

**ESERCIZIO N°1**

5 punti

$M$  è la matricola dello studente. Determinare i bit della rappresentazione binary 32 IEEE 754-2008 con la migliore approssimazione possibile dei seguenti numeri:

$M$ ,  $(3M)^6$ ,  $-(11M)^{-6}$

e valutare per ciascuno l'errore relativo  $\frac{\hat{x} - x}{x}$

**ESERCIZIO N°2**

8 punti

Realizzare una subroutine ritardo per il microcontrollore AVR XMEGA256A3BU che non altera alcun registro (compreso CPU\_SREG) e viene eseguita in  $M$  cicli di clock, da CALL (4 cicli) a RET (5 cicli) compresi.

**ESERCIZIO N°3**

5 punti

Progettare e disegnare lo schema logico di una rete sequenziale sincrona con un ingresso di abilitazione  $E$  (se  $E = 0$  lo stato della macchina non cambia) in grado di generare una sequenza periodica il cui periodo vale  $\{b_{11}, \bar{b}_5, b_9, b_7, \bar{b}_7, b_5, b_0\}$ . I valori  $b_0, b_1, \dots$  sono le cifre binarie della codifica BCD di  $M$ ;  $b_0$  è il bit meno significativo.

**ESERCIZIO N°4**

5 punti

Lo studente proponga una funzione combinatoria a scelta con 5 variabili di ingresso  $X_4, X_3, X_2, X_1, X_0$ , nella cui tabella di verità siano presenti 13 "1", 11 "0" e 8 "-". La funzione deve avere 2 e 2 soli implicanti principali di ordine 3 (compresi i don't care considerati come "1"), oltre ad altri implicanti di ordine minore e ovviamente nessuno di ordine maggiore. Sintetizzare la funzione in forma PS ottima (minimo numero di letterali), indicando esplicitamente quali sono gli implicati essenziali (non è richiesto il disegno dello schema logico).

**ESERCIZIO N°5**

5 punti

Lo studente realizzi la funzione dell'esercizio precedente avendo a disposizione invertitori e 12 multiplexer 2:1.

## ESERCIZIO N°6

5 punti

Determinare il diagramma di flusso del seguente sequenziatore. Agli stati deve essere attribuito il nome  $S_i$ , con  $i$  corrispondente al codice dello stato. Le 8 righe della SRAM contengono i seguenti 8 valori:

$$| M + 7777 |_{512}$$

$$| M^2 + 253793 |_{512}$$

$$| 3M + 111111 |_{512}$$

$$| 73M^3 + 3 |_{512}$$

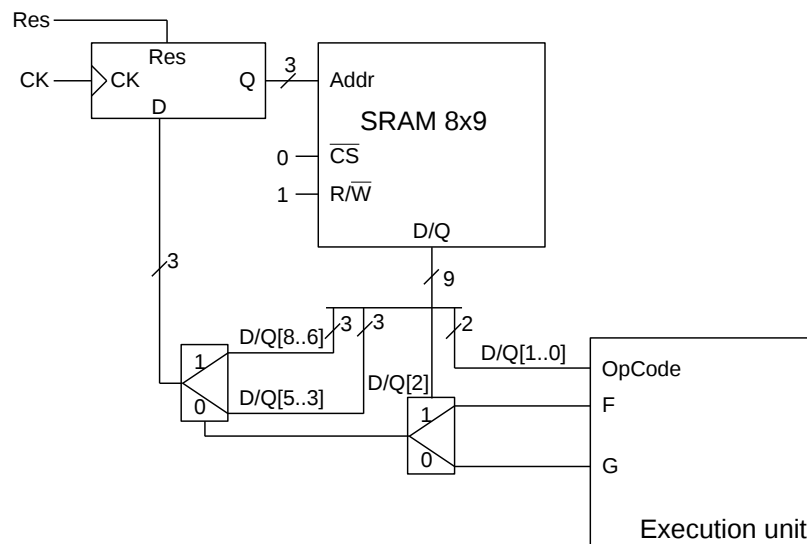
$$| 8^M + 7M |_{512}$$

$$| M! + 493 |_{512}$$

$$| 5M + 178 |_{512}$$

$$| 11M + 33 |_{512}$$

Sarebbe stato possibile realizzare il sequenziatore con un contatore a caricamento parallelo, risparmiando sulla dimensione della SRAM (spiegare)? La notazione  $| x |_{512}$  indica l'operazione " $x$  modulo 512".



① Determinare i bit delle rappresentazioni binary 32 di  
(a titolo di esempio,  $M = 500000$ )

$M$  essendo minore di  $2^{24}$ , avrà una rappresentazione  
ESATA

$$S = \phi \quad E = 145 \quad (0x91) \quad T = 7611392 \quad (0x742400)$$

$$0 | 10010001 | 111010000100100000000000 |$$

$$E_2 = \phi$$

$(3M)^6$  è vicino al limite superiore di rappresentabilità, ma  
è ancora rappresentabile senza problemi

$$S = \phi \quad E = 250 \quad (0xFA) \quad T = 597011 \quad (0x091C13)$$

$$0 | 11111010 | 000100100011100000010011 |$$

$$E_2 = 2,7985 \cdot 10^{-8}$$

$-(11M)^{-6}$  è invece minore del più piccolo dei normalizzabili  
(in modulo), quindi va rappresentato con la legge  
dei non norm.

$$\begin{cases} x_{norm} = (-1)^S \cdot 2^{E-127} (1 + T \cdot 2^{-23}) \\ x_{nonorm} = (-1)^S \cdot 2^{-149} \cdot T \quad (\text{con } E = \phi) \end{cases}$$

$$S = 1 \quad E = 0 \quad T = 25781 \quad (0x0064B5)$$

$$0 | 00000000 | 00000000110010010110101 |$$

$$E_2 = 1,5063 \cdot 10^{-5}$$

## 2

Realizzare una subroutine ritardo per il microcontrollore AVR XMEGA256A3BU che non altera alcun registro (compreso CPU\_SREG) e viene eseguita in M cicli di clock, da CALL (4 cicli) a RET (5 cicli) compresi.

```
ritardo:
.equ k1=0x10 //valori che garantiscono il ritardo richiesto
.equ k2=0x00
.equ k3=0xDD
    push R16 //registro di appoggio per il salvataggio di CPU_SREG
    lds R16,CPU_SREG
    push R16 //registri di appoggio per il ritardo
    push R17
    push R18
    ldi R16,k1 //primo byte
    ldi R17,k2
    ldi R18,k3
loop:
    dec R16 //valore iniziale k1 poi 0 alle successive iterazioni
    brne loop
    dec R17 //valore iniziale k2 poi 0 alle successive iterazioni
    brne loop
    dec R18
    brne loop
    nop // x nop per tuning fine (0<=x<=2)
    nop
    pop R18
    pop R17
    pop R16
    sts CPU_SREG,R16 //ripristina flag
    pop R16
    ret
```

```
/*
calcolo dei coefficienti k1, k2 e k3
ritardo complessivo in cicli di clock
```

```
9 rcall e ret
12 push e pop
4 lds e sts
3 ldi per inizializzazione
x nop eventuali
```

```
ritardo del loop composito (hp: k1, k2, k3 diversi da 0)
 $3*k1-1+3+(3*256-1+3)*(k2-1)-1+3+((3*256-1+3)*256-1+3)*(k3-1)=$ 
 $=3*(k1+256*k2+256^2*k3)-197888$ 
```

```
sommando tutto
 $R=3*(k1+256*k2+256^2*k3)-197860+x$ 
```

```
nel caso in cui per esempio M=5000000 si ottiene (passando in hex)
x=0 e k1=0xAC, k2=0x8C, k3=0x03
```

```
*/
```

③ la macchina richiesta può essere ottenuta tramite una architettura di MOORE costituita da un CONTATORE (down) modulo 7 con abilitazione E, seguito da una rete combinatoria RC2 per la produzione dell'uscita -  
 con  $M=500000$  la sequenza è  
 0100100

Sintesi di RC2

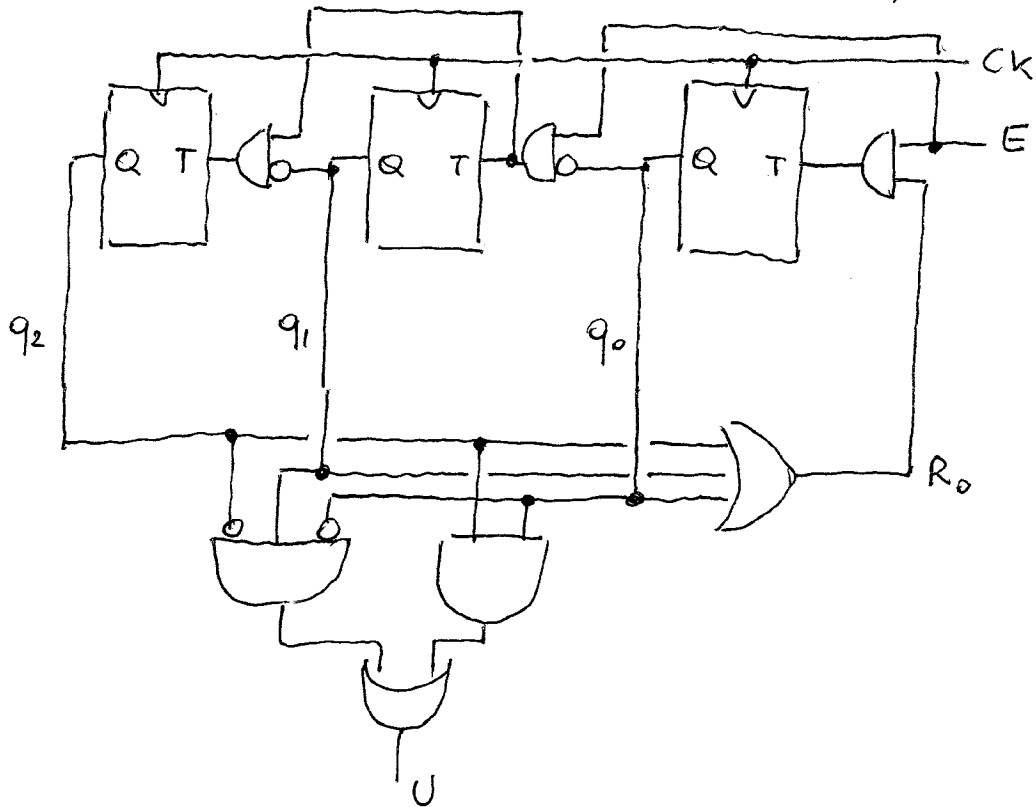
(6543210 stato contatore)

Scelgo down perché lievemente più semplice

	$q_1 q_0$	00	01	11	10
$q_2$	0	0 <sup>0</sup>	0 <sup>1</sup>	0 <sup>3</sup>	1 <sup>2</sup>
	1	4	5	-	6
		0	1	-	0

$$U = \bar{q}_2 q_1 \bar{q}_0 + q_2 q_0$$

Scheme logico (per passare da 000 a 110, --B)



4

Funzione combinatoria

13 "1"

11 "0"

8 "-"

2 soli implicenti di ordine 3

$$\bar{x}_4 \bar{x}_3 ; \bar{x}_4 \bar{x}_2$$

$x_3 x_2$		$x_1 x_0$			
		00	01	11	10
$x_4$	0	1	1	1	1
	1	0	0	0	0

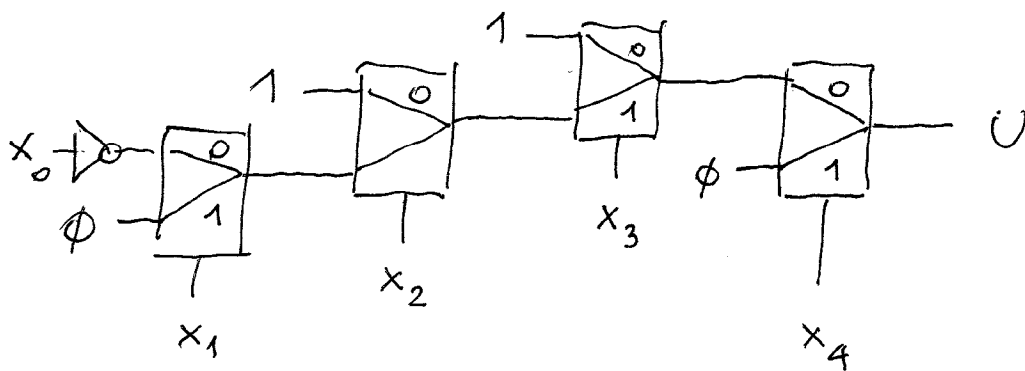
$x_4 = 0$                        $x_4 = 1$

si trova PS ottima, con tutti implicenti essenziali

$$\bar{x}_4 (\bar{x}_3 + \bar{x}_2 + \bar{x}_0) (\bar{x}_3 + \bar{x}_2 + \bar{x}_1)$$

- il pallino indica un mintermine coperto in esclusione dell'implicente essenziale che lo contiene

⑤ Della mappa si ricava direttamente:



6

Determino il contenuto della SRAM, separando i campi collegati al microcodice. Si noti che, con  $M$  grande,  $|M!|_{512} = \phi$  e  $|8^M|_{512} = \phi$  ( $M=500000$ ) per esempio

- $S_0 : 385$
- $S_1 : 353$
- $S_2 : 359$
- $S_3 : 3$
- $S_4 : 480$
- $S_5 : 493$
- $S_6 : 82$
- $S_7 : 129$

	Vero	Falso	OPCODE	
$S_0$	$S_6$	$S_0$	0	01
$S_1$	$S_5$	$S_4$	0	01
$S_2$	$S_5$	$S_4$	1	11
$S_3$	$S_0$	$S_0$	0	11
$S_4$	$S_7$	$S_4$	0	00
$S_5$	$S_7$	$S_5$	1	01
$S_6$	$S_1$	$S_2$	0	10
$S_7$	$S_2$	$S_0$	0	01

flag  
 0: G  
 1: F

non si può ridurre la complessità della SRAM con l'uso di un contatore perché, sia con FLAG VERO, sia con FLAG FALSO, NON ESISTE PERCORSO CICLICO COMPLETO. Infatti una condizione NECESSARIA (non SUFFICIENTE) per avere il percorso CICLICO COMPLETO è che in una delle colonne con lo STATO FUTURO TUTTI GLI STATI siano DIVERSI tra loro, e questo NON AVVIENE.

DIAGRAMMA di FLUSSO

