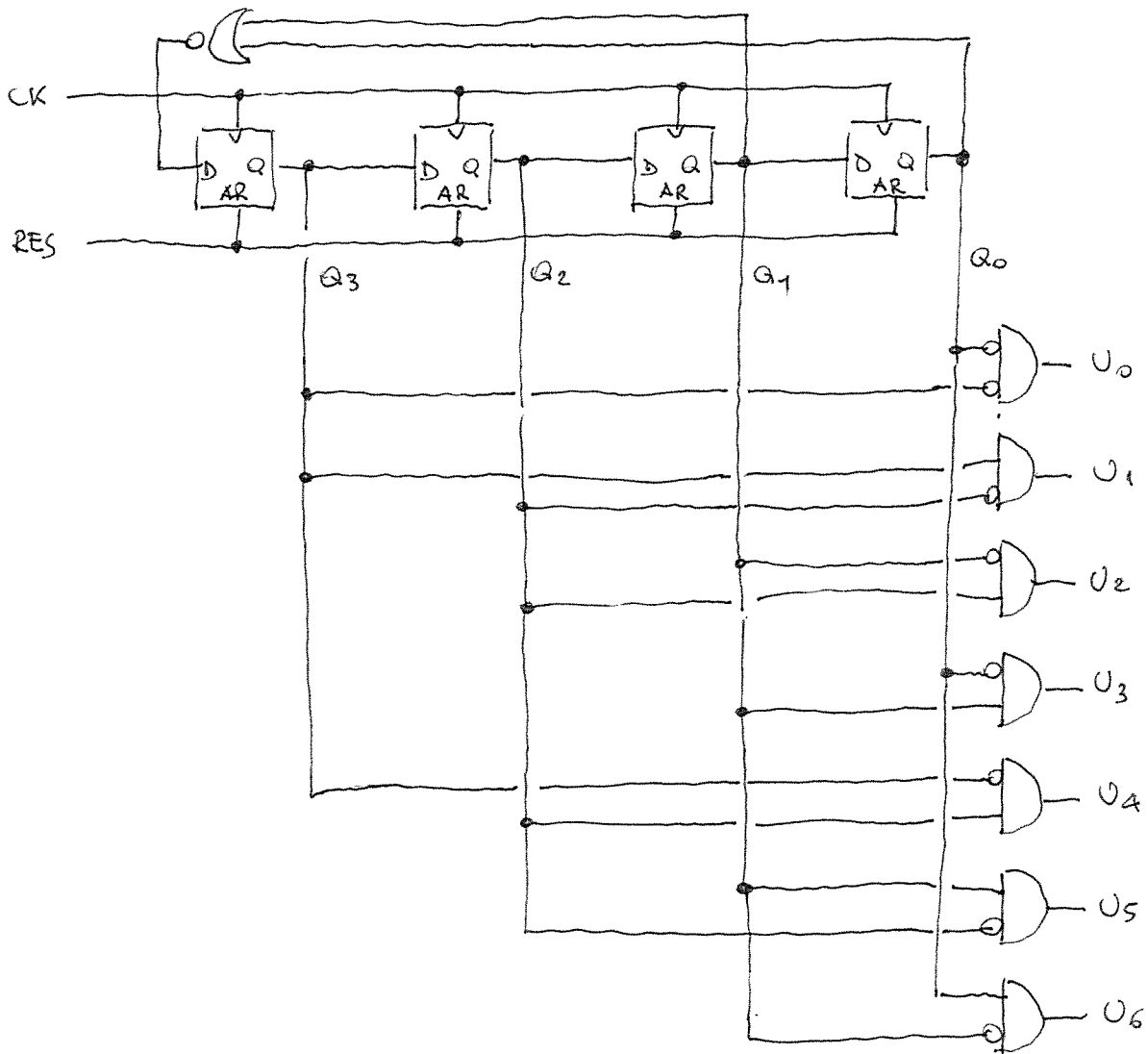
3

sequenza generata (reazione con NOR) a partire dallo stato di reset

	$Q_3$	$Q_2$	$Q_1$	$Q_0$
RES	0	0	0	0
	1	0	0	0
	1	1	0	0
	1	1	1	0
	0	1	1	1
	0	0	1	1
	0	0	0	1
	0	0	0	0

Uscite desiderate						
$U_6$	$U_5$	$U_4$	$U_3$	$U_2$	$U_1$	$U_0$
0	0	0	0	0	0	1
0	0	0	0	0	1	0
0	0	0	0	1	0	0
0	0	0	1	0	0	0
0	0	1	0	0	0	0
0	1	0	0	0	0	0
1	0	0	0	0	0	0

$$U_0 = \overline{Q_3} \overline{Q_0}$$
$$U_1 = Q_3 \overline{Q_2}$$
$$U_2 = Q_2 \overline{Q_1}$$
$$U_3 = Q_1 Q_0$$
$$U_4 = \overline{Q_3} Q_2$$
$$U_5 = \overline{Q_2} Q_1$$
$$U_6 = \overline{Q_1} Q_0$$



④ Mappe di Karnaugh a)

		$x_4 x_3$			
		00	01	11	10
$x_2 x_1$	00	0	1	1	1
	01	1	0	1	0
	11	0	1	1	1
	10	1	0	1	1

$x_0 = 0$

		$x_4 x_3$			
		00	01	11	10
$x_2 x_1$	00	1	0	1	0
	01	0	1	1	1
	11	1	0	1	1
	10	0	1	1	1

$x_0 = 1$

ci devono essere 10 mintermi, metà con parità pari.

b) sintesi a 2 livelli di logica, a minimo numero di letterali PS (tutti mintermi essenziali)

Occorrono 10 OR a 5 ingressi e 1 AND a 10 ingressi, più 5 NOT totale: 50 letterali

SP (tutti essenziali)

$$\begin{aligned}
 & x_4 x_3 + x_4 x_2 + x_4 x_1 x_0 + x_3 x_2 \bar{x}_1 x_0 + x_3 \bar{x}_2 x_1 x_0 + \bar{x}_4 \bar{x}_3 \bar{x}_2 \bar{x}_1 x_0 + \\
 & + \bar{x}_4 \bar{x}_3 x_2 x_1 x_0 + x_3 \bar{x}_2 \bar{x}_1 \bar{x}_0 + x_4 \bar{x}_2 \bar{x}_1 \bar{x}_0 + \bar{x}_4 \bar{x}_3 \bar{x}_2 x_1 \bar{x}_0 + x_3 x_2 x_1 \bar{x}_0 + \\
 & + \bar{x}_3 x_2 \bar{x}_1 \bar{x}_0
 \end{aligned}$$

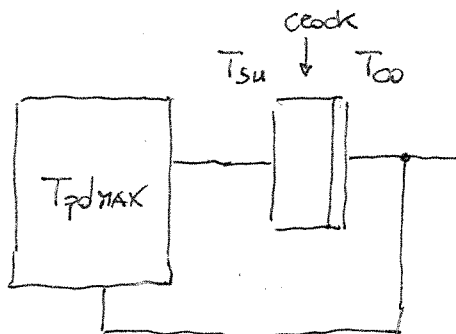
totale: 46 letterali

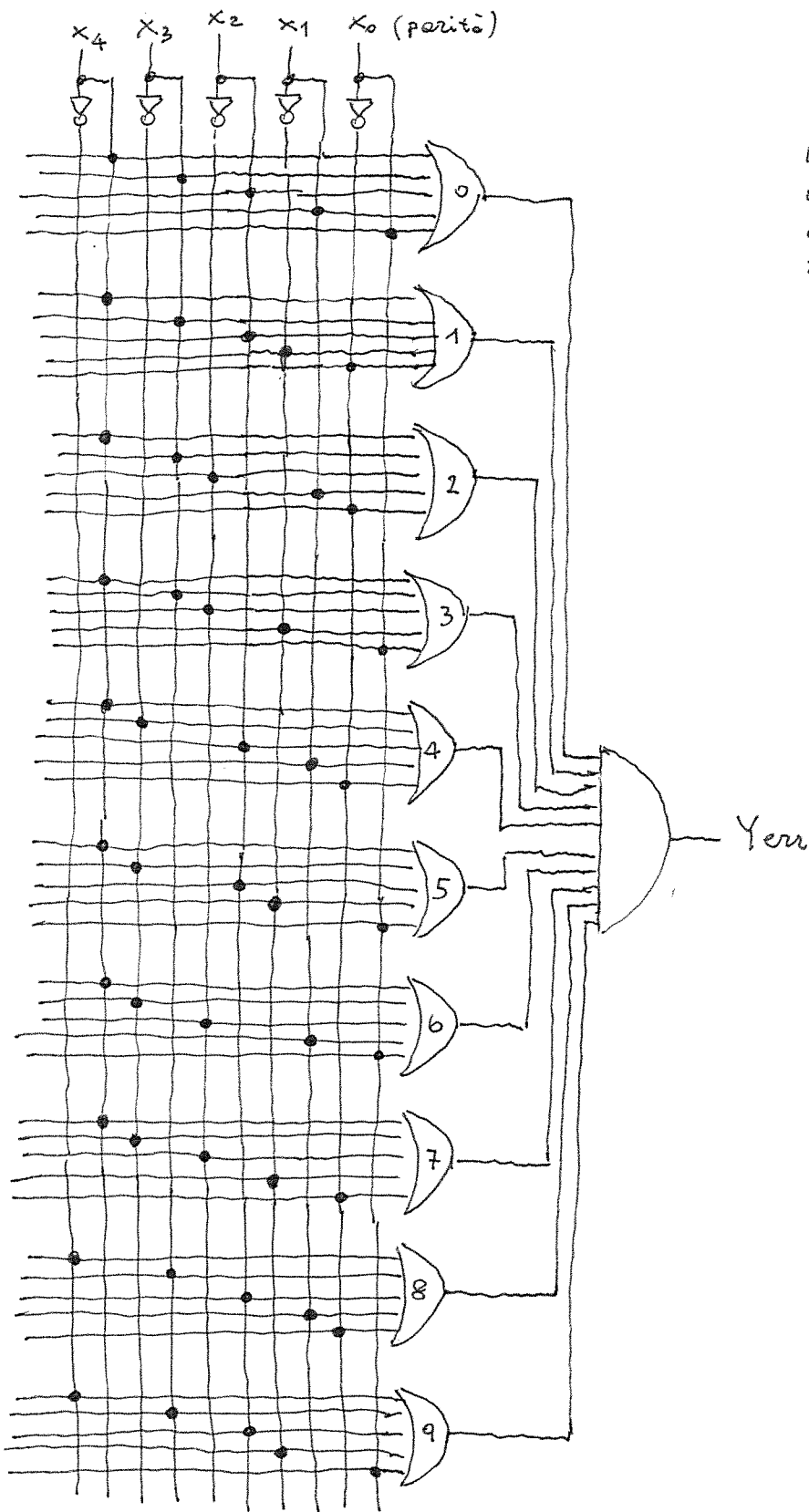
Per avendo meno letterali, la forma SP richiede più porte (13 + 5 not), privilegiamo quindi la forma PS (a pagine seguenti)

c)  $T_{NOT} = 0,30 \text{ ns}$      $T_{OR5} = 1,30 \text{ ns}$      $T_{AND10} = 2,55 \text{ ns}$

$T_{pdMAX} = T_{NOT} + T_{OR5} + T_{AND10} = 4,15 \text{ ns}$

$f_{max} = \frac{1}{T_{co} + T_{pdMAX} + T_{su}} = 194,17 \text{ MHz}$  (non ci va  $T_{hold}$ !)



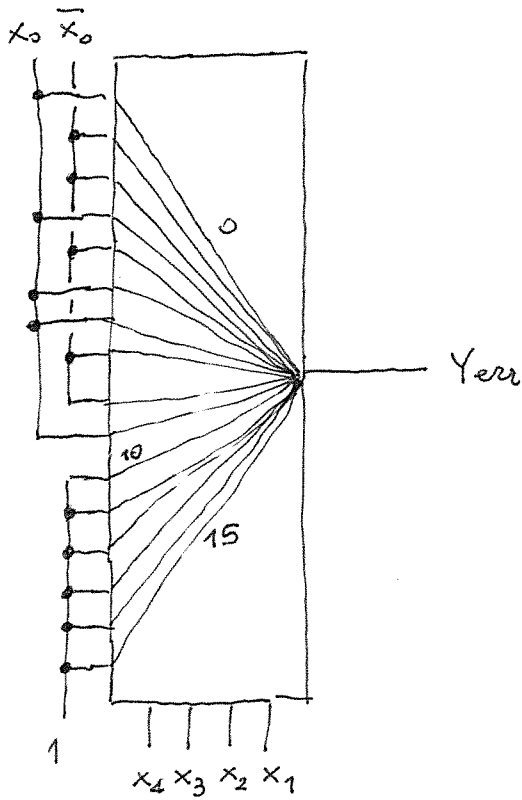


Le OR selezionano i 10  
 maxtermini corrispondenti  
 alle cifre BCD con la parità  
 pari.

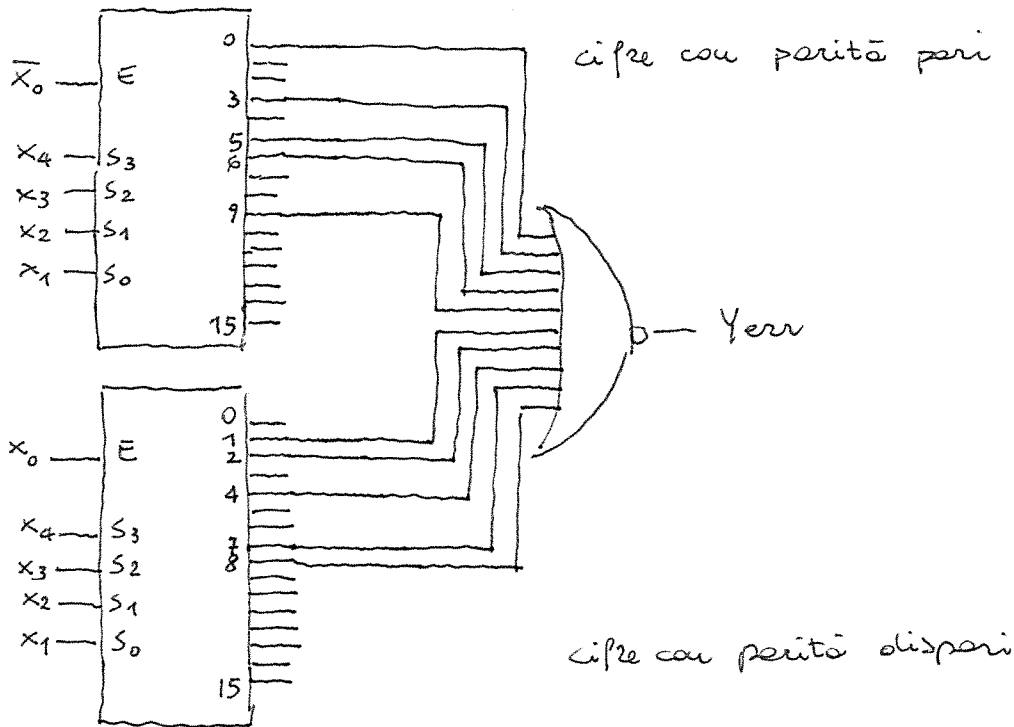


⑤ Realizzazione con 40X (16:1) 2)

L'uso di un 32:1 permette di evitare un invertor (a un costo maggiore, però...)



Realizzazione con 2 decoder 4:16 con abilitazione e porte NOR  
 Un decoder 5:32 permette una realizzazione diretta



⑥ Numeri da rappresentare

-1128,75      10,125      1118,625      -1138,875

a) Le parti frazionarie possono essere rappresentate senza errore su 3 bit (6/8 ; 1/8 ; 5/8 ; 7/8)

Le parti intere al segno richiedono 12 bit in tutte le rappresent. indicate. In totale occorrono 15 bit

-1128,75	MS	1:10001101000,110
	C1	101110010111,001
	C2	101110010111,010
	T	001110010111,010

10,125	MS	0:00000001010,001
	C1	000000001010,001
	C2	000000001010,001
	T	100000001010,001

1118,625	MS	0:10001011110,101
	C1	010001011110,101
	C2	010001011110,101
	T	110001011110,101

-1138,875	MS	1:10001110010,111
	C1	101110001101,000
	C2	101110001101,001
	T	001110001101,001

b) Usando 10 bit (eliminando i 5 bit MENO significativi) si introduce errore. In C2 si ha (caso TRONCAMENTO)

-1128,75	1011100101 00,000	$\epsilon_A = 3,25$	$\epsilon_r = 0,28\%$
10,125	0000000010 00,000	$\epsilon_A = 2,125$	$\epsilon_r = 20,99\%$
1118,625	0100010111 00,000	$\epsilon_A = 2,625$	$\epsilon_r = 0,235\%$
-1138,875	1011100011 00,000	$\epsilon_A = 1,125$	$\epsilon_r = 0,099\%$

c) la notazione IEEE 754, disponendo di un numero di cifre significative maggiore di 15, rappresenta senza errori i valori eseguiti.

$$x = (-1)^s 2^{e-127} \left\{ 1 + \sum_{i=1}^{23} b_{23-i} 2^{-i} \right\}$$

$$x = -1128,75 = (-1)^1 2^{10} (1 + 858112 \cdot 2^{-23})$$

$$s \quad e=137$$

$$[1|10001001|000110100011000000000000]$$

$$y = 10,125 = (-1)^0 2^3 (1 + 2228224 \cdot 2^{-23})$$

$$s \quad e=130$$

$$[0|10000010|010001000000000000000000]$$

$$A = 1118,625 = (-1)^0 2^{10} (1 + 775168 \cdot 2^{-23})$$

$$s \quad e=137$$

$$[0|10001001|000101111010100000000000]$$

$$B = -1138,875 = (-1)^1 \cdot 2^{10} (1 + 941056 \cdot 2^{-23})$$

$$s \quad e=137$$

$$[1|10001001|000111001011100000000000]$$