

SCHEDA ASE1909		Data: 22 Novembre 2019
Cognome	Nome	

ESERCIZIO N°1

8 punti

Realizzare una subroutine `char_maxcount` per il microcontrollore AVR XMEGA256A3BU che individua il carattere che si ripete più spesso nella stringa puntata da `Y`, ponendo in `R16` il codice del carattere e in `R17` il numero di ripetizioni. Se più caratteri si ripetono la stessa quantità di volte, indicare quello il cui codice è maggiore. La stringa è costituita da un primo byte che contiene la lunghezza della stringa stessa (da 0 per la stringa nulla a un massimo di 255) seguito dai caratteri codificati ASCII su 8 bit (quindi 1 byte per carattere).

ESERCIZIO N°2

5 punti

Disegnare lo schema logico in forma normale a minimo numero di letterali (scegliendo la migliore tra SP e PS) della funzione combinatoria che avendo in ingresso i bit di un numero x , valuta il bit B_2 della codifica BCD della terza cifra decimale della radice quadrata di $x + 3$.

ESERCIZIO N°3

5 punti

Realizzare, usando esclusivamente multiplexer 2:1, la rete combinatoria proposta nell'esercizio 2. Cercare di minimizzare il numero di multiplexer impiegati.

ESERCIZIO N°4

4 punti

Disegnare lo schema logico di un sequenziatore in grado di implementare il seguente microcodice. `R` è lo stato iniziale e va codificato con 0.

```

A:  IF L THEN R ELSE A; OP = 101
B:  IF L THEN R ELSE G; OP = 100
C:  IF M THEN F ELSE B; OP = 110
D:  IF M THEN G ELSE F; OP = 011
E:  IF L THEN E ELSE C; OP = 111
F:  IF M THEN D ELSE R; OP = 010
G:  IF L THEN A ELSE D; OP = 000
R:  IF M THEN B ELSE E; OP = 001

```

ESERCIZIO N°5

6 punti

Progettare una macchina di Moore con due ingressi (le cifre binarie di un numero x) e una uscita, che viene posta a 0 quando l'ingresso è 0 per 2 volte consecutive e viene posta a 1 (mantenendo poi tale valore fino alla condizione di azzeramento) se (e solo se) gli ingressi assumono di seguito (anche se non necessariamente in cicli consecutivi) i valori 1, 2, 3.

ESERCIZIO N°6

5 punti

Dati i numeri $X = \sqrt{8} + \pi$, $Y = -\sqrt{\pi}$, $Z = -25/17$ e ipotizzando che sia accettabile un errore assoluto in modulo minore o uguale a 10^{-2}

- a) Determinare il numero minimo di bit che permette in generale di rappresentare in C2 numeri con lo stesso ordine di grandezza di quelli proposti.
- b) Determinare la rappresentazione in virgola fissa C2 dei numeri dati usando il numero minimo di bit che raggiungere la specifica nel caso particolare proposto.
- c) Se si usa il microcontrollore AVR XMEGA analizzato durante il corso, quale errore assoluto $(\hat{x} - x)$ si commette nel rappresentare (C2) in un registro i numeri di cui sopra? (usare 4 cifre significative)

1

Realizzare una subroutine `char_maxcount` per il microcontrollore AVR XMEGA256A3BU che individua il carattere che si ripete più spesso nella stringa puntata da Y, ponendo in R16 il codice del carattere e in R17 il numero di ripetizioni. Se più caratteri si ripetono la stessa quantità di volte, indicare quello il cui codice è maggiore. La stringa è costituita da un primo byte che contiene la lunghezza della stringa stessa (da 0 per la stringa nulla a un massimo di 255) seguito dai caratteri codificati ASCII su 8 bit (quindi 1 byte per carattere).

```
char_maxcount:
push R18 //N lunghezza stringa
push R19 //contatore per la ricerca del codice
push R20 //codice da cercare
push R21 //occorrenze del codice
push R22 //appoggio per il carattere dalla stringa
push XL //appoggio per il puntatore
push XH

ldi R16,255 //risultato max codice se stringa nulla
ldi R17,0 //risultato nessuna occorrenza se stringa nulla
ld R18,Y+ //carica la lunghezza di stringa
tst R18
breq end //stringa nulla
clr R20 //esegui scansione dei codici a partire da codice 0
loop:
mov R19,R18 //copia N nel contatore
movw XH:XL,YH:YL //copia il puntatore Y a stringa
clr R21 //inizializza le occorrenze per il codice in esame
loop1:
ld R22,X+
cp R22,R20 //vede se il carattere coincide con quello da cercare
brne poi0
inc R21
poi0: dec R19
brne loop1
cp R21,R17 //vede se questo codice ha avuto le stesse occorrenze o più
brcs poi1
mov R16,R20 //nuovo codice più frequente
mov R17,R21 //occorrenze del nuovo codice
poi1: inc R20
brne loop //ripeti per il codice successivo fino a tornare a codice 0

end:
sbiw YH:YL,1 //ripristina Y
pop XH
pop XL
pop R22
pop R21
pop R20
pop R19
pop R18
ret
```

2

Determino la tabella di verità della funzione

n	$n+3$	$\sqrt{n+3}$	$3^{2 \text{ dec}}$	B_2
0	3	1.73205...	2	0
1	4	2.00000...	0	0
2	5	2.23606...	6	1
3	6	2.44948...	9	0
4	7	2.64575...	5	1
5	8	2.82842...	8	0
6	9	3.00000...	0	0
7	10	3.16227...	2	0
8	11	3.31662...	6	1
9	12	3.46410...	4	1
10	13	3.60555...	5	1
11	14	3.74165...	1	0
12	15	3.87298...	2	0
13	16	4.00000...	0	0
14	17	4.12310...	3	0
15	18	4.24264...	2	0

SP

$X_3 X_2$

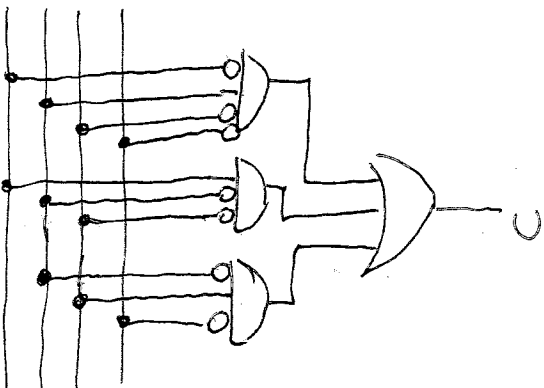
$X_1 X_0$	00	01	11	10
00	0 ⁰	1 ⁴	0 ¹²	1 ⁸
01	0 ¹	0 ⁵	0 ¹³	1 ⁹
11	0 ³	0 ⁷	0 ¹⁵	0 ¹¹
10	1 ²	0 ⁶	0 ¹⁴	1 ¹⁰

$X_3 X_2$

$X_1 X_0$	00	01	11	10
00	0	1*	0	1
01	0	0	0	1*
11	0	0	0	0
10	1*	0	0	1

10 mintermi

$X_3 X_2 X_1 X_0$



$X_3 X_2$ PS

$X_1 X_0$	00	01	11	10
00	0*	1	0*	1
01	0	0	0	1
11	0	0	0	0*
10	1	0*	0	1

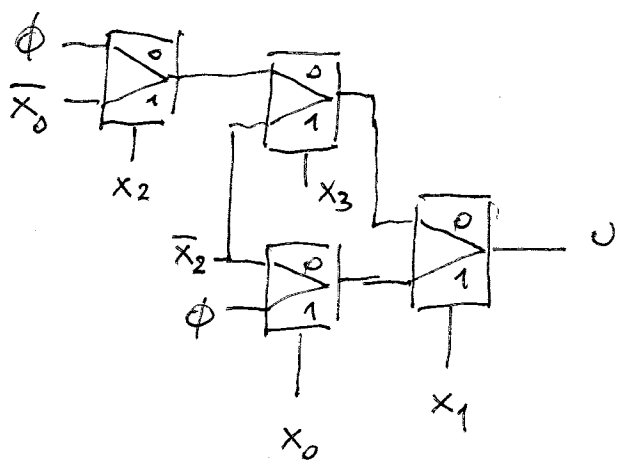
11 mintermi

$$U = \bar{X}_3 \bar{X}_2 \bar{X}_1 \bar{X}_0 + X_3 \bar{X}_2 \bar{X}_1 + \bar{X}_2 X_1 \bar{X}_0$$

(come SP)

3

Sintesi con RWX 2:1
(Una delle scelte possibili)



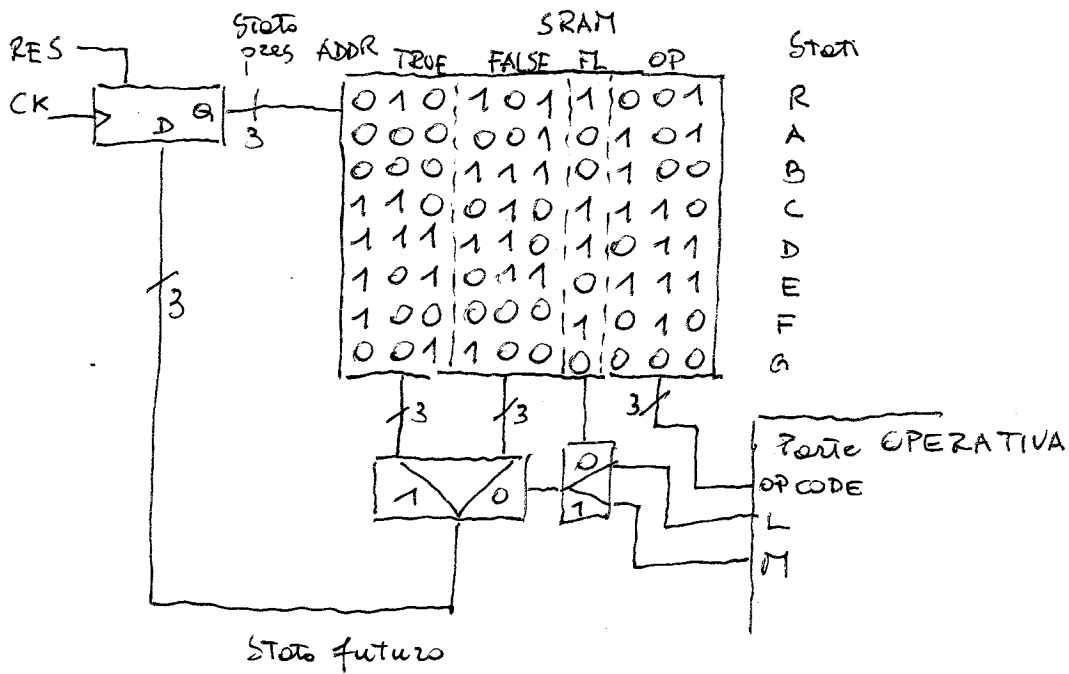
④ Sequenziatore

Scego architettura e ROM, che non presenta particolari requisiti.

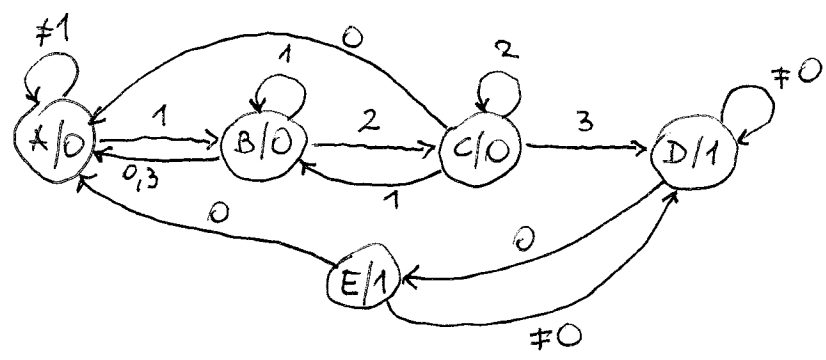
Codifica stati

R : 000 (come richiesto)
 A : 001
 B : 010
 C : 011
 D : 100
 E : 101
 F : 110
 G : 111

Flag L: 0
 M: 1



5) Macchine a STATI (Moore)

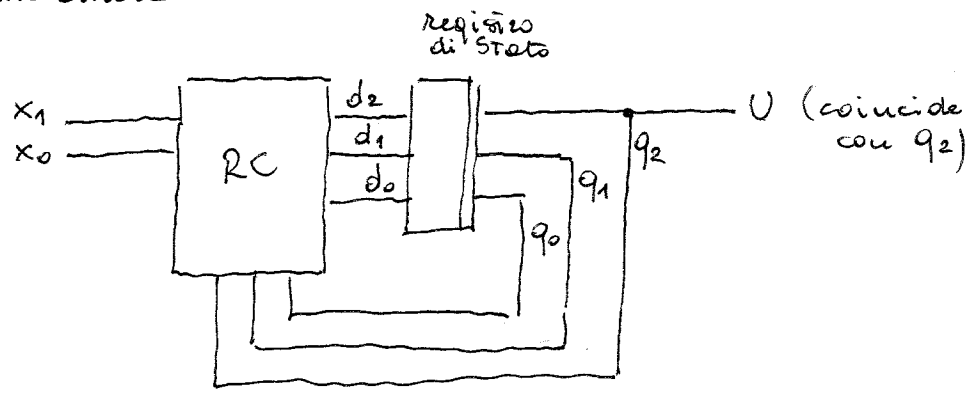


Codifica stati

A	000
B	001
C	011
D	111
E	110

$q_2 q_1 q_0$

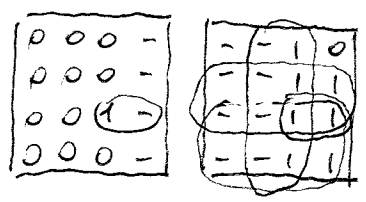
Architettura



Mappe di RC

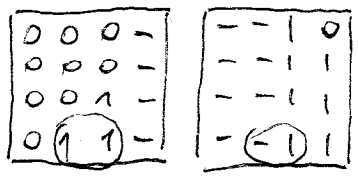
x_1, x_0	$q_1 q_0 \quad q_2 = 0$				$q_2 = 1$			
	00	01	11	10	00	01	11	10
00	000	000	000	-	-	-	110	000
01	001	001	001	-	-	-	111	111
11	000	000	111	-	-	-	111	111
10	000	011	011	-	-	-	111	111

d_2



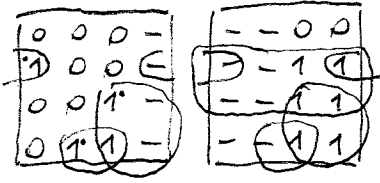
$$d_2 = x_1 x_0 q_1 + q_2 q_0 + q_2 x_0 + q_2 x_1$$

d_1



$$d_1 = d_2 + q_0 x_1 x_0$$

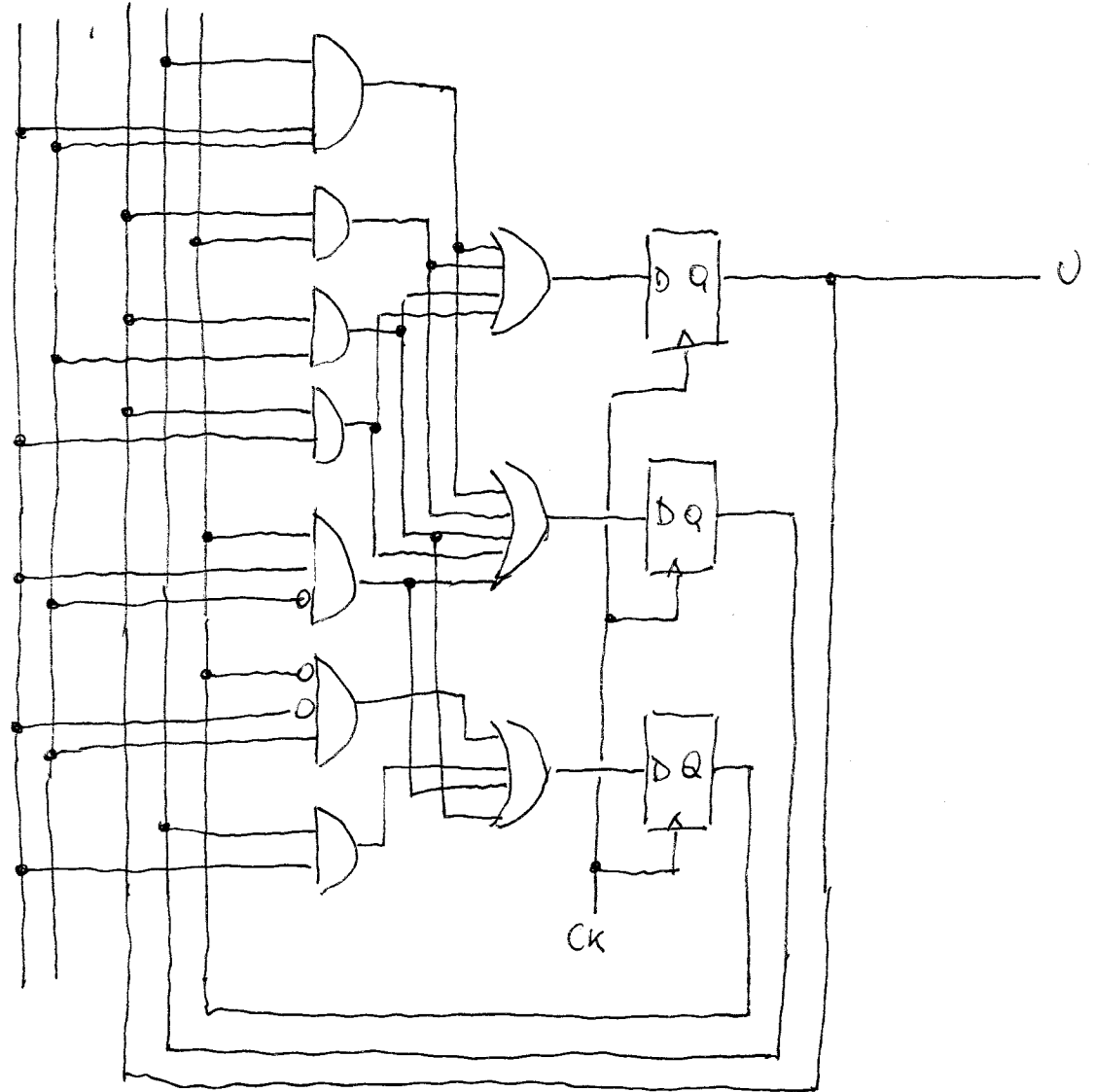
d3



$$d_0 = \bar{q}_0 \bar{x}_1 x_0 + q_1 x_1 + q_0 x_1 \bar{x}_0 + q_2 x_0$$

Scheme logico (con riutilizzo di implicanti)

$x_1 x_0 q_2 q_1 q_0$



⑥ Rappresentazioni

Per determinare la rappresentazione che soddisfa il requisito occorre valutare l'ODG dei numeri proposti

$$X \approx 5,9700 \dots \quad Y = -1,7724 \dots \quad Z = -1,4705 \dots$$

I valori sono tutti compresi tra -8 e $+7$ quindi:

a) Una rappresentazione C2 [4.6] con arrotondamento soddisfa il requisito proposto IN GENERALE, avendo errore assoluto $|E| \leq 2^{-7} < 10^{-2}$

b) Si determina la rappresentazione dei tre valori e si vede se è possibile rinunciare a cifre frazionarie senza aumentare l'errore.

$\hat{X} = 0101.1111:0$	dif.	$-0,0012698$	OK	
$\hat{Y} = 1110.0011:1$	ecc	$0,0068289$	(si può togliere	
$\hat{Z} = 1110.1000:0$	ecc	$0,0018382$	OK	$1/64: OK$)
Quindi per i numeri dati basta [4.5]			con errore accettabile	

c) Valuto la rappresentazione C2 [4.4] per i registri del μC

$\tilde{X} = 0110.0000$	$0,02998$	(4 cifre sig.)
$\tilde{Y} = 1110.0100$	$0,02245$	
$\tilde{Z} = 1110.1000$	$-0,02941$	

Come si vede l'errore con 8 cifre è maggiore di 10^{-2} .

Note: procedura per il calcolo delle codifiche C2 [4.6] con la calcolatrice

- 1) Calcolare il u^o con tutta la precisione disponibile
- 2) Se negativo, sommare $2^4 = 16$
- 3) Dividere per $2^6 = 64$
- 4) Passare in binario (10 cifre)