SCHEDA **ASE1505** | Data: **30 Giugno 2015**

Cognome | Nome

ESERCIZIO N°1

6 punti

a) Determinare la mappa di Karnaugh di una funzione logica $Y = f(X_4, X_3, X_2, X_1, X_0)$ con i seguenti set di mintermini che danno 1 (S_{on}) e don't care (S_{dc})
$S_{on} = \{0, 1, 6, 7, 14, 15\}$, $S_{dc} = \{4, 5, 16, 17, 30, 31\}$

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 il numero di porte logiche.

c) Se porte logiche elementari (AND, OR, NOT) a K ingressi hanno $T_{pd} = 0{,}05$ ns + 0,1 K 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 $T_{co} = 0.4$ ns, $T_{hold} = 0.3$ ns e $T_{setup} = 0.4$ ns, quale è la massima frequenza di lavoro possibile?

ESERCIZIO N°2

4 punti

a) Realizzare la funzione $Y = f(X_4, X_3, X_2, X_1, X_0)$ di cui all'esercizio 1 tramite decoder e porte 3-state

b) Realizzare la funzione $Y = f(X_4, X_3, X_2, X_1, X_0)$ di cui all'esercizio 1 tramite decoder e porte OR

ESERCIZIO N°3

6 punti

Dati i numeri $A = +130{,}25$ $B = -1{,}625$ e $C = -32{,}125$

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) Determinare la rappresentazione di A, B e C in virgola mobile in formato standard IEEE 754 singola precisione. Si commettono errori di rappresentazione? Se sì, di che entità sono gli errori in valore assoluto e percentuale?

c) Se si usa una ALU a 8 bit che opera in MS, si commettono errori di rappresentazione per A, B e C? Se sì, di che entità sono gli errori in valore assoluto e percentuale?

ESERCIZIO N°4

4 punti

Avendo a disposizione chip di memoria SRAM da 4 M x 3 (costo 0,30 €) e da 8 M x 2 (costo 0,31 €), progettare un modulo di memoria da 8 M x 11 a costo minimo.

ESERCIZIO N°5

5 punti

Disegnare e minimizzare il grafo di una macchina di Moore con 1 ingresso X e una uscita U che viene posta a 1 per un ciclo di clock nel caso in cui si presenti una delle seguenti sequenze non interallacciate: 0011, 0110 e 1010. La macchina deve essere dotata di reset asincrono che la riporta nello stato iniziale, con uscita nulla. Codificare gli stati e disegnare l'architettura della macchina.

ESERCIZIO N°6

8 punti

Realizzare una subroutine per il microcontrollore AVR XMEGA256A3BU, che valuti il prodotto tra un intero senza segno contenuto in R16 e un numero intero, anch'esso senza segno, costituito da 3 byte e contenuto in memoria a partire (LSB) dalla locazione puntata da X. Il risultato deve essere sommato al numero di 4 byte contenuto in memoria a partire (LSB) dalla locazione puntata da Y.

(1) Le mappe sono

$x_4 = 0$

x_1x_0 \ x_3x_2	00	01	11	10
00	1 (0)	— (4)	0 (12)	0 (8)
01	1 (1)	— (5)	0 (13)	0 (9)
11	0 (3)	1 (7)	1 (15)	0 (11)
10	0 (2)	1 (6)	1 (14)	0 (10)

$x_4 = 1$

x_1x_0 \ x_3x_2	00	01	11	10
00	— (16)	0 (20)	0 (28)	0 (24)
01	— (17)	0 (21)	0 (29)	0 (25)
11	0 (19)	0 (23)	— (31)	0 (27)
10	0 (18)	0 (22)	— (30)	0 (26)

Esaminiamo le possibili soluzioni ottime (minimo n° letterali) in forma normale SP e PS

SP: $U = \bar{x}_4 \bar{x}_3 \bar{x}_1 + \bar{x}_4 x_2 x_1$ — 6 letterali, 3 NOT, 2 AND3, 1 OR2

PS: $U = \bar{x}_4 \cdot (\bar{x}_3 + x_1)(x_2 + \bar{x}_1)$ — 5 letterali, 3 NOT, 2 OR2, 1 AND3

L'analisi delle due soluzioni, equivalenti rispetto al criterio del minimo numero di porte, mostra che i "don't care" sono stati gestiti allo stesso modo

$\{4, 5\} \rightarrow 1 \qquad \{16, 17, 30, 31\} \rightarrow \phi$

Tutti gli implicanti/implicati scelti risultano essenziali, tranne $(\bar{x}_3 + x_1)$ che però è il modo più conveniente per completare la copertura degli ϕ di Y.

Analisi del T_{pd} delle reti proposte:

SP: $T_{pd} = \overset{\text{NOT}}{(0{,}05 + 0{,}1)} + \overset{\text{AND}}{(0{,}05 + 0{,}3)} + \overset{\text{OR}}{(0{,}05 + 0{,}2)} = 0{,}75\,ns$

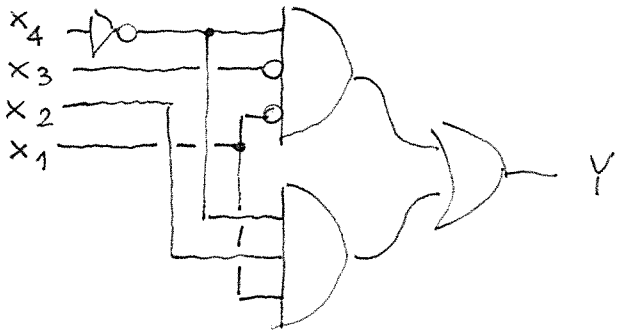
PS: $T_{pd} = \overset{\text{NOT}}{(0{,}05 + 0{,}1)} + \overset{\text{OR}}{(0{,}05 + 0{,}2)} + \overset{\text{AND}}{(0{,}05 + 0{,}3)} = 0{,}75\,ns$

Quindi le reti sono equivalenti anche come ritardo. In una macchina a stati si avrebbe

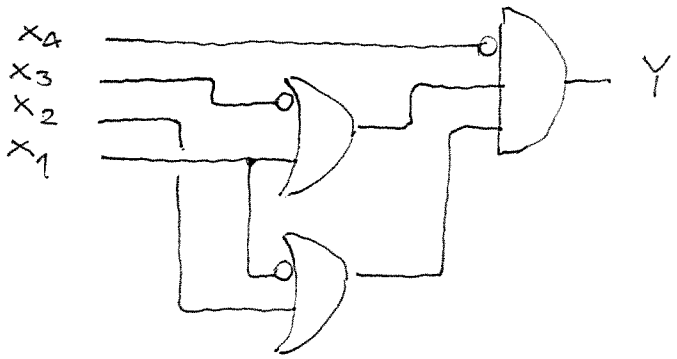
$$f_{MAX} = \frac{1}{T_{pd} + T_{su} + T_{co}} = 645\ MHz$$

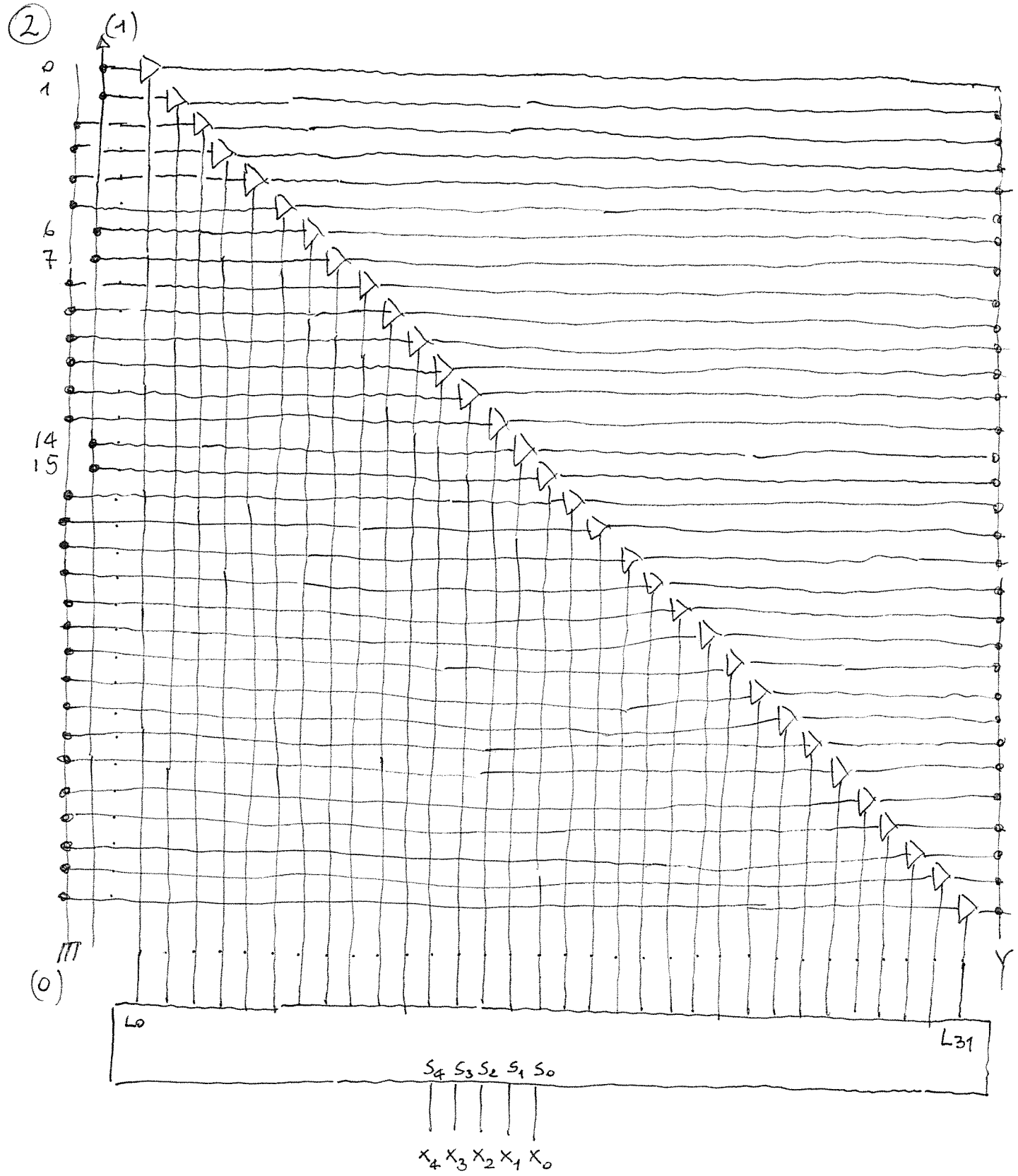
Schema logica

SP



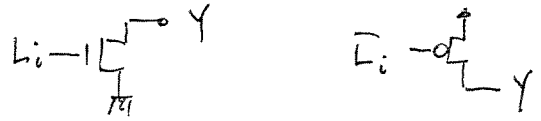
PS



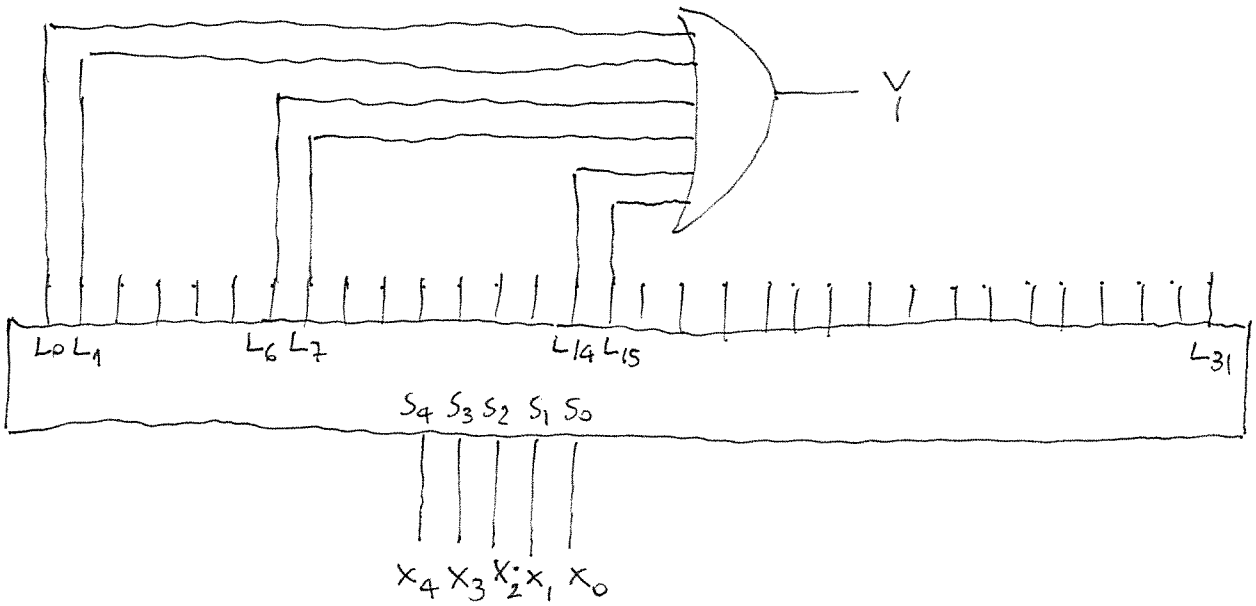


le linee "don't care" possono essere collegate a scelta a Ø o 1.

Si può osservare che una tri-state con ingresso collegato a Ø/1 è implementata con un solo ramo:



Soluzione con porta OR (in questo caso conviene limitarsi strettamente alle linee corrispondenti agli 1)



(3) Visti i valori proposti, si osserva che sono necessari 9b per rappresentare correttamente le parti intere (con segno) in tutte le rappresentazioni proposte e 3b per le parti frazionali, che sono esprimibili senza errore in multipli di 1/8.

In totale servono quindi 12b.

A: +130,25

MS: 0|10000010.010

C2: 01000010.010

C1: 01000010.010

T: 11000010.010

B: -1,625

MS: 1|00000001.101

C2: 11111110.011

C1: 11111110.010

T: 01111110.011

C: -32.125

MS: 1|00100000.001

C2: 11011111.111

C1: 11011111.110

T: 01011111.111

Legge di rappresentazione IEEE 754 (binary 32)

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

$$A = 130{,}25 = (-1)^0 \cdot 2^7 (1 + 147456 \cdot 2^{-23})$$

s e=134 B=147456

[0 | 1 0 0 0 0 1 1 0 | 0 0 0 . 0 0 1 0 . 0 1 0 0 . 0 0 0 0 . 0 0 0 0 . 0 0 0 0]

$$B = -1{,}625 = (-1)^1 \, 2^0 \, (1 + 5242880 \cdot 2^{-23})$$

s e=127 B=5242880

[1 | 0 1 1 1 1 1 1 1 | 1 0 1 . 0 0 0 0 . 0 0 0 0 . 0 0 0 0 . 0 0 0 0 . 0 0 0 0]

$$C = -32{,}125 = (-1)^1 \, 2^5 \, (1 + 32768 \cdot 2^{-23})$$

s e=132 B=32768

[1 | 1 0 0 0 0 1 0 0 | 0 0 0 . 0 0 0 0 . 1 0 0 0 . 0 0 0 0 . 0 0 0 0 . 0 0 0 0]

In tutti e 3 i casi NON si commettono errori di rappresentazione

Nel caso della ALU (MS) a 8b, per rappresentare i tre numeri occorre rinunciare ai 4b meno significativi, arrotondando per ridurre l'errore.

Si ottiene per le rappresentazioni arrotondate (2x)

A	0 | 1 0 0 0 0 0 1	+130	$\varepsilon_A = 0{,}25$	$\varepsilon_r = 0{,}19\%$
B	1 | 0 0 0 0 0 0 1	−2	$\varepsilon_A = 0{,}375$	$\varepsilon_r = 23\%$
C	1 | 0 0 1 0 0 0 0	−32	$\varepsilon_A = 0{,}125$	$\varepsilon_r = 0{,}39\%$

(4)

Considero : due moduli da 8M word

A: 8M×3 0,60€ (costo per 8Mb : 0,20€)

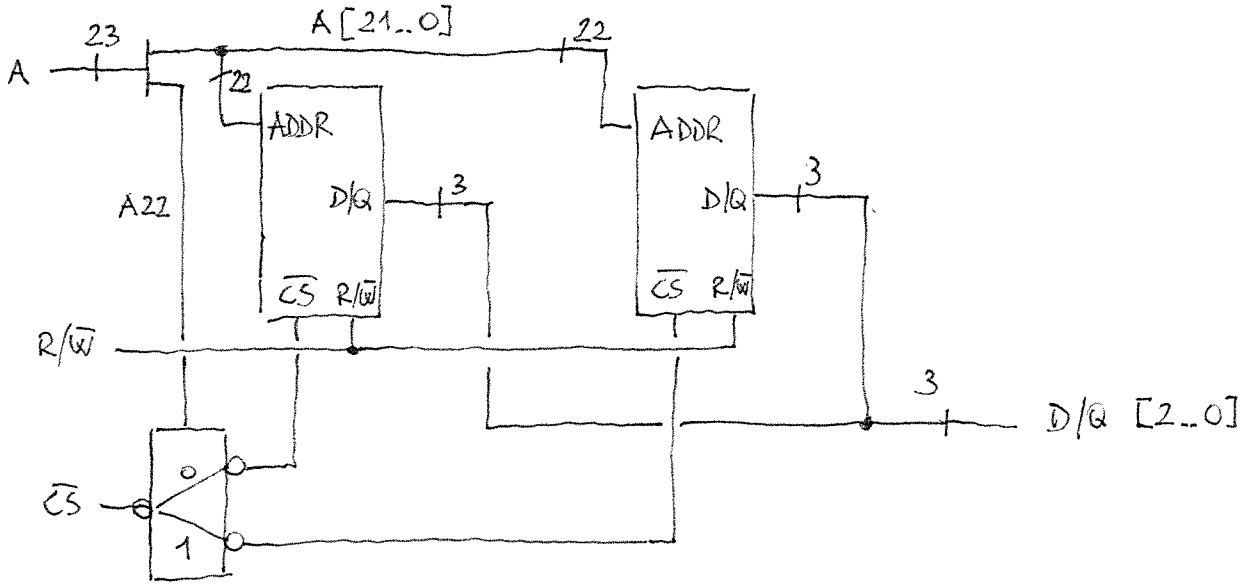
B: 8M×2 0,31€ (costo per 8Mb : 0,155€)

Considero le due soluzioni che massimizzano l'uso del modulo più economico B

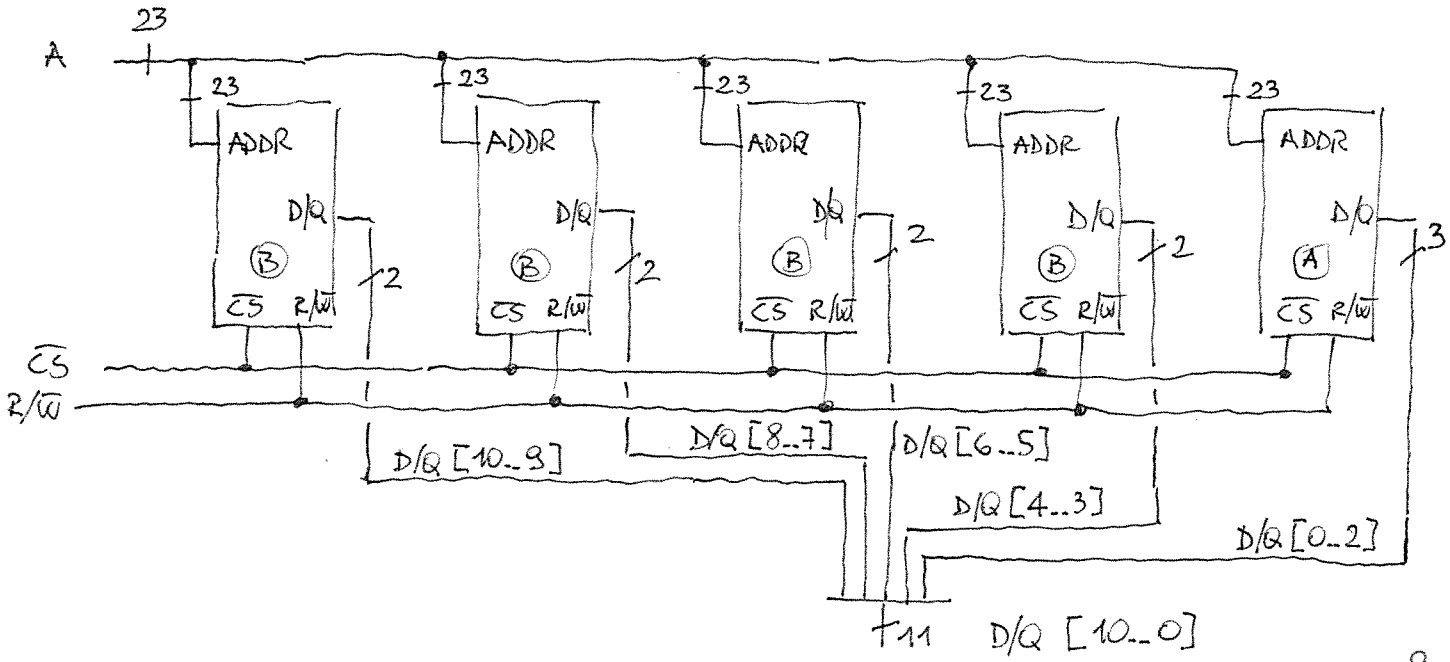
4B+A : 1,84€ ← Soluzione (senza spreco di bit) economicamente più vantaggiosa

6B : 1,86€

Montaggio per la costituzione del modulo A con 2 SRAM da 4M×3

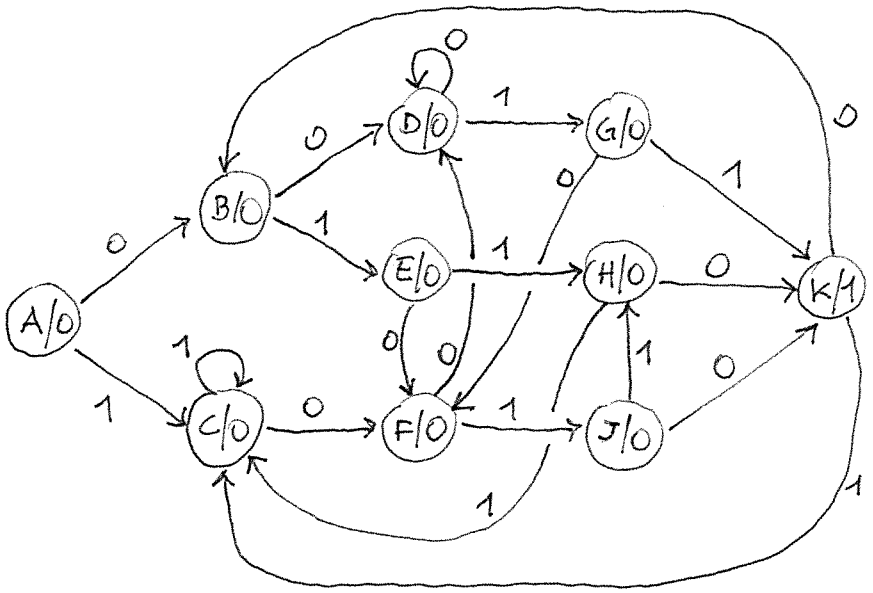


Montaggio finale



(5) Uso D-FF con Clear asincrono
Codifico lo stato iniziale con tutte le variabili di stato ϕ.

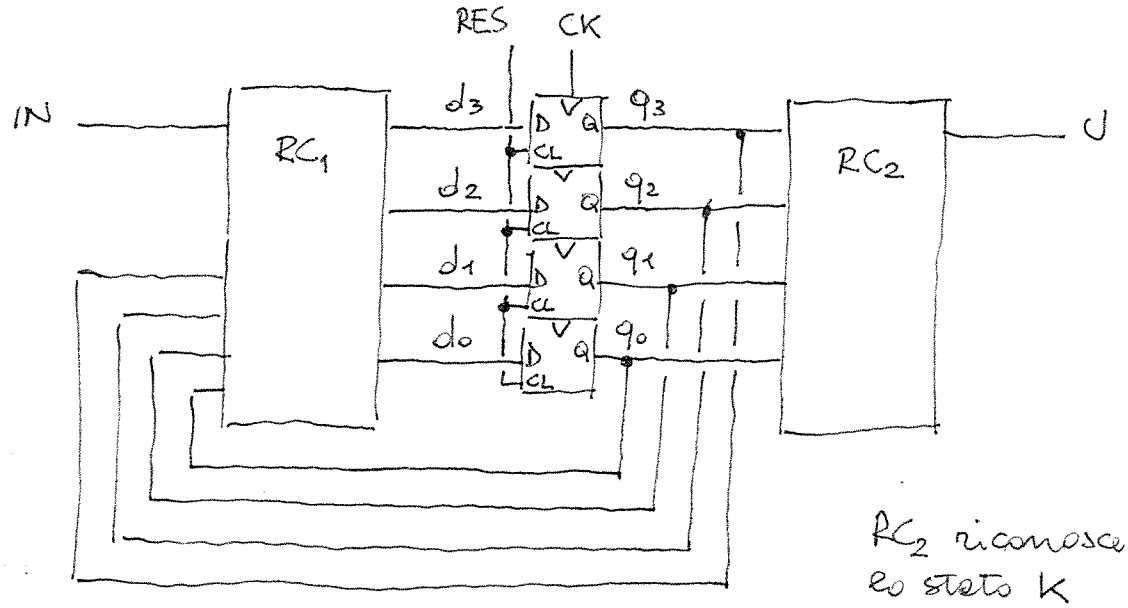
Grafo (seq: 0011, 0110, 1010 NON intersecciate)



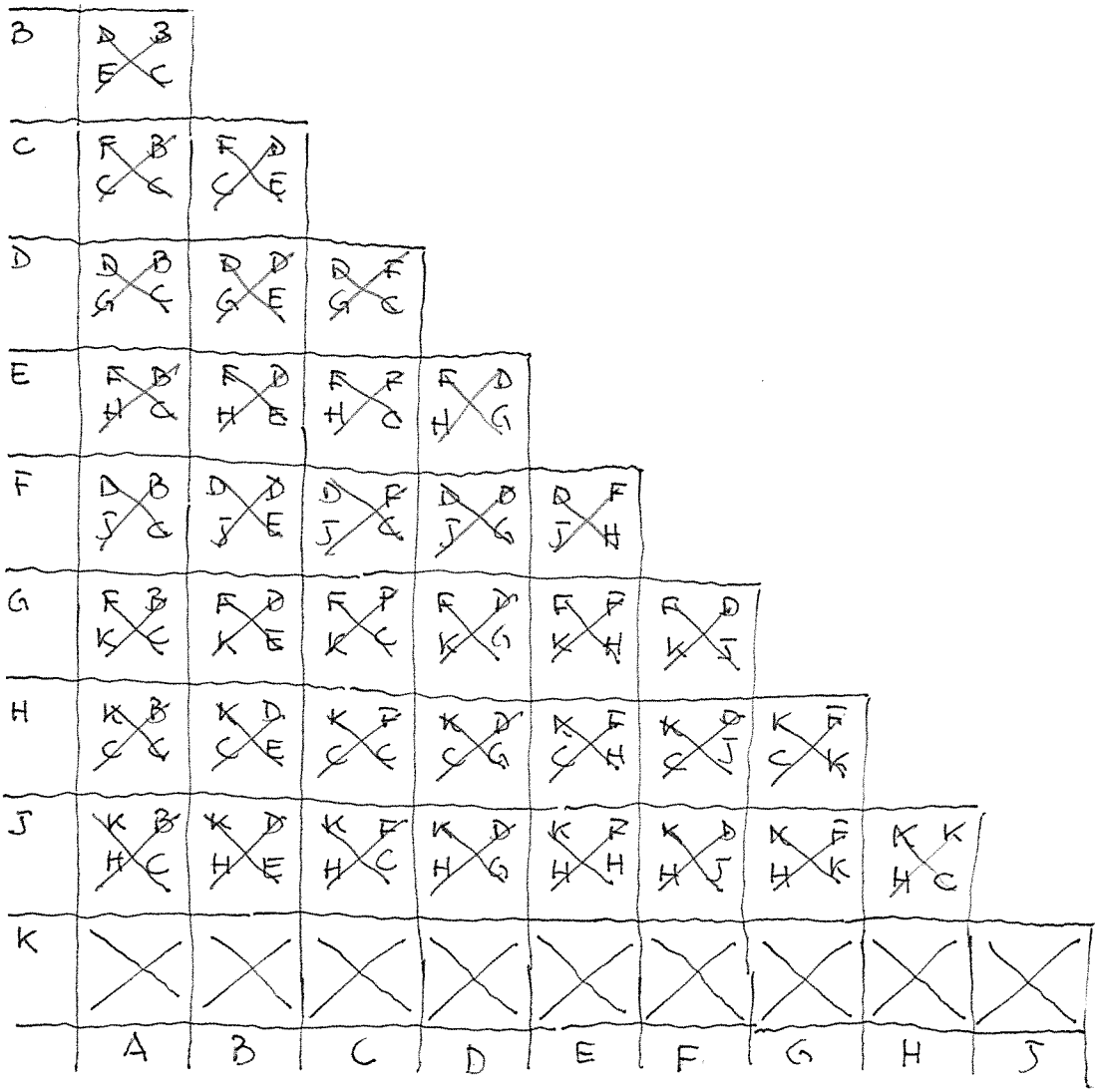
Voluto se nel grafo sono presenti stati equivalenti, tenendo conto dell'uscita (K non può essere equivalente a nessun altro) e degli stati futuri a parità di ingresso.

Matrice delle implicazioni (pag. seguente): NON ci sono stati equivalenti

Architettura (non è richiesta la sintesi delle reti)



Valutiamo la presenza di stati equivalenti, con la tabella delle implicazioni



Si parte constatando che K (uscita 1) non può essere equivalente a nessun altro stato e poi si eliminano a catena stati che portano (per uguale ingresso) a stati NON equivalenti

Quindi il grafo proposto NON può essere ulteriormente ridotto.

Codifica degli stati

Linea guida 1: predecessori di uno stesso stato a parità di ingresso → Adiacenti

Stato	$S^{-1}(0)$	$S^{-1}(1)$	$S^{+1}(0,1)$	Codifica proposta
A	–	–	B, C	0000 (reset)
B	A, K	–	D, E	0110
C	–	A, C, H, K	F, C	0001
D	D, B	F	D, G	1110
E	–	B	F, H	0011
F	C, E, G	–	D, J	1001
G	–	D	F, K	0101
H	–	E, J	K, C	0010
J	–	F	K, H	1010
K	H, J	G	B, C	0100

Linea guida 2: stati successivi di uno stesso stato per ingressi adiacenti (qui c'è solo un ingresso!) → adiacenti

Linea guida 3: stati con le stesse uscite → adiacenti
Nel nostro caso questa linea guida non dà indicazioni, in quanto tutti gli stati (tranne K) hanno uscita Ø

6

Realizzare una subroutine per il microcontrollore AVR XMEGA256A3BU, che valuti il prodotto tra un intero senza segno contenuto in R16 e un numero intero, anch'esso senza segno, costituito da 3 byte e contenuto in memoria a partire (LSB) dalla locazione puntata da X. Il risultato deve essere sommato al numero di 4 byte contenuto in memoria a partire (LSB) dalla locazione puntata da Y.

```
/* La soluzione proposta esegue il prodotto in modo diretto tra R16 e il numero
di 3 byte puntato da X e lo somma al numero di 4 byte puntato da Y
*/


mb_mult:
 push R0
 push R1 //R1:R0 usati dalla MUL
 push R2 //azzerato per usare la ADC
 push R18
 push R19 //di appoggio per caricare il dato da moltiplicare
 push R20
 push R21
 push R22
 push R23 //di appoggio per il risultato della moltiplicazione
 clr R2
 ld R18,X+
 mul R16,R18
 mov R20,R0
 mov R21,R1 //in R21:R20 il prodotto di R16 col byte meno significativo
 ld R18,X+
 ld R19,X
 mul R16,R19
 mov R22,R0
 mov R23,R1 //in R23:R22 il prodotto di R16 col byte più significativo
 mul R16,R18
 add R21,R0
 adc R22,R1
 adc R23,R2 //prodotto completo in R23:R20
 sbiw XH:XL,2 //ripristina X
 ld R18,Y
 add R18,R20
 st Y+,R18
 ld R18,Y
 adc R18,R21
 st Y+,R18
 ld R18,Y
 adc R18,R22
 st Y+,R18
 ld R18,Y
 adc R18,R23
 st Y,R18
 sbiw YH:YL,3 //ripristina Y
 pop R23 //ripristina i registri salvati
 pop R22
 pop R21
 pop R20
 pop R19
 pop R18
 pop R2
 pop R1
 pop R0
 ret
```