SCHEDA **ASE1506**	Data: **21 Luglio 2015**
Cognome	Nome

ESERCIZIO N°1

5 punti

a) Determinare la mappa di Karnaugh di una funzione logica $Y = f(X_3, X_2, X_1, X_0)$ dove X_3, X_2, X_1, X_0 rappresentano un numero in C1. Y vale 1 se la cifra rappresentata è pari.

b) Cambia qualcosa se la rappresentazione è in C2? E in Modulo e segno? Giustificare la risposta.

c) 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.

d) Se porte logiche elementari (AND, OR, NOT) a K ingressi hanno $T_{pd} = 0{,}1$ ns + $0{,}2\ K$ ns, quale è il T_{pd} massimo del circuito di cui al punto c)? Se ingressi e uscite del circuito combinatorio di cui al punto c) sono registrati con registri aventi $T_{co} = 0{,}2$ ns, $T_{hold} = 0{,}1$ ns e $T_{setup} = 0{,}2$ ns quale è la massima frequenza di lavoro possibile?

ESERCIZIO N°2

5 punti

a) Realizzare un decoder 5 to 32 con enable a partire da decoder 2 to 4 con enable. Se il T_{pd} dei decoder 2 to 4 è 3 ns quanto vale il T_{pd} del decoder 5 to 32?

b) Realizzare un mux 16 to 1 a partire da mux 2 to 1. Se il T_{pd} dei mux 2 to 1 è 2 ns quanto vale il T_{pd} del mux 16 to 1?

c) Realizzare il mux 16 to 1 tramite decoder e porte 3-state.

ESERCIZIO N°3

6 punti

Dati i numeri $A = -130{,}25$ $B = 2{,}625$ e $C = -55{,}125$

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) Se si usa una ALU a 8 bit che opera in C2, si commettono errori di rappresentazione per A, B e C? Se sì, di che entità sono gli errori in valore assoluto e percentuale?

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

ESERCIZIO N°4

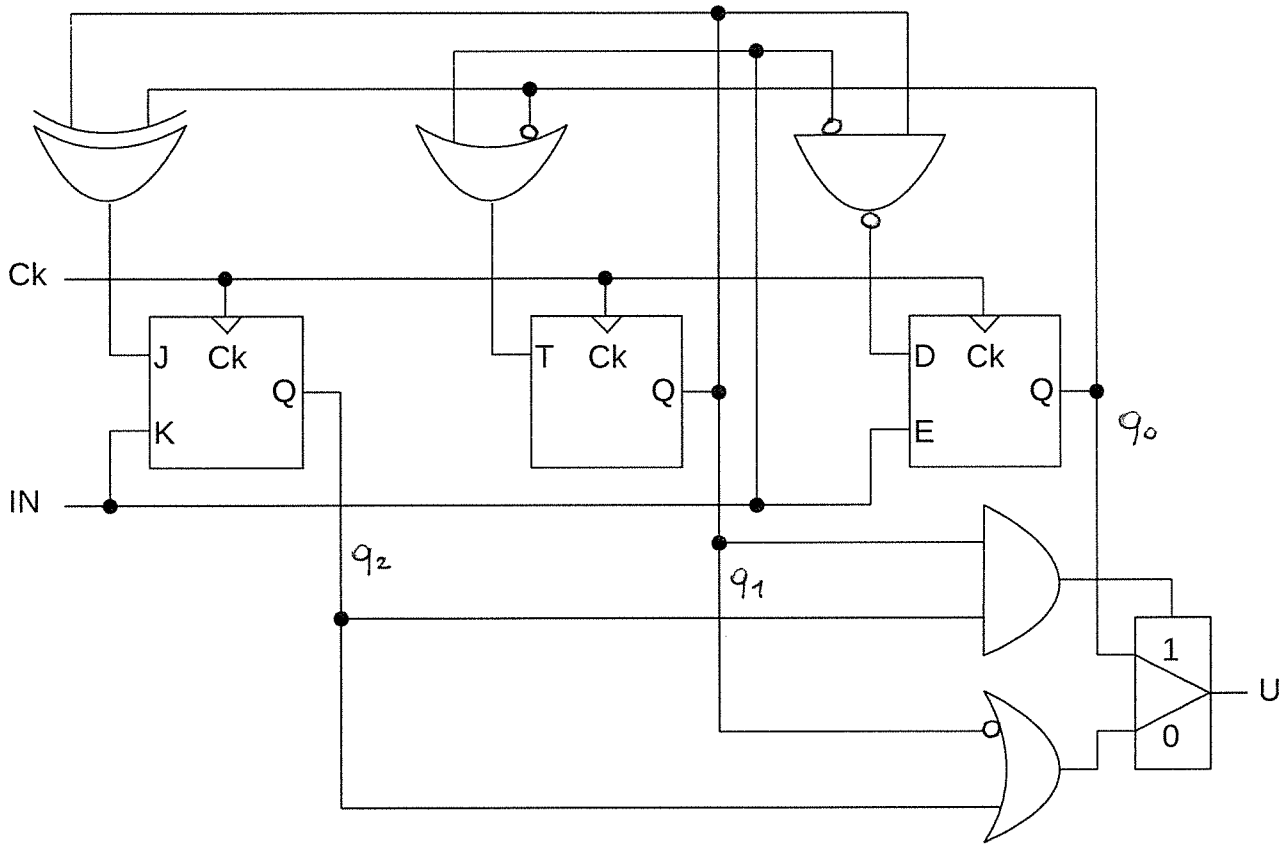
3 punti

Disegnare lo schema logico, usando T-FF, di un contatore modulo 12.

ESERCIZIO N°5

6 punti

Individuare la tipologia architetturale e ricostruire il grafo di flusso della seguente macchina sequenziale sincrona.



ESERCIZIO N°6

8 punti

Realizzare una subroutine per il microcontrollore AVR XMEGA256A3BU, che converta in binario il numero di 4 cifre BCD contenuto in X, ponendo il risultato in Y. Nel caso in cui il numero di partenza non sia valido, in Y deve essere messo il valore esadecimale 0xFFFF.

(1)

a) Mappa con valore numerico C_1

x_1x_0 \ x_3x_2	00	01	11	10
00	1^{0}	1^{4}	0^{-3}	0^{-7}
01	0^{1}	0^{5}	1^{-2}	1^{-6}
11	0^{3}	0^{7}	1^{0}	1^{-4}
10	1^{2}	1^{6}	0^{-1}	0^{-5}

(in piccolo il valore rappresentato)

b) Confronto con C_2 e MS

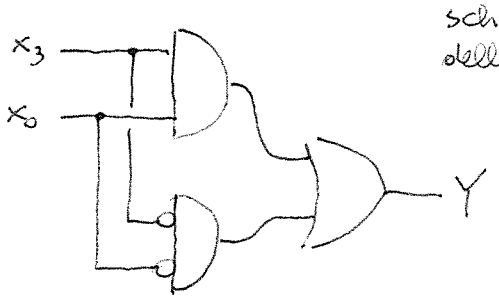
La funzione è diversa, poiché sia in C_2 sia in MS è il solo bit x_0 che determina se il valore rappresentato è pari o dispari, ovvero

$$Y = \bar{x}_0 \qquad \text{(tutti i pari hanno } x_0 = \emptyset\text{)}$$

c) La sintesi a due livelli di logica permette di ottenere due soluzioni SP e PS equivalenti per numero e tipo di porte

$$Y = \bar{x}_3\bar{x}_0 + x_3x_0 \qquad SP$$

$$Y = (x_3 + \bar{x}_0)(\bar{x}_3 + x_0) \qquad PS$$



schema a porte della versione SP

d) Per il ritardo $t_{pd} = t_{NOT} + t_{AND2} + t_{OR2} =$

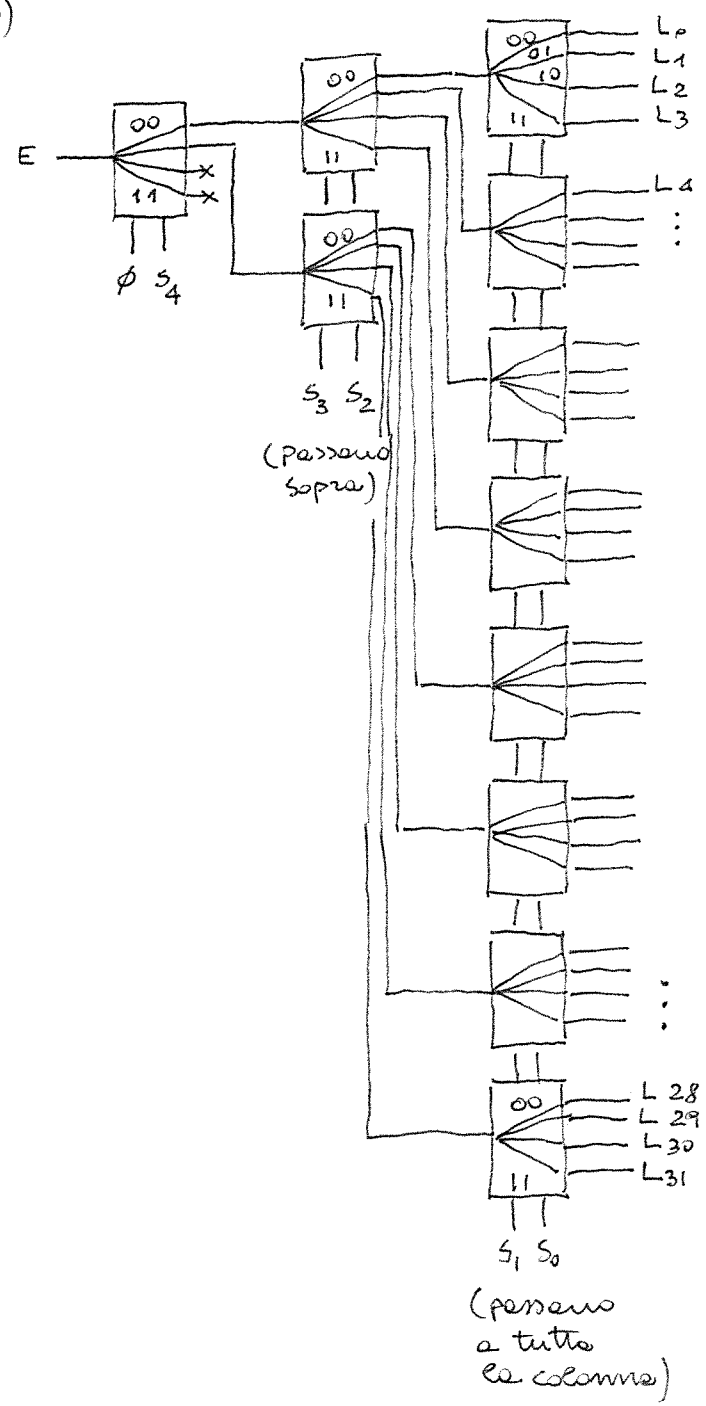
$$= (0{,}1 + 0{,}2) + (0{,}1 + 0{,}4) + (0{,}1 + 0{,}4) = 1{,}3\ ns$$

Se la rete viene usata in una macchina sequenziale

$$f_{MAX} = \frac{1}{t_{pd} + t_{SU} + t_{CO}} = \frac{1}{1{,}3 + 0{,}2 + 0{,}2}\ GHz = 588\ MHz$$

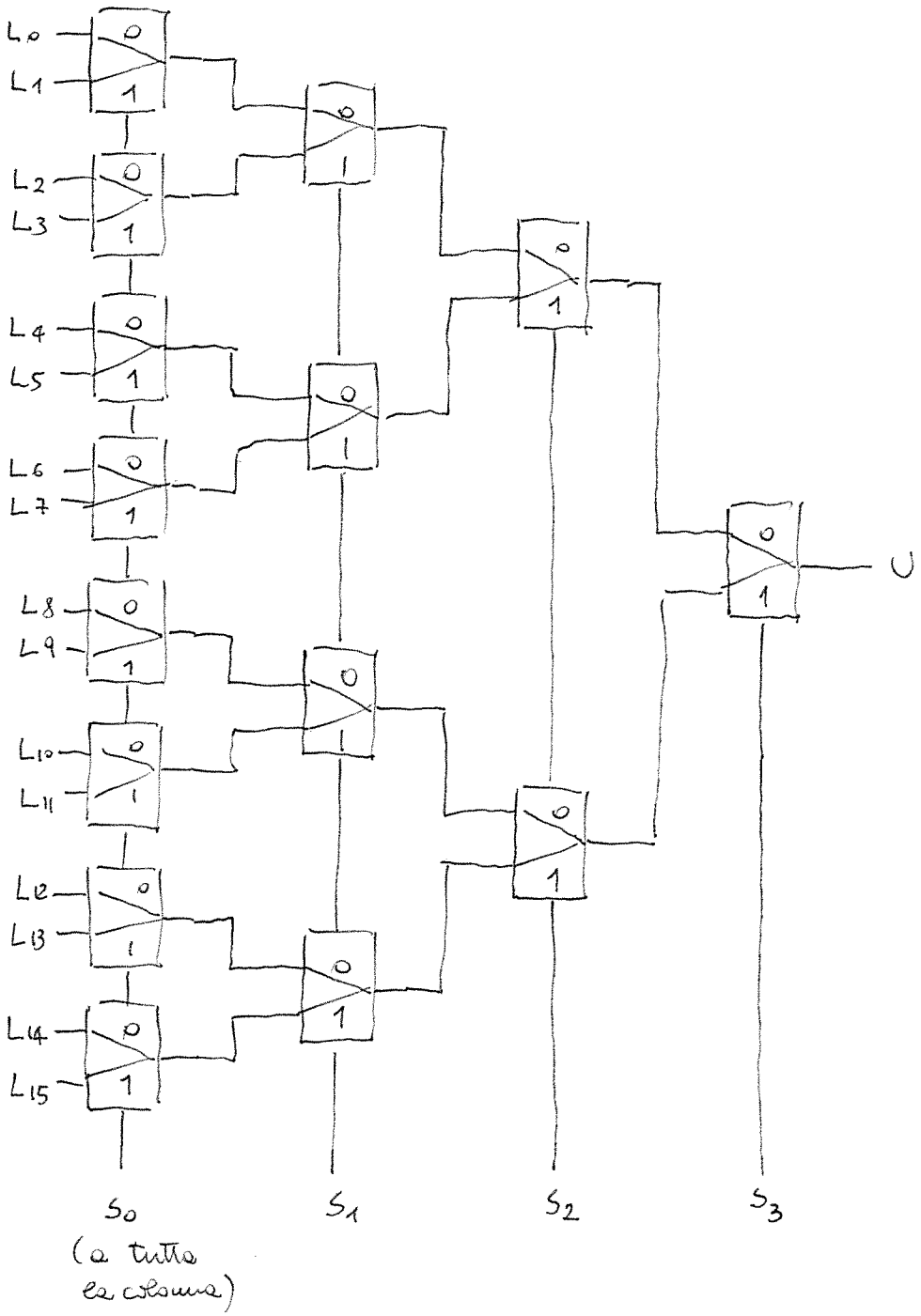
(2)

a)



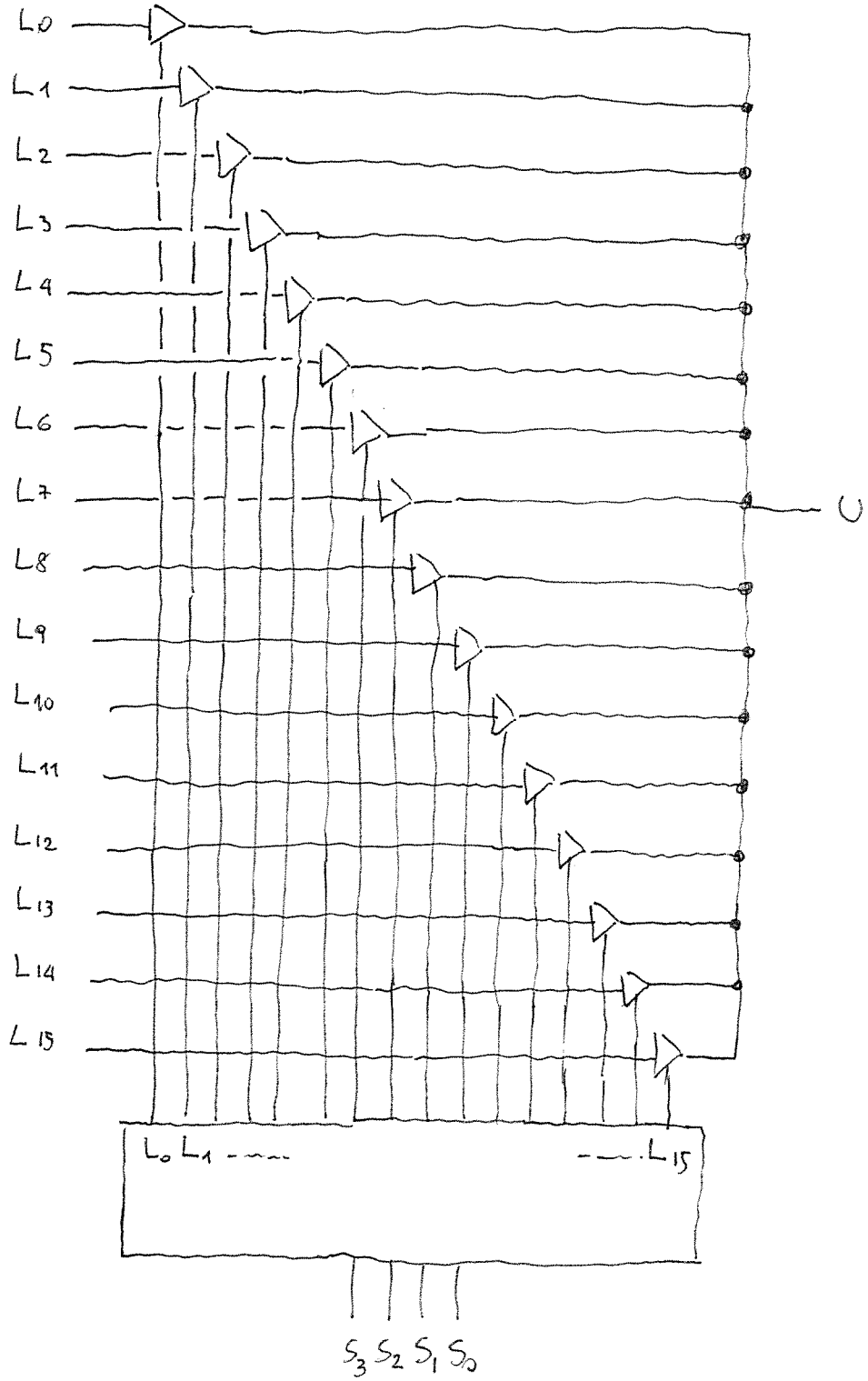
$T_{pd\,tot} = 3\,T_{pd\,decoder} = 9\,ns$

b)



$T_{pd\,tot} = 4\, T_{pd\,mux} = 8\ ns$

c)


L0
L1
L2
L3
L4
L5
L6
L7
L8
L9
L10
L11
L12
L13
L14
L15
U
L0 L1 ---
--- L15
S3 S2 S1 S0

(3)

a) Numeri da rappresentare

$-130,25$; $2,625$; $-55,125$

Le parti frazionarie possono essere rappresentate senza errori con 3 bit (2/8 ; 5/8 ; 1/8)

Le parti intere, col segno, richiedono 9 bit, per tutte le rappresentazioni indicate - In totale occorrono 12 bit

$-130,25$	MS	1	10000010.010
	C1	1	01111101.101
	C2	1	01111101.110
	T	0	01111101.110
$2,625$	MS	0	00000010.101
	C1	0	00000010.101
	C2	0	00000010.101
	T	1	00000010.101
$-55,125$	MS	1	00110111.001
	C1	1	11001000.110
	C2	1	11001000.111
	T	0	11001000.111

b) Usando una ALU a 8 bit C2, per rappresentare i numeri dati occorre rinunciare ai 4 bit MENO significativi -
Esaminiamo l'errore introdotto dal troncamento

$-130,25$	$\mapsto$	10111110	0.	-132	$\varepsilon_A = 1,75$ $\quad \varepsilon_r = 1,34\%$
$2,625$	$\mapsto$	00000010	0.	2	$\varepsilon_A = 0,625$ $\quad \varepsilon_r = 23,8\%$
$-55,125$	$\mapsto$	11100100	0.	-56	$\varepsilon_A = 0,875$ $\quad \varepsilon_r = 1,59\%$

Il troncamento porta sempre un errore assoluto positivo (in C2) tra 0 e il valore dell'LSB.
In questo caso $0 \leq \varepsilon_A < 2$

c) La notazione IEEE 754 in questo caso, disponendo di un numero di cifre significative molto maggiore di 12, rappresenta esattamente i 3 valori assegnati.

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

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

s e=134

[1 | 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]

$$B = 2,625 = (-1)^0 \, 2^1 \left(1 + 2621440 \cdot 2^{-23}\right)$$

s e=128

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

$$C = -55,125 = (-1)^1 \cdot 2^5 \left(1 + 6062080 \cdot 2^{-23}\right)$$

s e=132

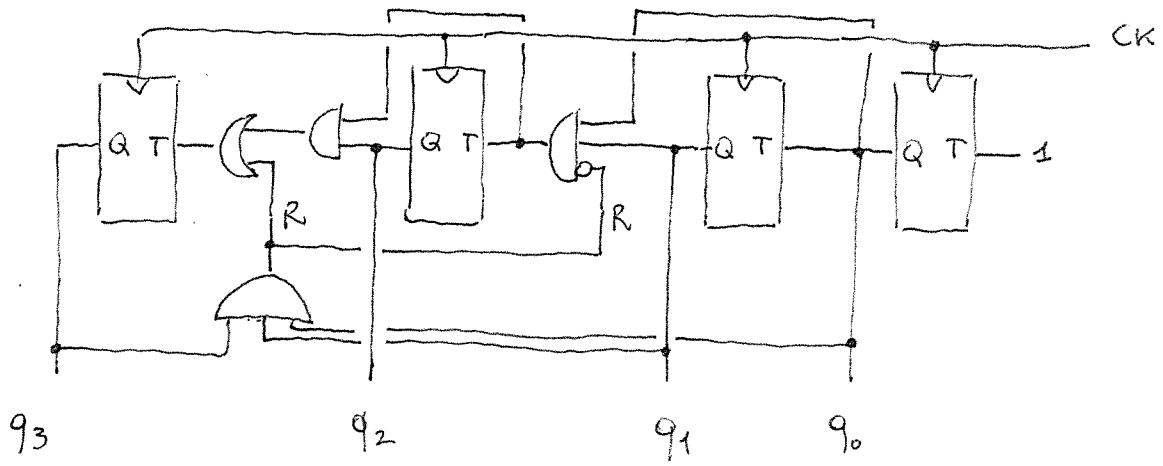
[1 | 1 0 0 0 0 1 0 0 | 1 0 1 . 1 1 0 0 . 1 0 0 0 . 0 0 0 0 . 0 0 0 0 . 0 0 0 0]

(4)

La sequenza di conteggio è

⋮
1 0 0 0
1 0 0 1
1 0 1 0
1 0 1 1
(1 1 0 0) nel contatore modulo 16
0 0 0 0
0 0 0 1
⋮

Quindi occorre Forzare q_3
Bloccare q_2



R: riconosce 1011 (anche 1111, ma questo stato non si presenta mai)

(5) Poiché l'uscita U dipende solo dallo stato $\{q_2\ q_1\ q_0\}$, si tratta di una macchina di MOORE

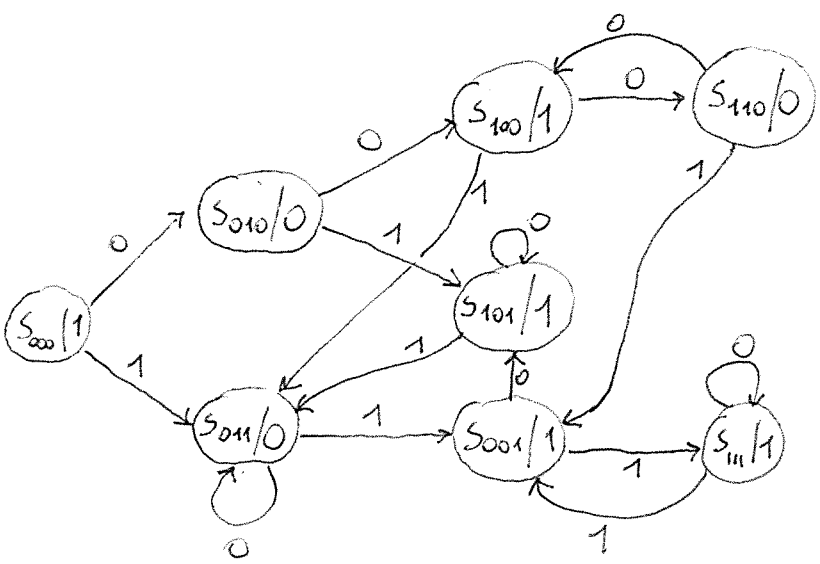
Per costruire il grafo di flusso occorre partire dalla tabella di eccitazione e poi transizione

q_2	q_1	q_0	IN	J	K	q_2^+	T	q_1^+	D	E	q_0^+	S	S^+	U
0	0	0	0	0	0	0	1	1	1	0	0	S_{000}	S_{010}	1
			1	0	1	0	1	1	1	1	1		S_{011}	
0	0	1	0	1	0	1	0	0	1	0	1	S_{001}	S_{101}	1
			1	1	1	1	1	1	1	1	1		S_{111}	
0	1	0	0	1	0	1	1	0	0	0	0	S_{010}	S_{100}	0
			1	1	1	1	1	0	1	1	1		S_{101}	
0	1	1	0	0	0	0	0	1	0	0	1	S_{011}	S_{011}	0
			1	0	1	0	1	0	1	1	1		S_{001}	
1	0	0	0	0	0	1	1	1	1	0	0	S_{100}	S_{110}	1
			1	0	1	0	1	1	1	1	1		S_{011}	
1	0	1	0	1	0	1	0	0	1	0	1	S_{101}	S_{101}	1
			1	1	1	0	1	1	1	1	1		S_{011}	
1	1	0	0	1	0	1	1	0	0	0	0	S_{110}	S_{100}	0
			1	1	1	0	1	0	1	1	1		S_{001}	
1	1	1	0	0	0	1	0	1	0	0	1	S_{111}	S_{111}	1
			1	0	1	0	1	0	1	1	1		S_{001}	

$J = q_1 \oplus q_0$ $\quad K = IN$ $\quad T = IN + \overline{q_0}$ $\quad D = IN + \overline{q_1}$ $\quad E = IN$ $\quad U = q_2 q_1 q_0 + \overline{q_2 q_1}\,(q_2 + \overline{q_1})$

$$U = q_2 q_1 q_0 + \overline{q_2 q_1}\,(q_2 + \overline{q_1}) = q_2 q_1 q_0 + \overline{q_2}\,\overline{q_1} + \overline{q_1} q_2 + \overline{q_1} = q_2 q_1 q_0 + \overline{q_1}$$

Grafo



6

Realizzare una subroutine per il microcontrollore AVR XMEGA256A3BU, che converta in binario il numero di 4 cifre BCD contenuto in X, ponendo il risultato in Y. Nel caso in cui il numero di partenza non sia valido, in Y deve essere messo il valore esadecimale 0xFFFF.

```
/* La soluzione proposta esegue la conversione riconducendola a quella
   della parte bassa e della parte alta di X
*/


Xbcd2Ybin:
 push R0
 push R1
 push R16
 push R17
 ldi YL,0xFF
 ldi YH,0xFF
 ldi R16,0x99
 cp R16,XL //verifica che le cifre [d,u] siano valide
 brhs error
 brcs error
 cp R16,XH //verifica che le cifre [m,c] siano valide
 brhs error
 brcs error
 mov R16,XL
 rcall bcd_2bin //converte R16 da BCD a binario
 mov YL,R16
 clr YH
 mov R16,XH
 rcall bcd_2bin
 ldi R17,100
 mul R16,R17 //moltiplica per 100
 add YL,R0
 adc YH,R1
error:
 pop R17
 pop R16
 pop R1
 pop R0
 ret

bcd_2bin: //converte R16 da BCD a binario
 push R17
 mov R17,R16
 andi R16,0x0F //isola u in R16
 andi R17,0xF0 //in R17 resta 16*d
 lsr R17 //in R17 resta 8*d
 add R16,R17 //in R16 ci va 8d+u
 lsr R17 //in R17 resta 4*d
 lsr R17 //in R17 resta 2*d
 add R16,R17 //in R16 ci va 10*d+u
 pop R17
 ret
```