

SCHEDA ASE2406		Data: 24 Giugno 2024
Cognome	Nome	

Il testo deve essere riconsegnato

ESERCIZIO N°1

8 punti

Realizzare un sottoprogramma per il microcontrollore XMEGA256A3BU che trova il codice del carattere più ripetuto nel segmento di memoria compreso tra gli indirizzi 0x2100 e 0x217F (compresi gli estremi). Il valore del codice deve essere lasciato in R16 (in caso di parità, privilegiare il codice di valore minore).

ESERCIZIO N°2

5 punti

Individuare la notazione frazionale che permette di rappresentare la radice quadrata (positiva) dei numeri naturali da 1 a 10 con un errore assoluto in modulo inferiore a 0,001. Valutare quindi l'errore assoluto in modulo delle rappresentazioni di tali radici.

ESERCIZIO N°3

5 punti

Assemblare chip di memoria in modo da ottenere un modulo da 1M×11 a costo minimo. Sono disponibili chip da 128k×3 (0,83 €), 256k×5 (2,71 €) e 512k×7 (6,80 €)

ESERCIZIO N°4

5 punti

Realizzare in forma NOR-NOR ottima una rete combinatoria non completamente specificata a 5 ingressi, X_4 , X_3 , X_2 , X_1 e X_0 , e 1 uscita Y , i cui mintermini siano $\{0, 2, 3, 5, 8, 11, 15, 17, 19, 22, 27, 31\}$ e l'insieme dei don't care sia $\{1, 4, 6, 20, 23, 29\}$. Indicare gli implicati essenziali.

ESERCIZIO N°5

5 punti

Progettare una macchina sequenziale sincrona secondo il modello di Moore con un ingresso e una uscita in grado di generare una forma d'onda di periodo $6T_{clk}$, il cui ciclo di lavoro valga $1/3$ se l'ingresso letto in corrispondenza della fine di ciascun ciclo vale 1 e $2/3$ altrimenti; la forma d'onda prodotta deve presentare l'impulso alto al centro del periodo (le sequenze sono rispettivamente 001100 con ingresso alto e 011110 con ingresso basso).

ESERCIZIO N°6

5 punti

Realizzare la funzione dell'esercizio 4, se possibile, usando al massimo un multiplexer 8:1, 3 multiplexer 4:1 e 5 multiplexer 2:1.

```
/*Realizzare un sottoprogramma per il microcontrollore XMEGA256A3BU
che trova il codice del carattere più ripetuto nel segmento di memoria
compreso tra gli indirizzi 0x2100 e 0x217F (compresi gli estremi).
Il valore del codice deve essere lasciato in R16
(in caso di parità, privilegiare il codice di valore minore).*/
```

```
max_instances:
  push R17      //numero di occorrenze massime
  push R18      //carattere da provare di cui contare le occorrenze
  push R19      //occorrenze del carattere in prova
  push R20      //dimensione del segmento di memoria (128 celle)
  push R21      //variabile di appoggio per il carattere
  push R22      //conteggio dei codici da provare (0 per 256)
  push XL       //puntatore
  push XH
  ldi XH,high(0x2100) //la parte bassa va reinizializzata sempre
  clr R22       //0 per 256 codici da provare
  clr R16       //soluzione corrente con il massimo di occorrenze
  clr R17       //numero di occorrenze della soluzione corrente
  clr R18       //nuovo carattere da provare
loop_ext:
  ldi R20,128   //reinizializza loop interno
  ldi XL,low(0x2100) //riparte dall'inizio del segmento
  clr R19       //occorrenze del nuovo carattere
  loop_int:
    inc R19     //se poi sono diversi, ridecrementa
    ld R21,X+
    cpse R21,R18
    dec R19     //toglie l'incremento se diversi
    dec R20
    brne loop_int //conta le occorrenze di R18 in R19
  cp R19,R17
  brcc mantieni
  movw R17:R16,R19:R18 //nuova coppia vincente
  mantieni:
    inc R18     //codice successivo
    dec R22
  brne loop_ext
  pop XH
  pop XL
  pop R22
  pop R21
  pop R20
  pop R19
  pop R18
  pop R17
ret
```

2

Individuare la notazione frazionale che permette di rappresentare la radice quadrata (positiva) dei numeri naturali da 1 a 10 con un errore assoluto in modulo inferiore a 0,001. Valutare quindi l'errore assoluto e relativo in modulo delle rappresentazioni di tali radici.

In generale, per rappresentare numeri minori di 4 con errore dopo arrotondamento inferiore a 1/1000 occorre una rappresentazione [2.9]. Infatti l'errore in questo caso sarà sempre minore di LSB/2, cioè di 1/1024.

Se tutte le rappresentazioni dei valori richiesti dovessero avere n cifre meno significative tutte nulle, si potrebbe passare a una rappresentazione [2.(9-n)] a parità di errore.

n	x=sqrt(n)	round(512x)	hex	bin	err in modulo
1	1,000	512	200	01.000000000	0,00E+00
2	1,414	724	2D4	01.011010100	151,06E-06
3	1,732	887	377	01.101110111	371,07E-06
4	2,000	1024	400	10.000000000	0,00E+00
5	2,236	1145	479	10.001111001	260,15E-06
6	2,449	1254	4E6	10.011100110	270,99E-06
7	2,646	1355	54B	10.101001011	733,06E-06
8	2,828	1448	5A8	10.110101000	302,12E-06
9	3,000	1536	600	11.000000000	0,00E+00
10	3,162	1619	653	11.001010011	168,29E-06

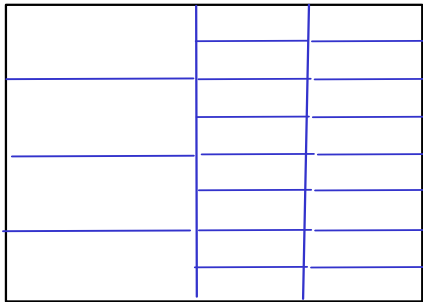
Assemblare chip di memoria in modo da ottenere un modulo da 1M x 11 a costo minimo.

Sono disponibili chip da

a) 128k x 3 (0,83 €),

b) 256k x 5 (2,71 €) e

c) 512k x 7 (6,80 €)



A: 1M x 3 6,64
 B: 1M x 5 10,84
 C: 1M x 7 13,60

Soluzioni possibili

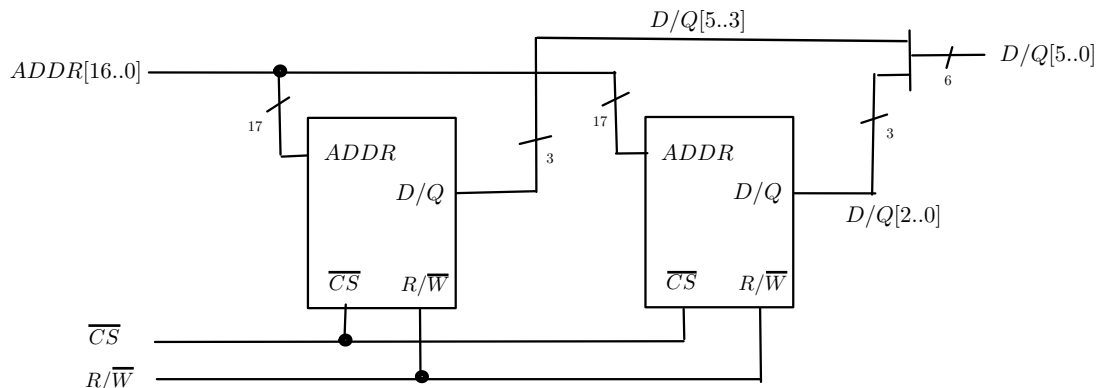
2C	27,20	S:3b
C+B	24,44	S:1b
C+2A	26,88	S:2b
3B	32,52	S:4b
2B+A	28,32	S:2b
B+2A	24,12	S:0
4A	26,56	S:1b

*

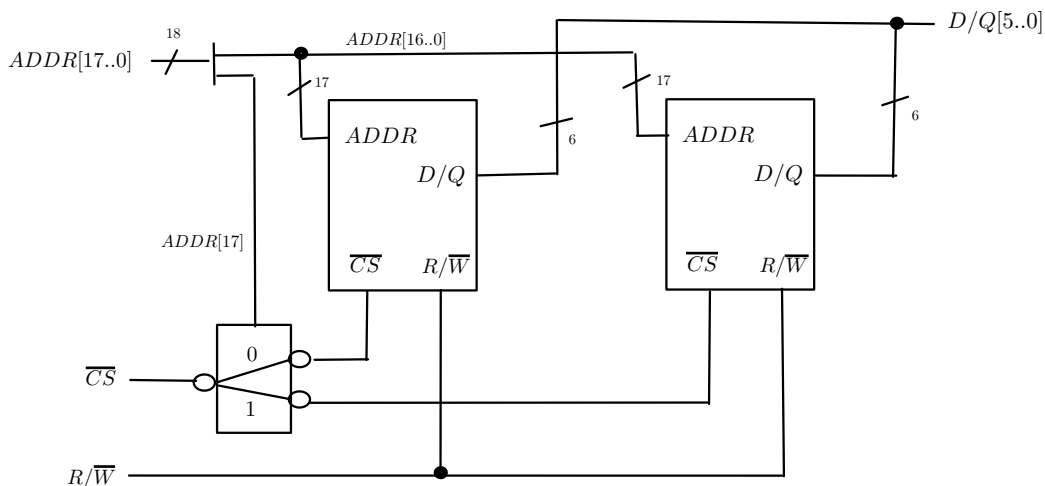
L'analisi delle soluzioni mostra che quella a costo minimo è la sesta, senza scarto.

Realizzo inizialmente 128k x 6, poi raddoppio le parole a 256k x 6, poi aumento la dimensione di parola a 256k x 11, infine quadruplico le parole, arrivando a 1M x 11, che soddisfa la richiesta.

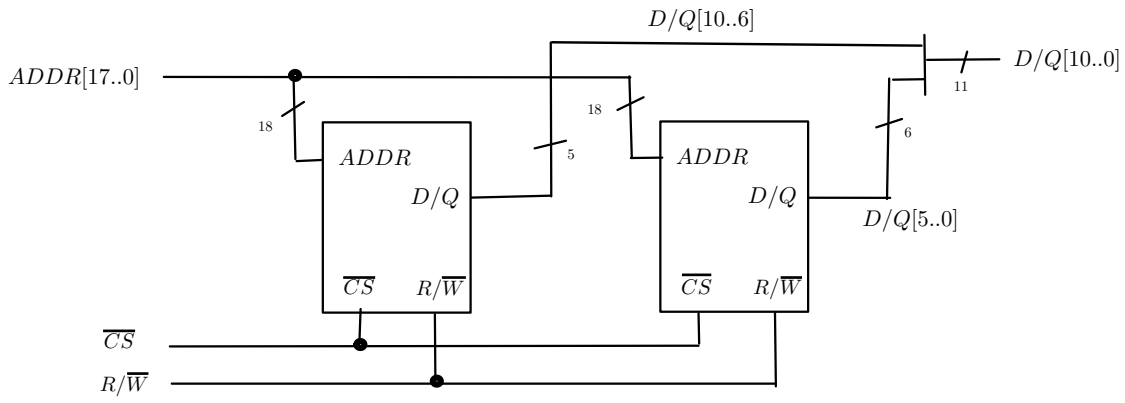
Raddoppio dimensione di parola



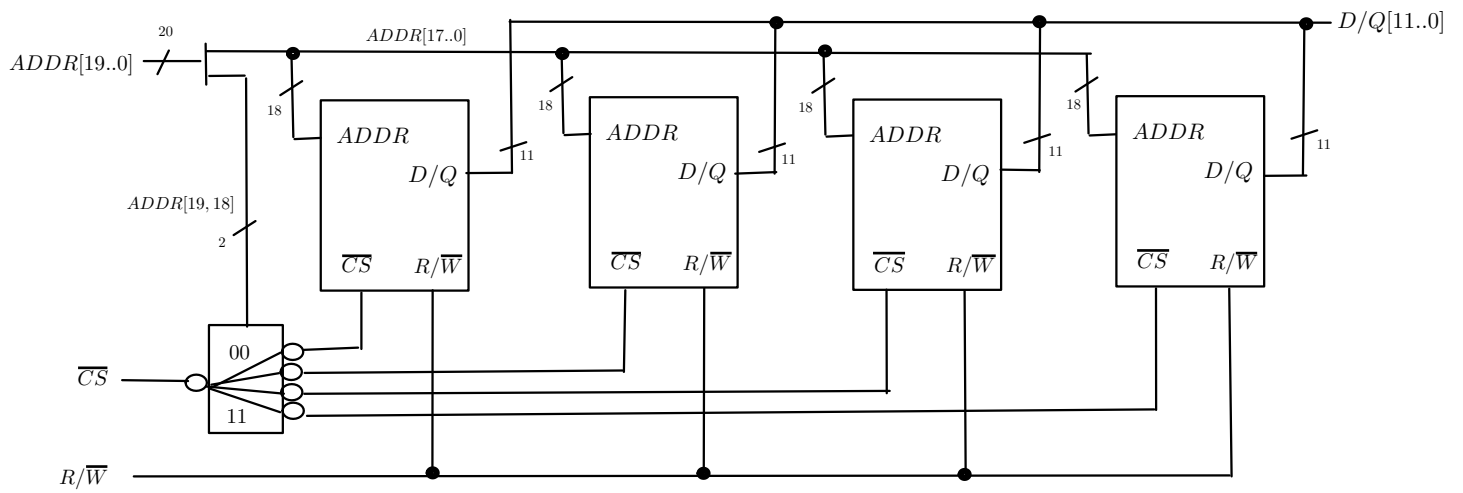
Raddoppio numero di parole



Arrivo alla dimensione di parola richiesta (11 bit) per un modulo da 256k

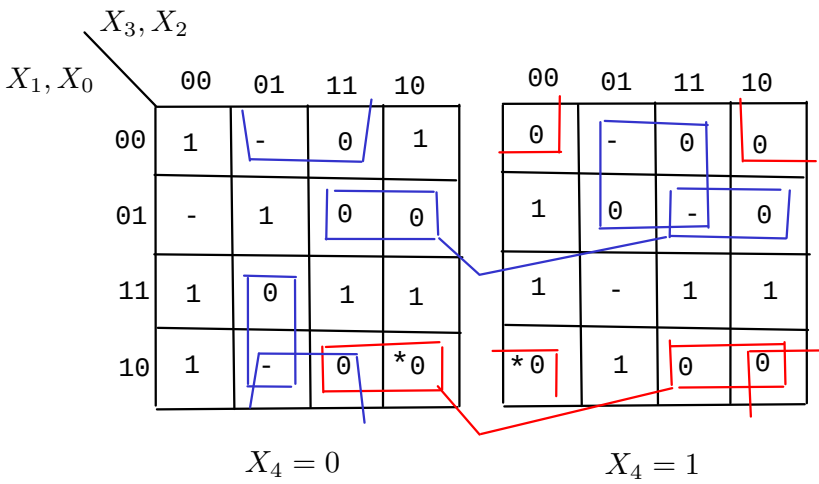


Quadruplico il numero di parole



Realizzare in forma NOR-NOR ottima una rete combinatoria non completamente specificata a 5 ingressi e 1 uscita, i cui mintermini siano $\{0, 2, 3, 5, 8, 11, 15, 17, 19, 22, 27, 31\}$ e l'insieme dei don't care sia $\{1, 4, 6, 20, 23, 29\}$. Indicare gli implicati essenziali.

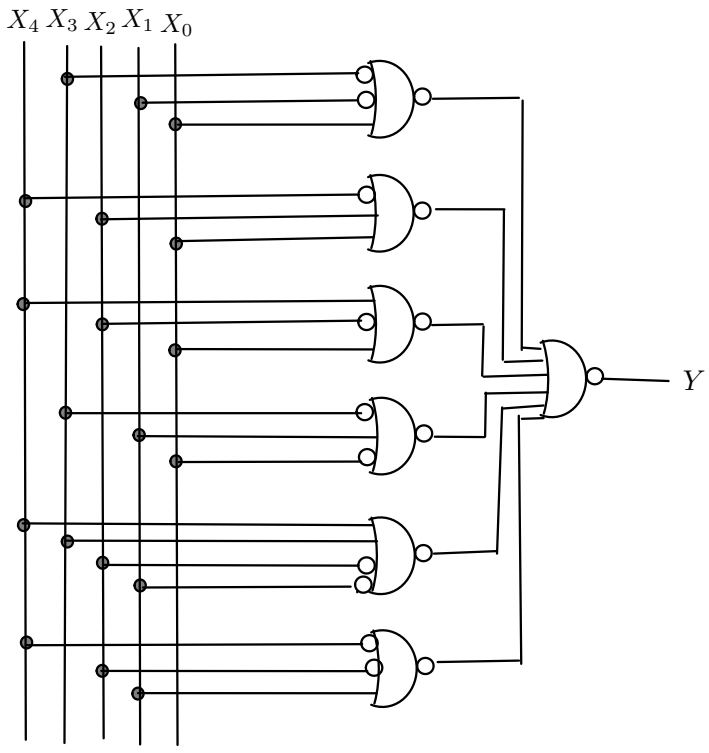
Colloco i mintermini in mappa e poi eseguo la sintesi cercando implicati essenziali e principali che coprono la funzione.



Forma PS (6 implicati di cui 2 essenziali, con *)

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

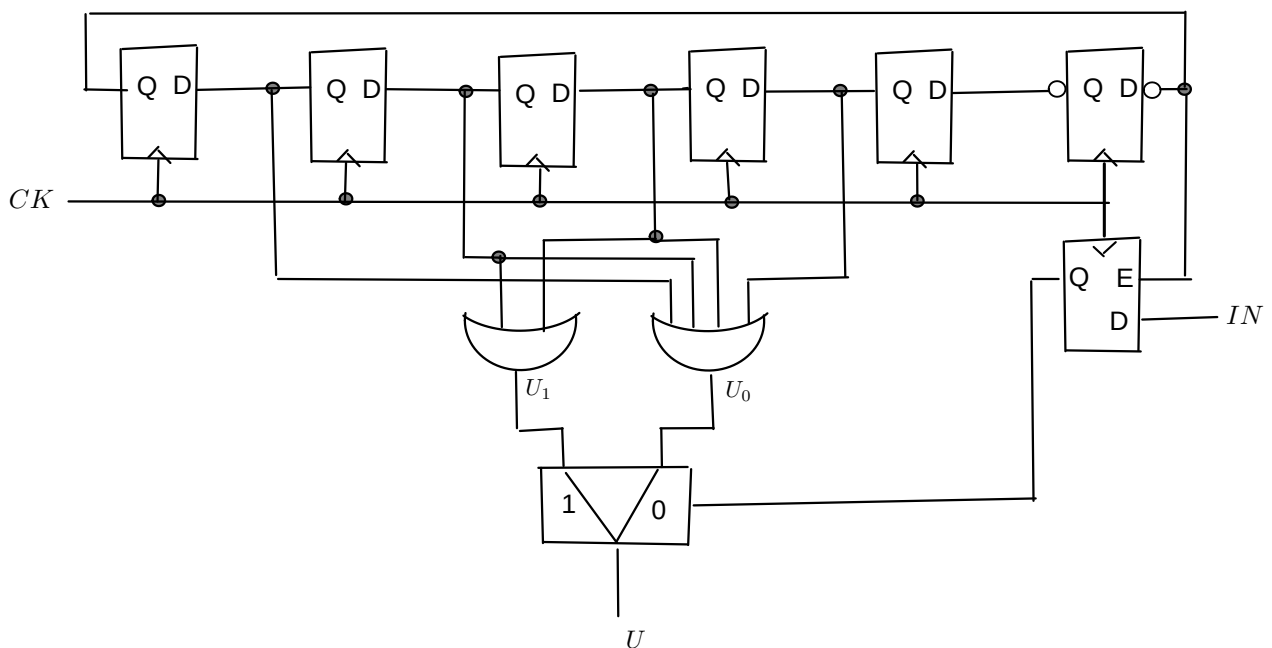
Schema logico a NOR



5

Progettare una macchina sequenziale sincrona secondo il modello di Moore con un ingresso e una uscita in grado di generare una forma d'onda di periodo $6T_{clk}$, il cui ciclo di lavoro valga $1/3$ se l'ingresso letto in corrispondenza della fine di ciascun ciclo vale 1 e $2/3$ altrimenti; la forma d'onda prodotta deve presentare l'impulso alto al centro del periodo (le sequenze sono rispettivamente 001100 con ingresso alto e 011110 con ingresso basso).

Cerchiamo una soluzione basata su un ring counter modulo 6, in grado di generare sequenze one-hot, a partire dalle quali produrre le 2 uscite richieste. Il comando che decide la struttura del periodo viene campionato al termine del periodo del contatore. Uso D-FF che all'accensione hanno uscita nulla garantita.

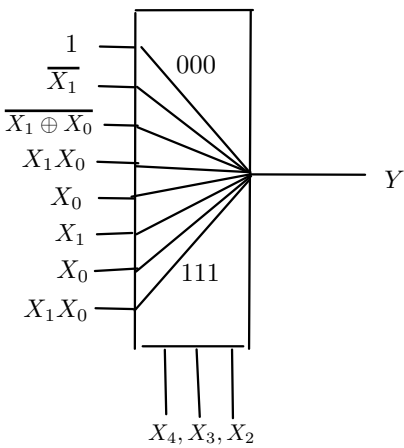


6

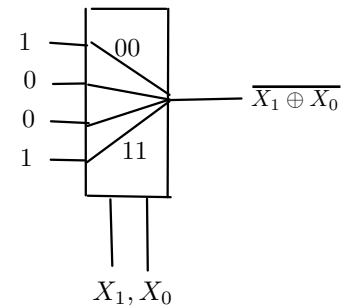
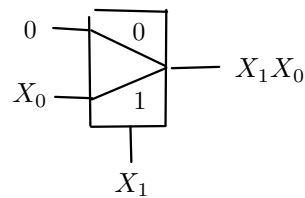
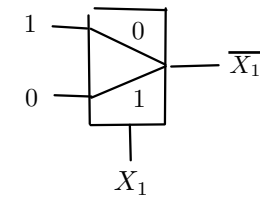
Realizzare la funzione dell'esercizio 4, se possibile, usando al massimo un multiplexer 8:1, 3 multiplexer 4:1 e 5 multiplexer 2:1.

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

Usa il mux 8:1 per selezionare le colonne. In ingresso avrò funzioni delle 2 rimanenti variabili.



Usa gli altri mux per ottenere le funzioni necessarie.



Avanzano 2 mux 4:1 e 3 mux 2:1.