

Determinare il cubo dei seguenti numeri relativi (C2) rappresentati in notazione frazionale [1.7].

Il risultato deve essere espresso a sua volta in C2 con notazione frazionale [1.7], approssimandolo con il codice che dà il minimo errore, che deve essere indicato usando strettamente 4 cifre significative.

$$\begin{aligned} A &= 0b01010101 = 85/128 = 0,6640625 \\ B &= 0b10000000 = -1 \\ C &= 0b11111101 = -3/128 = -0,0234375 \end{aligned}$$

$$A^3 \simeq 0,292837619781$$

$$B^3 = -1$$

$$C^3 \simeq -0,0000128746032715$$

Per rappresentare nuovamente in C2 [1.7] moltiplico per 128 (sommo 128 se negativo) e arrotondo all'intero più vicino (nel caso in cui abbia sommato 128 metto 1 nel MSb).

$$\hat{A}^3 = 0b00100101 = 0,2890625$$

$$|\hat{A}^3 - A^3| \simeq 0,003775$$

$$\hat{B}^3 = 0b10000000 = -1$$

$$|\hat{B}^3 - B^3| = 0$$

$$\hat{C}^3 = 0b00000000 = 0$$

Il valore è negativo, ma più vicino a 0 che a -1/128

$$|\hat{C}^3 - C^3| \simeq 0,00001287$$

```
/* Realizzare un programma per il microcontrollore AVR XMEGA256A3BU,
che legge continuamente la locazione di memoria
all'indirizzo dello spazio di memoria "ingresso",
interpretandola come valore C2 a 8 bit, e pone in uscita
a partire dalla locazione "uscita" il cubo del valore letto,
sempre in C2, su 3 byte. */
```

```
/* Range del risultato:  $-2^{21}..2^{14}*(2^7-1)$ ;
3 byte sono più che sufficienti
con range di rappresentazione  $-2^{23}..2^{23}-1$  */
```

```
cubo_c2:
```

```
    loop:
    lds R16,ingresso
    muls R16,R16           //pone il quadrato, positivo, in R1:R0
    movw R19:R18,R1:R0    //salva il quadrato
    mulsu R16,R18
    movw R21:R20,R1:R0    //salva il parziale
    sbc R22,R22           //se negativo estende il segno
    mulsu R16,R19         //parte più significativa
    add R21,R0
    adc R22,R1            //somma nel risultato
    sts uscita,R20
    sts uscita+1,R21
    sts uscita+2,R22
    rjmp loop
```

Esprimere in forma SP a minimo numero di letterali la funzione combinatoria U delle 5 variabili  $X_4, X_3, X_2, X_1$  e  $X_0$ , non completamente specificata la cui tabella di verità è ottenuta eseguendo l'operazione riportata di seguito. Nelle caselle per cui è risultato il valore 2 si ponga un "don't care". Indicare gli implicanti essenziali, evidenziando un mintermine che motiva la proprietà individuata.

$$|(x + 7)^5|_3$$

Sfruttando le proprietà del modulo si ottiene subito:

x	U
0	1
1	2
2	0

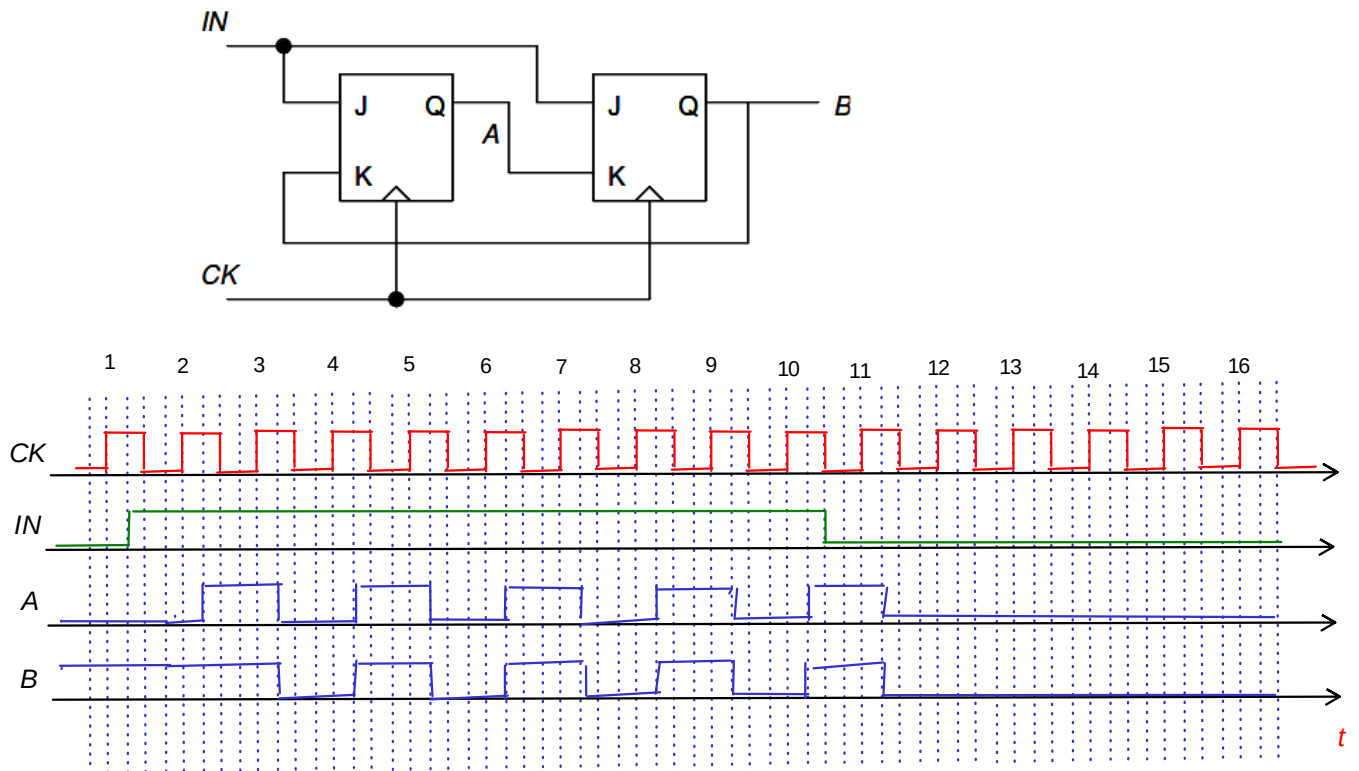
e questi valori si ripetono fino in fondo alla tabella di verità.

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

C'è un solo implicante essenziale. Poi occorrono 10 altri implicanti, tutti di ordine 1, uno per ciascun 1 della funzione.

$$\begin{aligned}
 U = & X_4 \overline{X_3} X_2 \overline{X_1} X_0 + \overline{X_4} \overline{X_3} \overline{X_2} \overline{X_1} + \overline{X_4} X_2 \overline{X_1} \overline{X_0} + \overline{X_4} X_3 \overline{X_1} X_0 + \\
 & + \overline{X_4} \overline{X_3} X_1 X_0 + \overline{X_4} X_2 X_1 X_0 + \overline{X_4} \overline{X_3} X_2 \overline{X_0} + \\
 & + X_4 X_3 \overline{X_1} \overline{X_0} + X_4 X_3 \overline{X_2} X_0 + X_4 \overline{X_3} \overline{X_2} X_1 + X_4 X_2 X_1 \overline{X_0}
 \end{aligned}$$

Determinare l'andamento delle grandezze A e B nel tempo.



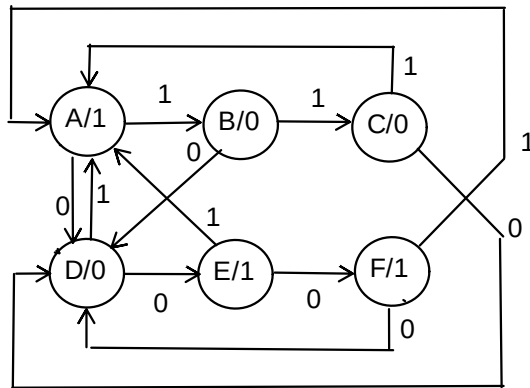
Si tratta di una macchina sincrona correttamente pilotata. Tutte le transizioni in uscita dai JK-FF avvengono dopo un  $t_{co}$  dal fronte del clock che ha campionato gli ingressi.

È quindi possibile determinare la sequenza delle uscite e quindi riportarla in grafico.

Ck	IN,B	IN,A	A+,B+ (il + indica il nuovo valore dopo il clock)
1	0,1	0,0	0,1
2	1,1	1,0	1,1
3	1,1	1,1	0,0
4	1,0	1,0	1,1
5	1,1	1,1	0,0
6	1,0	1,0	1,1
7	1,1	1,1	0,0
8	1,0	1,0	1,1
9	1,1	1,1	0,0
10	1,0	1,0	1,1
11	0,1	0,1	0,0
12	0,0	0,0	0,0
13	0,0	0,0	0,0
14	0,0	0,0	0,0
15	0,0	0,0	0,0
16	0,0	0,0	0,0

5

Progettare una macchina sincrona a un ingresso che, con ingresso 1, genera la sequenza periodica di 3 valori 100100100..., oppure, con ingresso 0, l'altra sequenza sempre di 3 valori 011011011..., ripartendo dopo ogni commutazione dell'ingresso dal valore iniziale della sequenza corrispondente.

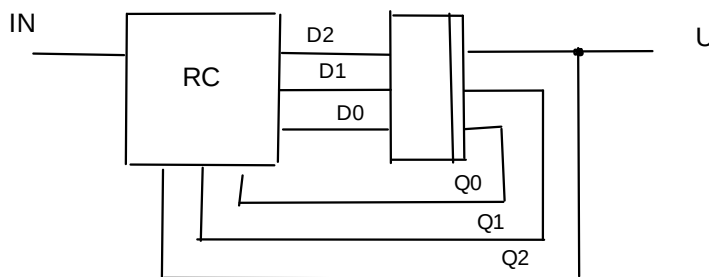


Scelgo modello di Moore  
Codifica stati (Q2,Q1,Q0)

- A 100
- B 001
- C 011
- D 000
- E 101
- F 111

U=Q2

Architettura



Sintesi della RC per lo stato futuro

Q1,Q0		Q2,IN			
		00	01	11	10
00	101	000	000	---	
01	100	011	100	---	
11	001	100	100	---	
10	000	111	000	---	

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

$$D_2 = \overline{Q_2} \overline{Q_0} + INQ_1 + Q_2 \overline{Q_1} Q_0$$

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

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

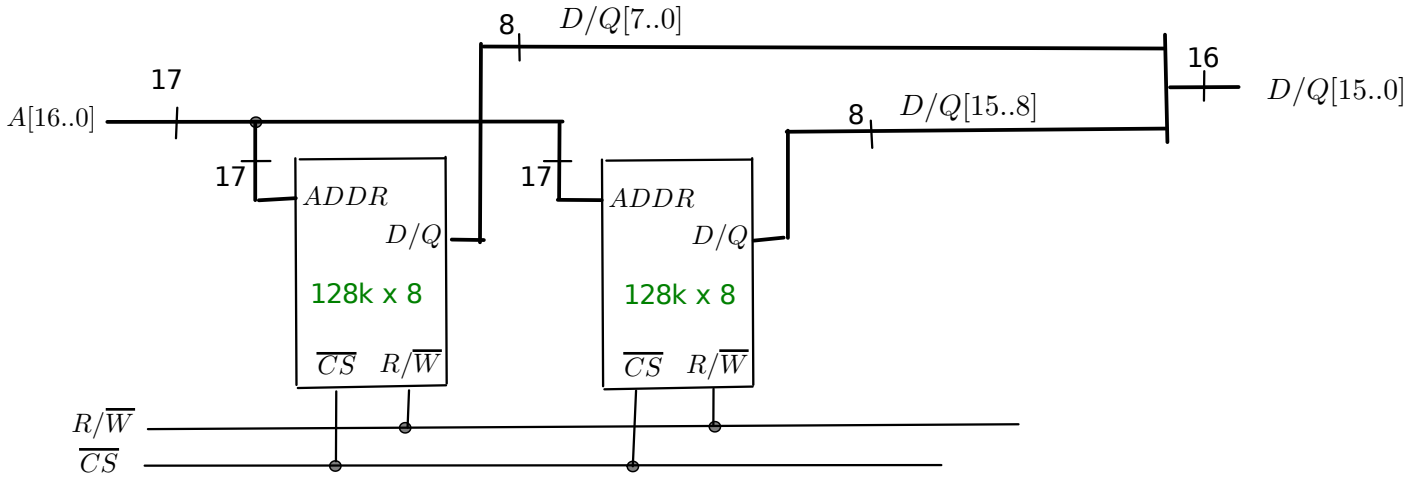
1	0	0	-
0	1	0	-
1	0	0	-
0	1	0	-

$$D_0 = \overline{IN} \overline{Q_2} \overline{Q_0} + IN\overline{Q_2} \overline{Q_1} Q_0 + \overline{IN} Q_2 \overline{Q_1} Q_0 + INQ_2 \overline{Q_1} \overline{Q_0}$$

6

Usando chip di memoria SRAM da 128k x 8 realizzare un modulo di memoria da 512k x 16.

Passaggio a 128k x 16



Modulo finale da 512k x 16

