SCHEDA **ASE2603**	Data: **16 Febbraio 2026**

Il testo deve essere riconsegnato nella cartellina. Non usare il colore rosso nello svolgimento.

ESERCIZIO N°1

8 punti

Realizzare una subroutine per il microcontrollore AVR XMEGA256A3BU che determina il valore assoluto di un numero intero rappresentato in C2 su 8 byte e contenuto in memoria nelle locazioni consecutive (LSB first) a partire dall'indirizzo contenuto nel puntatore Y. Il risultato, rappresentato sempre su 8 byte come intero senza segno (LSB first), deve essere posto in memoria a partire dall'indirizzo contenuto nel puntatore X. I blocchi del dato e del risultato non hanno intersezione.

ESERCIZIO N°2

5 punti

Disegnare lo schema logico in forma NOR-NOR ottima di una rete combinatoria a 5 ingressi, X_4, X_3, X_2, X_1, e X_0 caratterizzata dalla tabella di verità:
{0, 1, 0, -, 1, -, 0, 1, -, 0, 1, 1, 0 -, 1, 0, 0, 1, -, 0, 1, 1, 0, -, 1, 0, -, 1, -, 0, 1, 0}.
Indicare gli implicati 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 2 diverse sequenze 0010 e 0100, non interallacciate in alcun modo.

ESERCIZIO N°4

5 punti

In un collegamento digitale le informazioni inviate sono cifre esadecimali, su 4 bit. Per rendere più robusta la connessione, ai 4 bit b_3, b_2, b_1, b_0 sono aggiunti altri 3 bit, secondo la seguente legge:
$r_2 = b_1 \oplus b_2 \oplus b_3$
$r_1 = b_0 \oplus b_2 \oplus b_3$
$r_0 = b_0 \oplus b_1 \oplus b_3$
Dimostrare che il codice così ottenuto permette di rivelare e correggere un singolo errore.

ESERCIZIO N°5

5 punti

Realizzare un banco di memoria da 1M × 8 a costo minimo, avendo a disposizione chip da 256k × 3 (costo 0,45 €) e 512k × 5 (costo 1,90 €).

ESERCIZIO N°6

5 punti

Disegnare lo schema logico di un contatore Johnson modulo 5. Si hanno a disposizione D-FF e porte logiche elementari (AND, OR, NOT). L'uscita *one-hot*, su 5 bit, deve presentare la sequenza {10000; 01000; 00100; 00010; 00001}. Discutere le conseguenze sul circuito del valore dello stato iniziale, in particolare individuando per ogni caso non *one-hot* dopo quanti cicli di clock la macchina inizia a comportarsi correttamente.

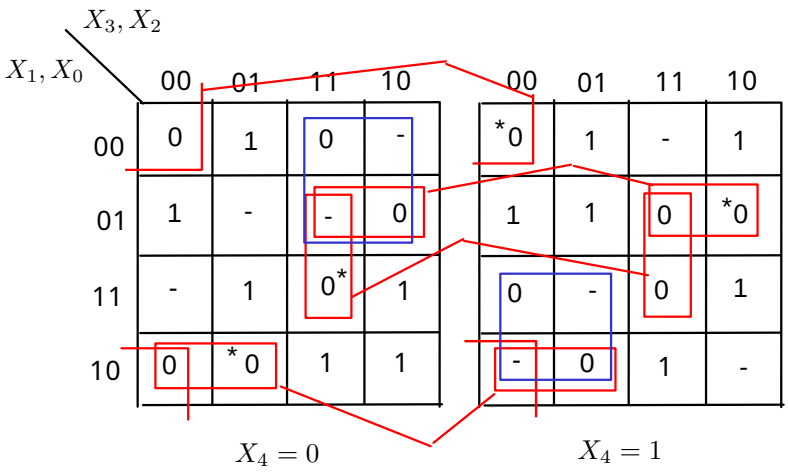
1

```
/*Realizzare una subroutine per il microcontrollore AVR XMEGA256A3BU che determina
  il valore assoluto di un numero intero rappresentato in C2 su 8 byte e
  contenuto in memoria nelle locazioni consecutive (LSB first)
  a partire dall'indirizzo contenuto nel puntatore Y. Il risultato, rappresentato
  sempre su 8 byte come intero senza segno (LSB first), deve essere posto
  in memoria a partire dall'indirizzo contenuto nel puntatore X.
  I blocchi del dato e del risultato non hanno intersezione.*/

absolute:
  push R16
  push R17
  push R18
  clc                   //parte con C=0
  ldi R17,8             //contatore
  ldd R16,Y+7           //MSB per vedere il segno
  sbrs R16,7
  rjmp copy             //dato positivo, basta copiare
  chs:                  //dato negativo, cambia segno
    ld R16,Y+
    clr R18             //non tocca C
    sbc R18,R16         //esegue 0-(byte_dato) con prestito
    st X+,R18
    dec R17             //non tocca C
    brne chs
  rjmp end
  copy:
    ld R16,Y+
    st X+,R16
    dec R17
    brne copy
  end:
  sbiw YH:YL,8          //ripristina i puntatori
  sbiw XH:XL,8
  pop R18
  pop R17
  pop R16
  ret
```

2

Disegnare lo schema logico in forma NOR-NOR ottima di una rete combinatoria a 5 ingressi caratterizzata dalla tabella di verità:
{0, 1, 0, -, 1, -, 0, 1, -, 0, 1, 1, 0 -, 1, 0, 0, 1, -, 0, 1, 1, 0, -, 1, 0, -, 1, -, 0, 1, 0}.
Indicare gli implicati essenziali, giustificando l'affermazione.

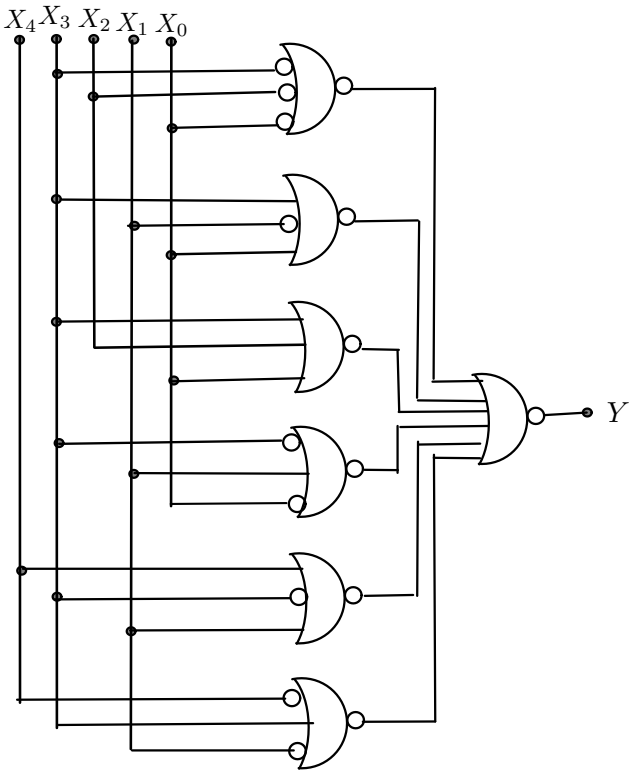


Servono 6 implicati di cui 4 sono essenziali. La soluzione ottima ha 18 letterali.

Soluzione PS. Per avere NOR-NOR applico il teorema di De Morgan alla AND finale.

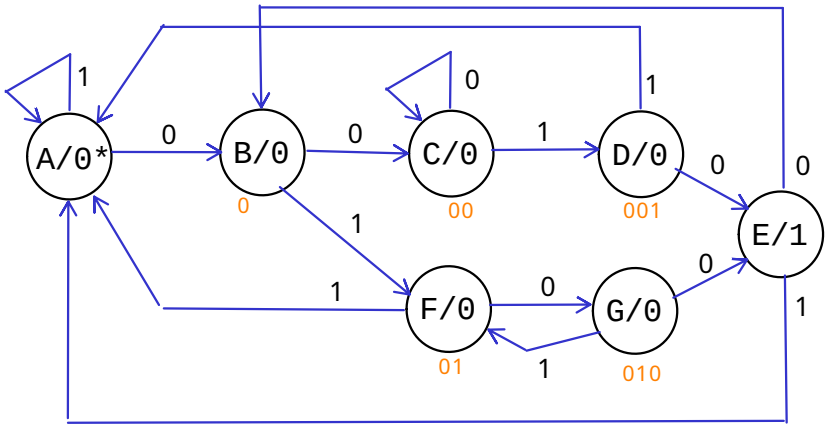
$$Y = (\overline{X_3} + \overline{X_2} + \overline{X_0})(X_3 + \overline{X_1} + X_0)(X_3 + X_2 + X_0)(\overline{X_3} + X_1 + \overline{X_0}) \cdot$$
$$\cdot (X_4 + \overline{X_3} + X_1)(\overline{X_4} + X_3 + \overline{X_1})$$

Schema NOR-NOR:



3

Realizzare una macchina sequenziale sincrona secondo il modello di Moore, con 1 ingresso e 1 uscita che riconosce le 2 diverse sequenze 0010 e 0100, non interallacciate in alcun modo.

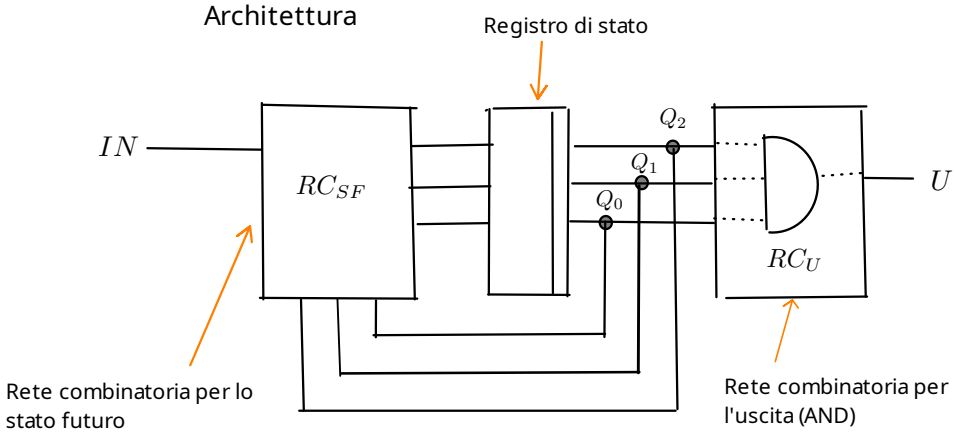


Codifica degli stati

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

Dopo la sintesi, per IN=0,1

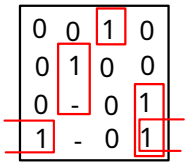
Architettura



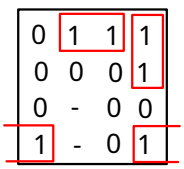
$$U = Q_2 Q_1 Q_0$$

Sintesi

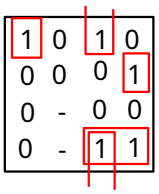
Q_2, IN \ Q_1, Q_0	00	01	11	10
00	001 B	010 C	111 E	010 C
01	000 A	100 F	000 A	011 D
11	000 A	---	000 A	100 F
10	110 G	---	001 B	111 E



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



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



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

4

In un collegamento digitale le informazioni inviate sono cifre esadecimali, su 4 bit.
Per rendere più robusta la connessione, ai 4 bit sono aggiunti altri 3 bit,
secondo la legge indicata nel seguito.
Dimostrare che il codice così ottenuto permette di rivelare e correggere un singolo errore.

$$r_2 = b_1 \oplus b_2 \oplus b_3$$
$$r_1 = b_0 \oplus b_2 \oplus b_3$$
$$r_0 = b_0 \oplus b_1 \oplus b_3$$

La parola che viene inviata è quindi: $b_3, b_2, b_1, b_0, r_2, r_1, r_0$

La parola ricevuta sia: $b_3^r, b_2^r, b_1^r, b_0^r, r_2^r, r_1^r, r_0^r$

Con i bit della parola ricevuta possono essere ricalcolati i bit di ridondanza.
Se non ci sono errori, i bit di ridondanza calcolati coincideranno con quelli ricevuti.

$$r_2^c = r_2^r; \quad r_1^c = r_1^r; \quad r_0^c = r_0^r$$

Se è presente un singolo errore, può appartenere all'insieme dei dati
o a quello dei bit di ridondanza. In quest'ultimo caso i dati sono corretti e si avrà

$$r_i^c \neq r_i^r \quad \text{per uno e un solo } i; \quad r_i^c \text{ è il valore corretto}$$

Se c'è un singolo errore nell'insieme dei dati, si avrà più di una differenza nei bit di ridondanza.
Si avrà:

$$r_2^c \neq r_2^r; \quad r_1^c \neq r_1^r; \quad r_0^c \neq r_0^r \quad \text{se e solo se } b_3^r \neq b_3$$

$$r_2^c \neq r_2^r; \quad r_1^c \neq r_1^r; \quad r_0^c = r_0^r \quad \text{se e solo se } b_2^r \neq b_2$$

$$r_2^c \neq r_2^r; \quad r_1^c = r_1^r; \quad r_0^c \neq r_0^r \quad \text{se e solo se } b_1^r \neq b_1$$

$$r_2^c = r_2^r; \quad r_1^c \neq r_1^r; \quad r_0^c \neq r_0^r \quad \text{se e solo se } b_0^r \neq b_0$$

Quindi tutti i singoli errori sono rivelati e possono essere corretti.

5

Realizzare un banco di memoria da 1M × 8 a costo minimo,
avendo a disposizione chip da 256k × 3 (A: costo 0,45 €) e 512k × 5 (B: costo 1,90 €).

Vediamo il costo a megabit presentato dai 2 diversi chip.

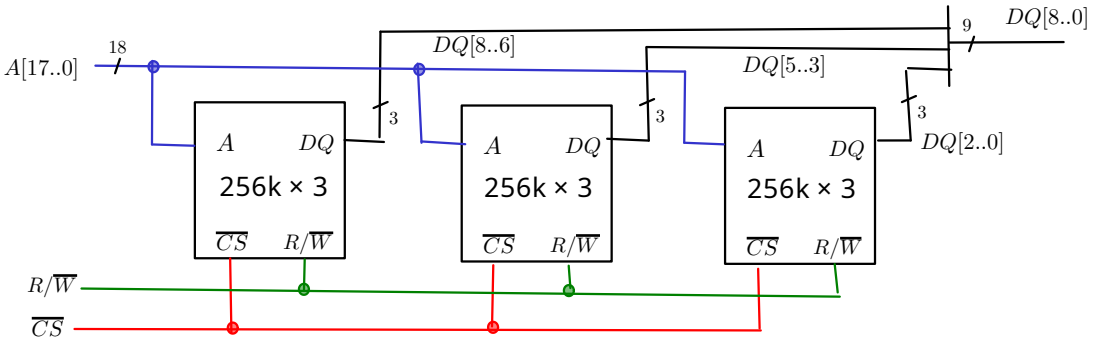
A: $\frac{4}{3} \cdot 0,45 = 0,60$ B: $\frac{2}{5} \cdot 1,90 = 0,76$

Considero comunque assemblaggi con ugual numero di parole (1M)
A' (1M × 3) 1,80 [4 chip]
B' (1M × 5) 3,80 [2 chip]

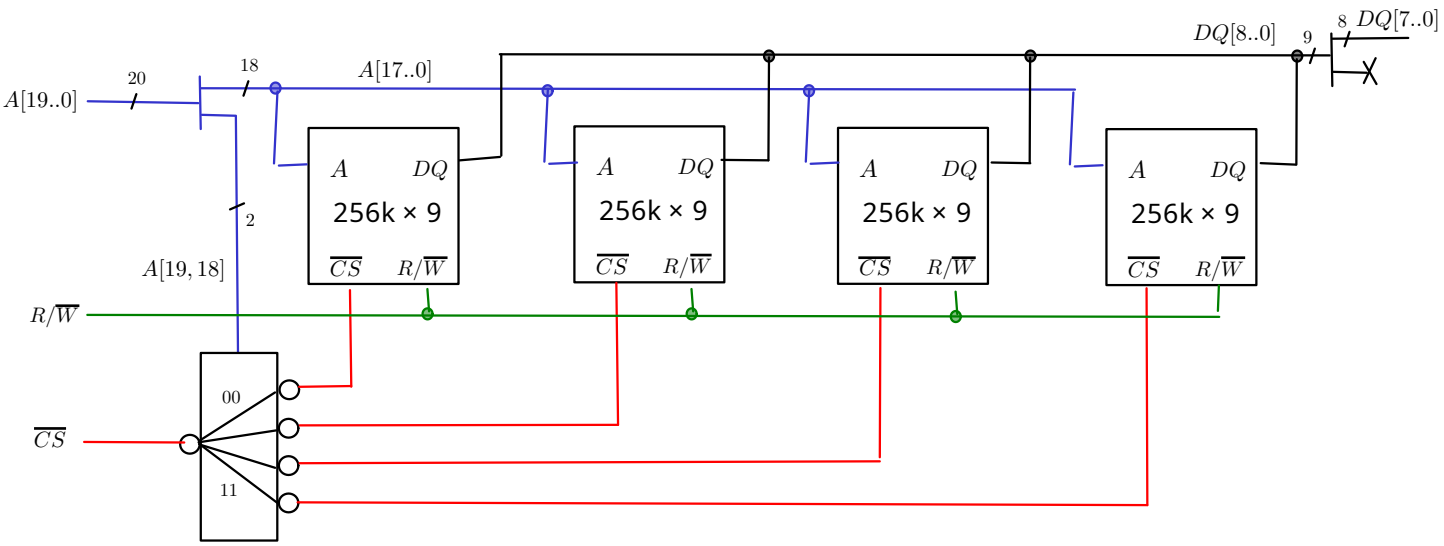
Possiamo arrivare alla taglia richiesta in diversi modi:
3A' 5,40 con 1 bit inutilizzato
A'+B' 5,60
2B' 7,60 con 2 bit inutilizzati

La prima soluzione è la più conveniente, pur avendo un bit non utilizzato,
a causa del minor costo unitario di A.

Inizio col realizzare un modulo 256k × 9

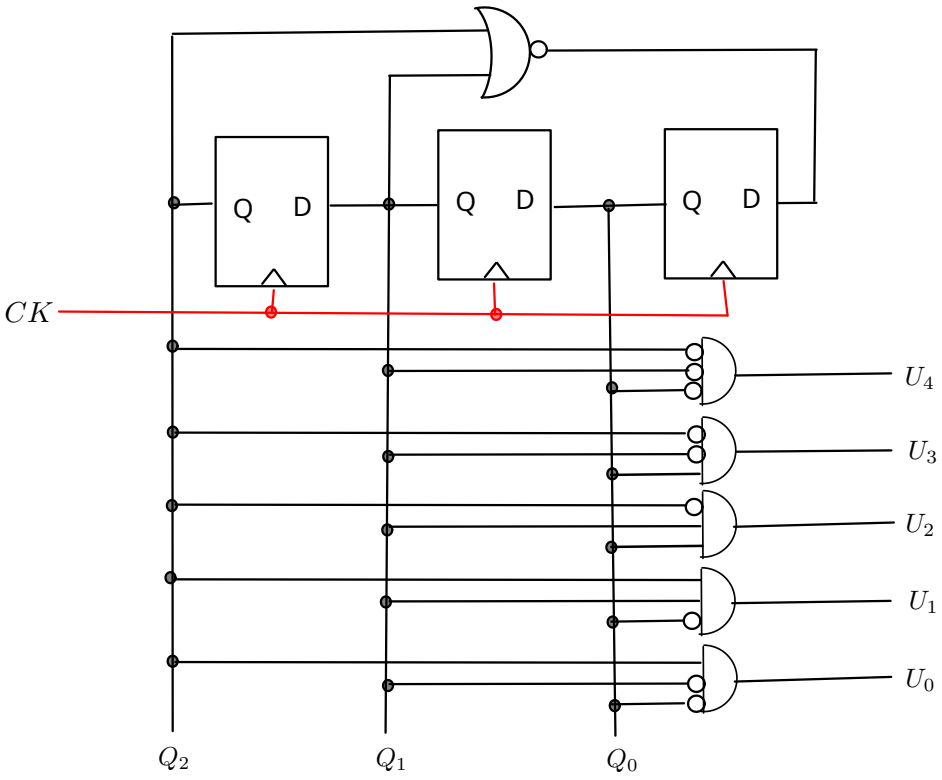


Aumento il numero di parole usando 4 di questi moduli, fino a 1M × 8 (lasciando scollegato un bit)



6

Disegnare lo schema logico di un contatore Johnson modulo 5.
Si hanno a disposizione D-FF e porte logiche elementari (AND, OR, NOT).
L'uscita one-hot, su 5 bit, deve presentare la sequenza {10000; 01000; 00100; 00010; 00001}.
Discutere le conseguenze sul circuito del valore dello stato iniziale,
in particolare individuando per ogni caso non one-hot
dopo quanti cicli di clock la macchina inizia a comportarsi correttamente.



Sequenza

Q2	Q1	Q0	
0	0	0	0
0	0	1	1
0	1	1	3
1	1	0	6
1	0	0	4
(0	0	0)	

Le AND avrebbero potuto essere ottimizzate (2 ingressi), ma così ci sarebbero potute essere uscite one-hot anche con stati non previsti. Così gli stati non previsti danno in uscita 00000

Sequenze con ingresso diverso da 0, 1, 3, 6, 4

Q2	Q1	Q0	
0	1	0	2
1	0	0	4*
1	0	1	5
0	1	0	2
1	0	0	4*
1	1	1	7
1	1	0	6*

Al più dopo 2 cicli di clock si ritorna a una successione di valori one-hot