

Il testo deve essere riconsegnato nella cartellina. **Non usare il colore rosso nello svolgimento.**
Appello straordinario: l'orale si deve tenere in questa stessa sessione.

ESERCIZIO N°1

8 punti

Realizzare una subroutine per il microcontrollore AVR XMEGA256A3BU che determina il numero di occorrenze complessivo dei codici 0x00, 0x80 e 0xFF contenuti nello spazio di memoria dati compreso tra gli indirizzi 0x2222 e 0x2321. Il risultato deve essere lasciato nel registro R16.

ESERCIZIO N°2

5 punti

Realizzare in forma SP ottima di una rete combinatoria a 5 ingressi, X_4 , X_3 , X_2 , X_1 , e X_0 caratterizzata dalla tabella di verità:

{-, -, -, -, 1, 0, -, 1, -, 1, 0, 1, 0, 1, -, -, 1, 0, 0, 0, -, 1, -, 1, 0, 1, 0, -, 1, -, 1, 1}.

Indicare gli implicanti essenziali, giustificando l'affermazione.

ESERCIZIO N°3

5 punti

Realizzare una macchina sequenziale sincrona secondo il modello di Moore, con 1 ingresso e 1 uscita che riconosce le 3 diverse sequenze 011, 010 e 100, non interallacciate in alcun modo.

ESERCIZIO N°4

5 punti

Determinare la rappresentazione secondo lo standard IEEE-754 (binary 32) dei seguenti due valori e del loro rapporto x/y . Determinare l'errore relativo (in modulo) di rappresentazione che si commette nei 3 casi.

$$x = 48 \cdot 10^{-20}; \quad y = 12 \cdot 10^{20}$$

ESERCIZIO N°5

5 punti

Realizzare un banco di memoria da $1M \times 8$ a costo minimo, avendo a disposizione chip da $256k \times 3$ (costo 0,48€) e $512k \times 5$ (costo 1,20 €).

ESERCIZIO N°6

5 punti

Disegnare lo schema logico di un contatore \overline{U}/D modulo 11 con abilitazione.

/*Realizzare una subroutine per il microcontrollore AVR XMEGA256A3BU che determina il numero di occorrenze complessivo dei codici 0x00, 0x80 e 0xFF contenuti nello spazio di memoria dati compreso tra gli indirizzi 0x2222 e 0x2321. Il risultato deve essere lasciato nel registro R16.*/

```
code_counter:
    push R17
    push XL
    push XH
    clr R16                //per il risultato
    ldi XL,low(0x2222)
    ldi XH,high(0x2222)
loop:
    ld R17,X+             //carica carattere da confrontare
    cpi R17,0x00
    breq ok
    cpi R17,0x80
    breq ok
    cpi R17,0xFF
    breq ok
    dec R16              //neutralizza l'incremento successivo
ok:
    inc R16
    cpi XL,low(0x2321+1)
    brne loop
    cpi XH,high(0x2321+1)
    brne loop
    pop XH
    pop XL
    pop R17
ret
```

2

Realizzare in forma SP ottima di una rete combinatoria a 5 ingressi caratterizzata dalla seguente tabella di verità:

{-, -, -, -, 1, 0, -, 1, -, 1, 0, 1, 0, 1, -, -, 1, 0, 0, 0, -, 1, -, 1, 0, 1, 0, -, 1, -, 1, 1}.

Indicare gli implicanti essenziali, giustificando l'affermazione.

X_3, X_2		$X_4 = 0$				$X_4 = 1$			
		X_1, X_0	00	01	11	10	00	01	11
*	00	-	1	0	-	1*	-	*1	0
	01	-	0	1*	1	0	1	-	1
	11	-	1	-	1	0	1	1	-
	10	-	-	-	0	0	-	1	0

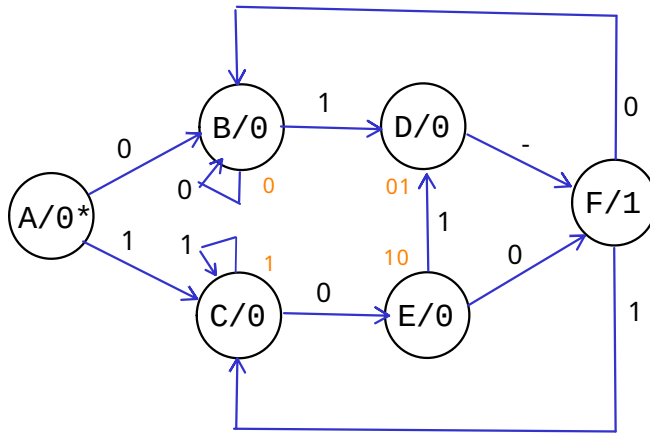
Servono 4 implicanti di cui 3 sono essenziali. La soluzione ottima ha 9 letterali.

Soluzione SP richiesta:

$$Y = \overline{X_3} \overline{X_1} \overline{X_0} + X_3 X_0 + X_4 X_2 + X_2 X_1$$

3

Realizzare una macchina sequenziale sincrona secondo il modello di Moore, con 1 ingresso e 1 uscita che riconosce le 3 diverse sequenze 011, 010 e 100, non interallacciate in alcun modo.

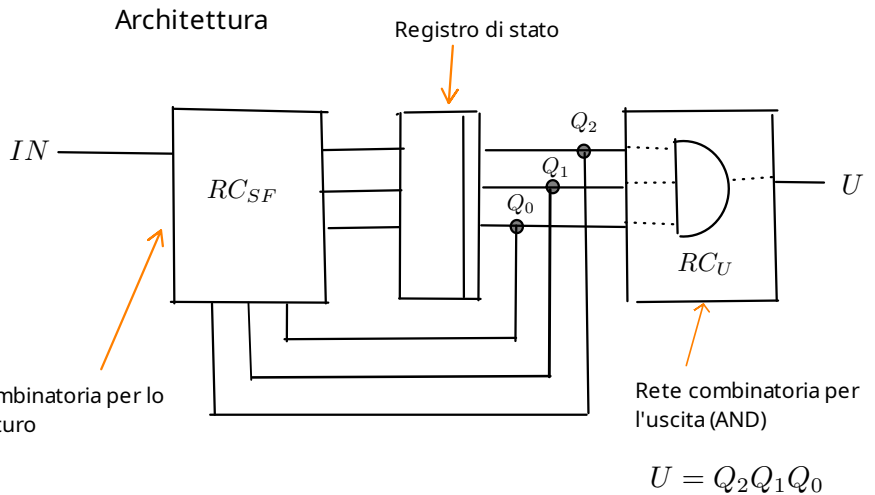


Codifica degli stati

	code	U
A	000	0
B	001	0
C	010	0
D	011	0
E	110	0
F	111	1
X	100 -> 011, 011	
Y	101 -> 001, 010	

Dopo la sintesi, per IN=0,1
Nessuna condizione critica.

Architettura



Sintesi

Q_2, IN	Q_1, Q_0			
	00	01	11	10
00	001 B	001 B	111 F	110 E
01	010 C	011 D	111 F	010 C
11	---	---	010 C	011 D
10	---	---	001 B	111 F

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

$$D_2 = \overline{Q_2}Q_1Q_0 + \overline{IN}Q_1\overline{Q_0}$$

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

$$D_1 = \overline{Q_2}Q_1 + IN + \dot{Q_2}\overline{Q_0}$$

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

$$D_0 = \overline{IN}\overline{Q_1} + \overline{Q_2}Q_0 + Q_2\overline{Q_0} + \overline{IN}Q_2$$

4

Determinare la rappresentazione secondo lo standard IEEE-754 (binary 32) dei seguenti due valori e del loro rapporto x/y.

Determinare l'errore relativo (in modulo) di rappresentazione che si commette nei 3 casi.

$$x = 48 \cdot 10^{-20} \simeq 2^{-61} (1 + 895942 \cdot 2^{-23})$$

$$S=0 \quad E=-61+127=66=0x42 \quad T=895942=0x0DABC6$$

0 01000010 000 1101 1010 1011 1100 0110

$$\epsilon_x = \frac{\hat{x} - x}{x} \simeq -3,17 \cdot 10^{-8}$$

$$y = 12 \cdot 10^{20} \simeq 2^{70} \cdot (1 + 137905 \cdot 2^{-23})$$

$$S=0 \quad E=70+127=197=0xC5 \quad T=137905=0x021AB1$$

0 11000101 000 0010 0001 1010 1011 0001

$$\epsilon_y = \frac{\hat{y} - y}{y} \simeq 2,00 \cdot 10^{-8}$$

Il rapporto x/y esce dal range dei numeri normalizzabili e deve essere rappresentato come non normalizzabile.

$$q = x/y = 4 \cdot 10^{-40} \simeq 2^{-149} \cdot 285450$$

$$S=0 \quad E=0=0x00 \quad T=285450=0x044B0A$$

0 00000000 000 0100 0100 1011 0000 1010

$$\epsilon_q = \frac{\hat{q} - q}{q} \simeq 1,62 \cdot 10^{-6}$$

Come prevedibile, l'errore relativo di una rappresentazione non normalizzabile è maggiore di quello tipico dei valori normalizzabili.

5

Realizzare un banco di memoria da $1M \times 8$ a costo minimo, avendo a disposizione chip da $256k \times 3$ (costo 0,48€) e $512k \times 5$ (costo 1,20 €).

Vediamo il costo a megabit presentato dai 2 diversi chip.

A: $\frac{4}{3} \cdot 0,48 = 0,64$

B: $\frac{2}{5} \cdot 1,20 = 0,48$

Considero comunque assemblaggi con ugual numero di parole (1M)

A' ($1M \times 3$) 1,92 [4 chip]

B' ($1M \times 5$) 2,40 [2 chip]

Possiamo arrivare alla taglia richiesta in diversi modi:

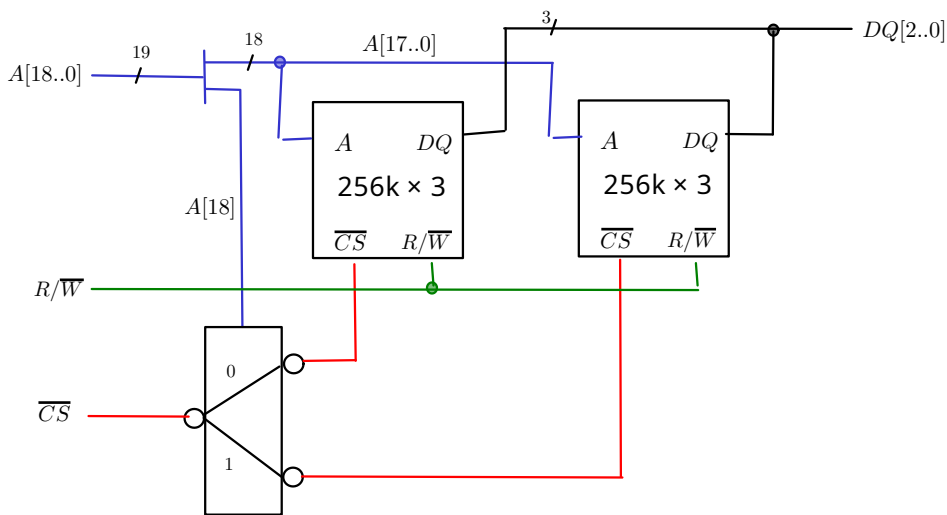
3A' 5,76 con 1 bit inutilizzato

A'+B' 4,32

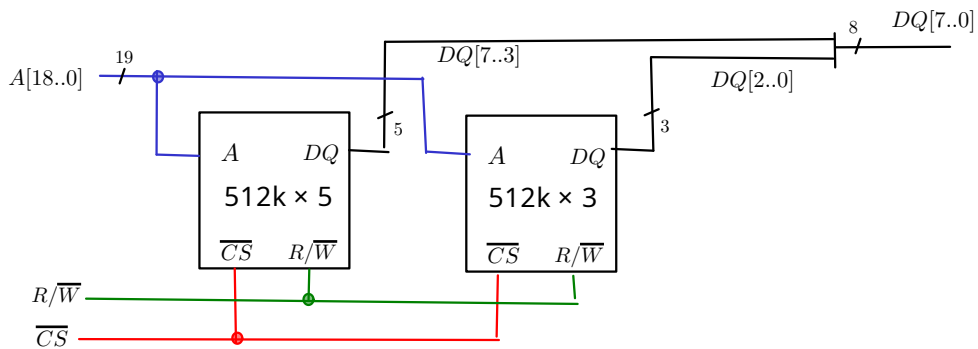
2B' 4,80 con 2 bit inutilizzati

La seconda soluzione è la più conveniente, e usa tutti i bit dei chip disponibili.

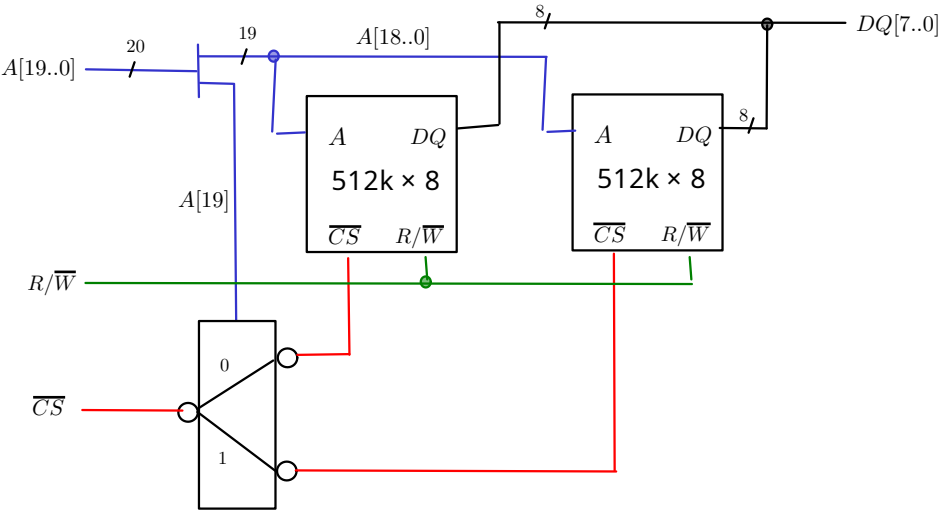
Aumento il numero di parole usando 2 chip A, fino a $512k \times 3$



Combino questo modulo da $512k \times 3$ con un chip da $512k \times 5$ per ottenere $512k \times 8$



Uso infine 2 di questi moduli per ottenere il sistema finale da 1M x 8



Disegnare lo schema logico di un contatore U/D modulo 11 con abilitazione.

Sequenza e comandi

... ...

1001 0001

1010 0000

1011 1111

0000 1010

0001 1001

... ...

UP DOWN

F-FB -B-B

Comando comune, che va bene per entrambe le direzioni:
FBFB

