

Cognome

Nome

ESERCIZIO N°1

8 punti

Realizzare un sottoprogramma per il microcontrollore AVR XMEGA256A3BU, in grado di valutare la somma dei quadrati di 256 valori interi con segno su 8 bit contenuti in memoria a partire dalla locazione puntata da Y. Il risultato (rappresentato come intero senza segno su un numero opportuno di byte, da valutare da parte dello studente) deve essere collocato in memoria a partire dalla locazione puntata da Z (LSByte first).

ESERCIZIO N°2

4 punti

Determinare la rappresentazione secondo lo standard IEEE754 (binary32) dei seguenti valori, usando se necessario l'arrotondamento classico.

2865799,896

-1/197863

$\ln[(155^{155})^{155}]$

$1804!/(1803! + 1805!)$

ESERCIZIO N°3

5 punti

Realizzare in forma NOR-NOR ottima una rete combinatoria a 5 ingressi (le cifre binarie di un numero intero senza segno) e 1 uscita, che vale 0 in corrispondenza dei multipli di 2 o di 3 o di 7, e vale 1 in corrispondenza dei multipli di 5 non compresi nei precedenti insiemi. Per i valori rimanenti, l'uscita può assumere un valore qualsiasi.

ESERCIZIO N°4

5 punti

Disegnare lo schema logico di un contatore sincrono down modulo 12, con reset sincrono, realizzato con T-FF.

ESERCIZIO N°5

5 punti

Disegnare lo schema logico di un comparatore digitale (in grado di fornire l'indicazione $A < B$, $A > B$ oppure $A = B$) tra numeri relativi a 4 bit rappresentati in C1.

ESERCIZIO N°6

6 punti

Progettare una rete sequenziale sincrona secondo il modello di Moore con 2 ingressi e 1 uscita, in grado generare le seguenti forme d'onda:

00: l'uscita è fissa a 0

01: l'uscita ha frequenza pari a $f_{clk}/2$ (il periodo è costituito dai valori 01)

10: l'uscita ha frequenza pari a $f_{clk}/4$ (il periodo è costituito dai valori 0011)

11: l'uscita è fissa a 1

La macchina assume subito la nuova configurazione impostata dopo il fronte di clock.

1

Realizzare un sottoprogramma per il microcontrollore AVR XMEGA256A3BU, in grado di valutare la somma dei quadrati di 256 valori interi con segno su 8 bit contenuti in memoria a partire dalla locazione puntata da Y. Il risultato (rappresentato come intero senza segno su un numero opportuno di byte, da valutare da parte dello studente) deve essere collocato in memoria a partire dalla locazione puntata da Z (LSByte first).

```
/* Sottoprogramma squaresum
   Il risultato, senza segno, sarà su 3 byte.
   Il massimo valore ottenibile infatti è
   R_Max =  $(-2^7)^2 * 2^8 = 2^{22}$  e quindi  $R\_Max < 2^{24} - 1$ 
*/
squaresum:
push R0
push R1
push R2
push R16
push R18
push R20
push R21
push R22
clr R16 //contatore per 256 iterazioni
clr R2 //registro di appoggio nullo
clr R20 //contenitore provvisorio del risultato
clr R21
clr R22
loop:
ld R18,Y+
muls R18,R18 //il risultato sarà sicuramente positivo
add R20,R0
adc R21,R1
adc R22,R2
dec R16
brne loop
st Z,R20
std Z+1,R21
std Z+2,R22
dec YH //ripristina il puntatore Y togliendo 256
pop R22
pop R21
pop R20
pop R18
pop R16
pop R2
pop R1
pop R0
ret
```

② Rappresentazione binary 32 (normalizzati)

$$x = (-1)^S \cdot 2^{(E-127)} \cdot (1 + T \cdot 2^{-23})$$

$$2865799,896 = +2^{21} (1 + 3074592 \cdot 2^{-23}) \quad \text{con error.}$$

quindi, convertendo nel formato

$$\begin{array}{c} 0 | 10010100 | 0101110111010100010000 \\ S \quad E \quad T \end{array}$$

$$-1/197863 = -2^{-18} (1 + 2725260 \cdot 2^{-23}) \quad \text{con error.}$$

da cui

$$\begin{array}{c} 1 | 01101101 | 01010011001010110001100 \\ S \quad E \quad T \end{array}$$

$$\ln [(155^{155})^{155}] = 155^2 \cdot \ln 155 = +2^{16} (1 + 7120933 \cdot 2^{-23}) \quad \text{err.}$$

da cui

$$\begin{array}{c} 0 | 10001111 | 11011001010100000100101 \\ S \quad E \quad T \end{array}$$

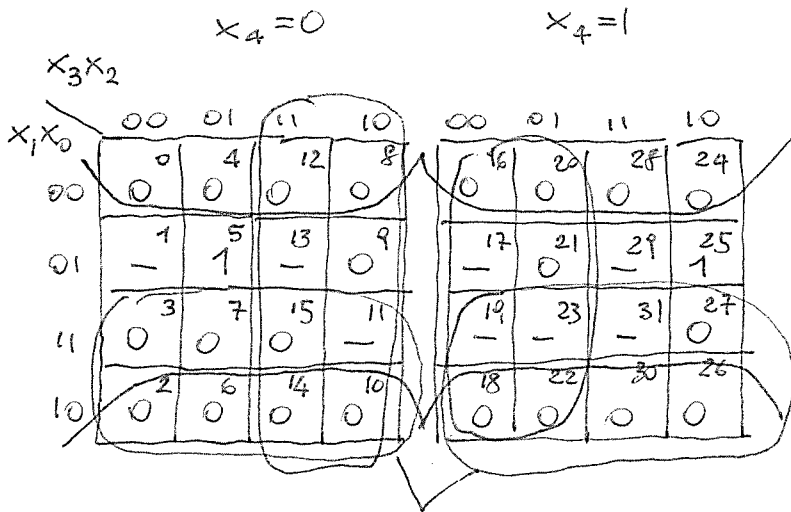
$$\frac{1804!}{1803! + 1805!} = \frac{1804}{1 + 1804 \cdot 1805} = +2^{-11} (1 + 1129322 \cdot 2^{-23})$$

da cui

$$\begin{array}{c} 0 | 01110100 | 00100010011101101101010 \\ S \quad E \quad T \end{array}$$

3

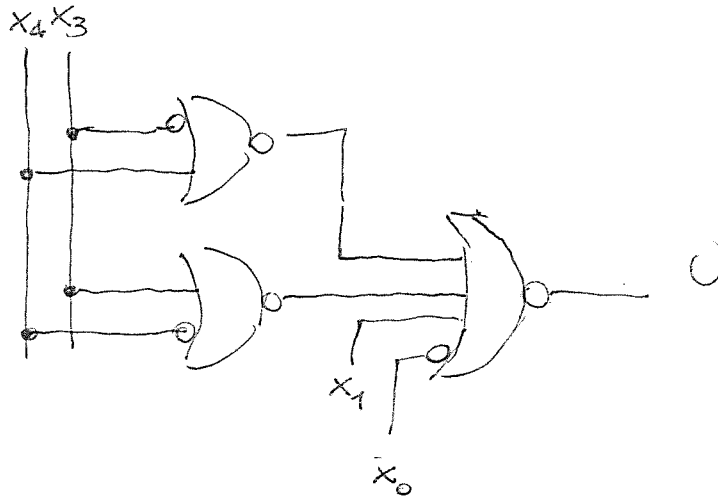
mappe



Porto della PS

$$U = x_0 \cdot \bar{x}_1 (x_4 + \bar{x}_3)(\bar{x}_4 + x_3)$$

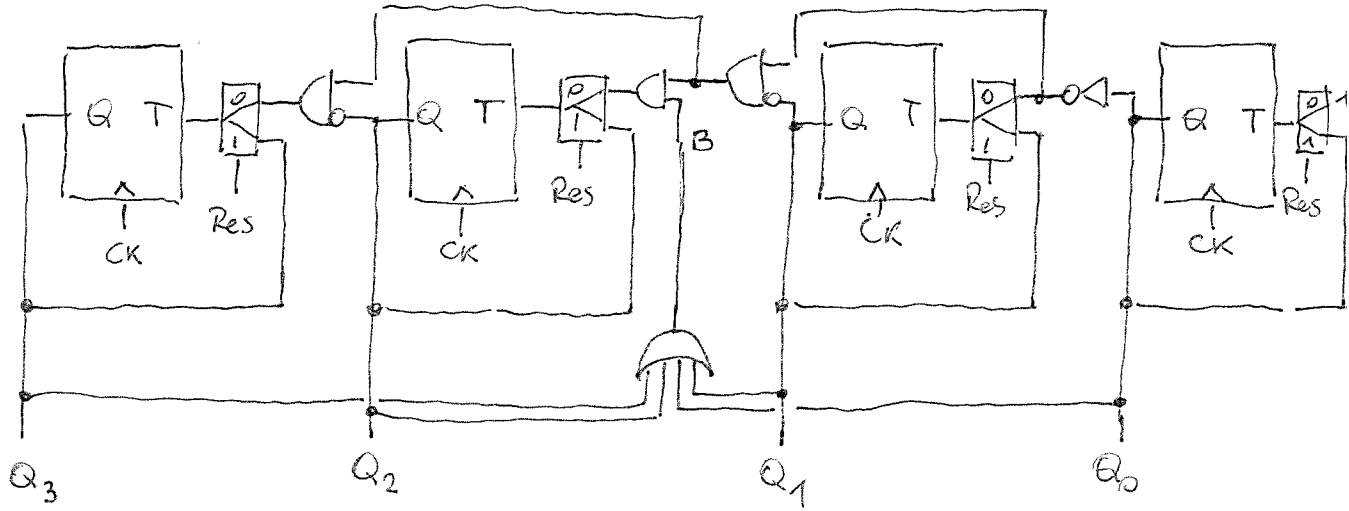
Sintesi NOR-NOR (De Morgan)



④ counter down (mod 12) can reset

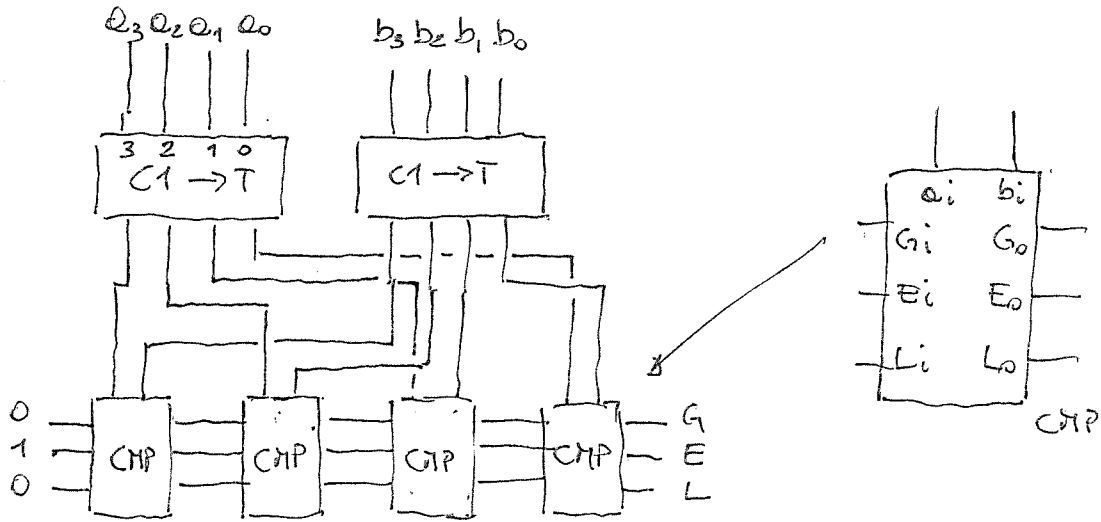
0000
1111

1011
-B--



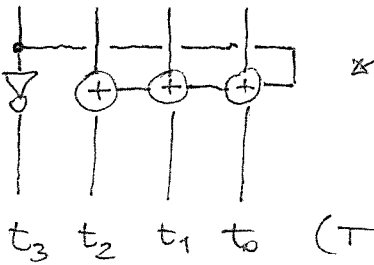
5

Per superare l'ambiguità del ± 0 (0000, 1111) e il fatto che tutti i negativi SEGUONO i positivi (pur se ordinati CORRETTAMENTE) conviene pensare in base 2 e usare un comparatore standard

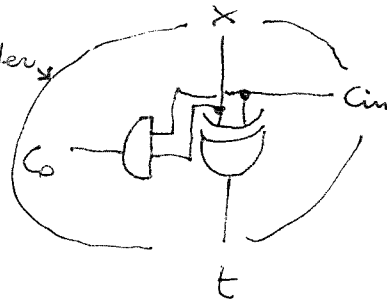


Conversione

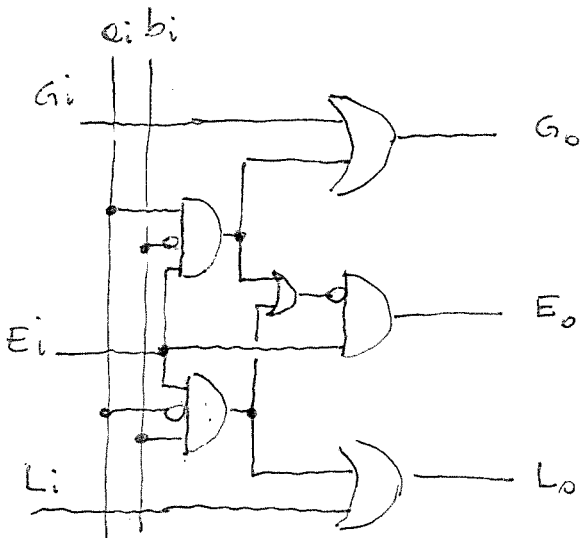
$x_3 x_2 x_1 x_0$ (C1)



Half Adder

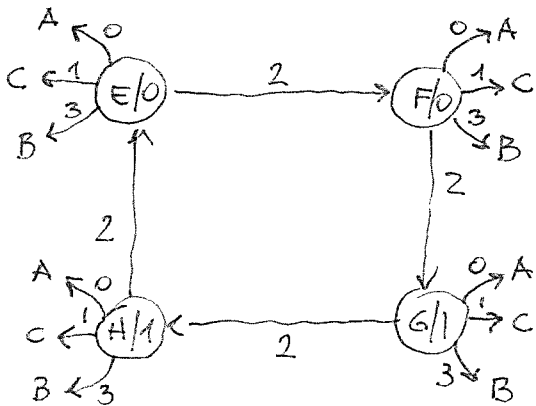
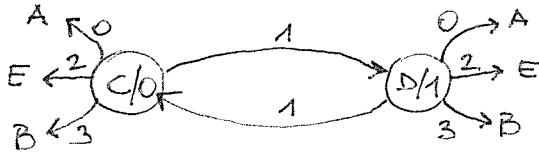
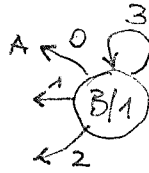
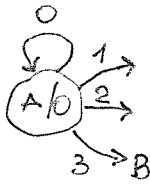


comparatore (MSB fissa)



6 GRAFO DELLE TRANSIZIONI

ingressi x_1, x_0 (0, 1, 2, 3)

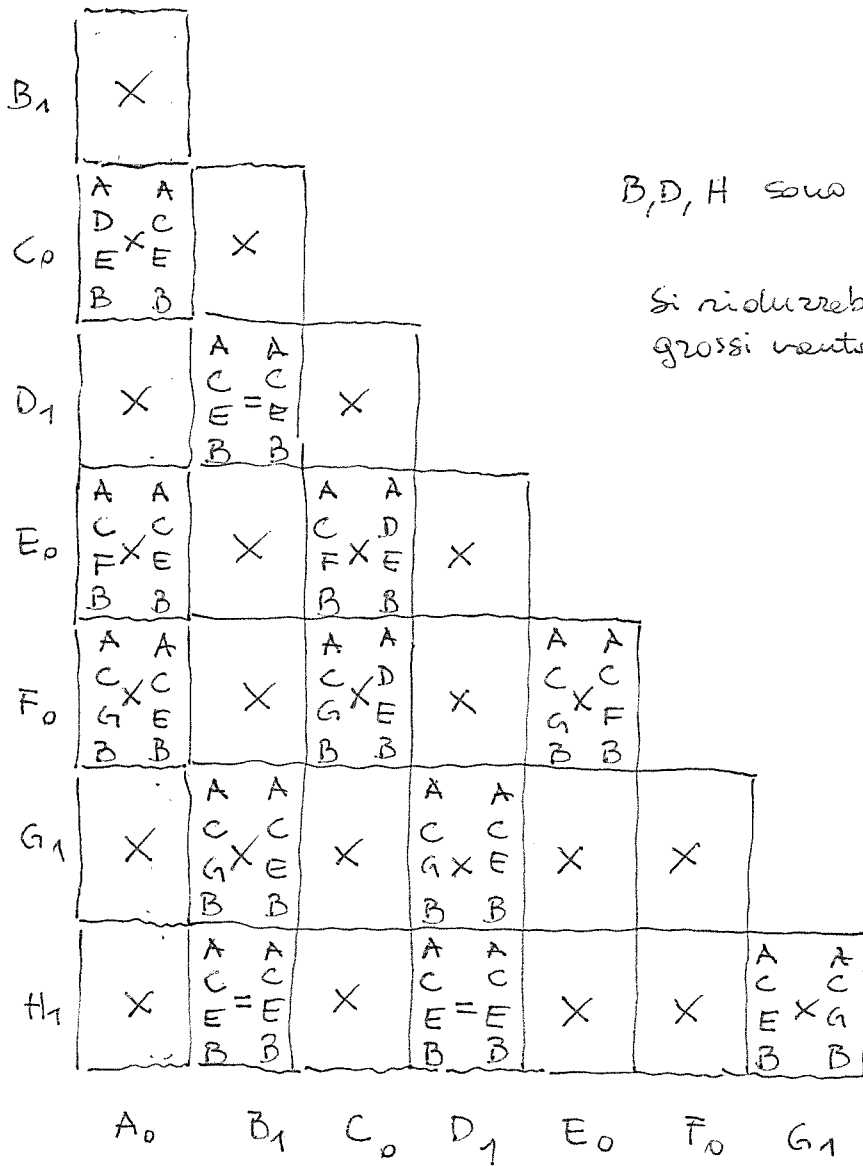


8 STATI

È improbabile che l'analisi delle equivalenze porti all'uso di solo 2 variabili di stato.

(Viene comunque svolta a pag. seguente)

ANALISI DI EQUIVALENZA (opt.)

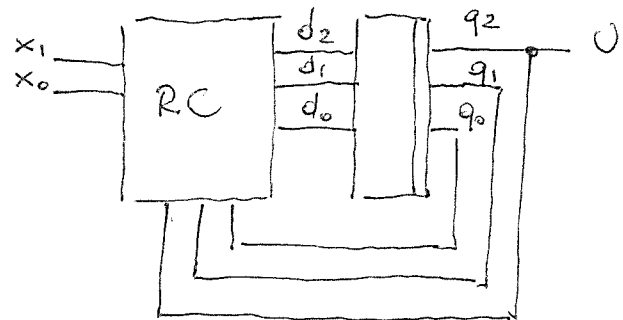


Codifica (per semplificare rete di uscita)

	q ₂	q ₁	q ₀
A	0	0	0
C	0	0	1
E	0	1	0
F	0	1	1
B	1	0	0
D	1	0	1
G	1	1	0
H	1	1	1

$$U \equiv q_2$$

Architettura

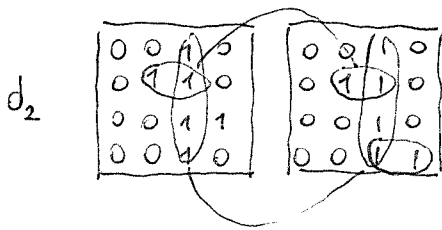


Sintesi RC

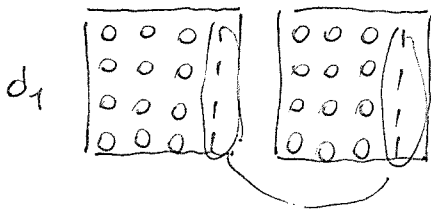
		IN								
		x_1x_0				(1)		(2)		
		(0)	1	3	2	(0)	(1)	(3)	(2)	
		00	01	11	10	00	01	11	10	
(A)	00	000 ^A	001 ^C	100 ^B	010 ^E	000 ^A	001 ^C	100 ^B	010 ^E	(B)
(C)	01	000 ^A	101 ^D	100 ^B	010 ^E	000 ^A	101 ^D	100 ^B	010 ^E	(D)
(F)	11	000 ^A	001 ^C	100 ^B	110 ^G	000 ^A	001 ^C	100 ^B	010 ^E	(H)
(E)	10	000 ^A	001 ^C	100 ^B	011 ^F	000 ^A	001 ^C	100 ^B	110 ^G	(G)

$$q_2 = 0$$

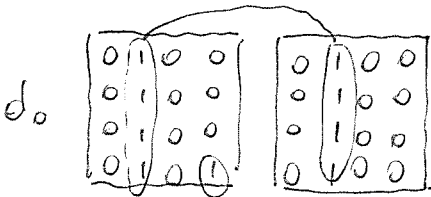
$$q_2 = 1$$



$$d_2 = x_1x_0 + x_0\bar{q}_1q_0 + x_1q_2q_1\bar{q}_0$$



$$d_1 = x_1\bar{x}_0$$



$$d_0 = \bar{x}_1x_0 + x_1\bar{x}_0\bar{q}_2q_1\bar{q}_0$$