

Il testo deve essere riconsegnato nella cartellina. Non è ammessa la consultazione degli appunti e dei compiti precedenti. Si possono consultare i data sheet. **Non usare il colore rosso nello svolgimento.**

**ESERCIZIO N°1**

5 punti

I seguenti valori esadecimali costituiscono un testo codificato in UTF8. Individuare quanti caratteri sono presenti esplicitando i singoli codici dei caratteri, evidenziando quali sono quelli che appartengono alla tabella ASCII e cosa rappresentano.

C3 A0 C3 A9 C3 AC C3 B2 41 20 42 39 30 C4 8D C3 AD 20 5A 5A 5A 0A

**USASCII code chart**

Bits					0 0 0	0 0 1	0 1 0	0 1 1	1 0 0	1 0 1	1 1 0	1 1 1
b <sub>4</sub>	b <sub>3</sub>	b <sub>2</sub>	b <sub>1</sub>	Column	0	1	2	3	4	5	6	7
↓	↓	↓	↓	Row	0	1	2	3	4	5	6	7
0	0	0	0	0	NUL	DLE	SP	0	@	P	`	p
0	0	0	1	1	SOH	DC1	!	1	A	Q	a	q
0	0	1	0	2	STX	DC2	"	2	B	R	b	r
0	0	1	1	3	ETX	DC3	#	3	C	S	c	s
0	1	0	0	4	EOT	DC4	\$	4	D	T	d	t
0	1	0	1	5	ENQ	NAK	%	5	E	U	e	u
0	1	1	0	6	ACK	SYN	&	6	F	V	f	v
0	1	1	1	7	BEL	ETB	'	7	G	W	g	w
1	0	0	0	8	BS	CAN	(	8	H	X	h	x
1	0	0	1	9	HT	EM	)	9	I	Y	i	y
1	0	1	0	10	LF	SUB	*	:	J	Z	j	z
1	0	1	1	11	VT	ESC	+	;	K	[	k	{
1	1	0	0	12	FF	FS	,	<	L	\	l	
1	1	0	1	13	CR	GS	-	=	M	]	m	}
1	1	1	0	14	SO	RS	.	>	N	^	n	~
1	1	1	1	15	SI	US	/	?	O	_	o	DEL

**ESERCIZIO N°2**

8 punti

Realizzare un sottoprogramma per il microcontrollore AVR XMEGA256A3BU che inserisce il valore contenuto in R16 in un vettore di 64 byte, al posto del valore il cui indice è il contenuto di R0 (modulo 64). Il vettore ha 64 elementi in sequenza, e la prima componente (di indice 0) è puntata da Z.

**ESERCIZIO N°3**

5 punti

Realizzare in forma PS ottima una rete combinatoria a 5 ingressi ( $X_4, X_3, X_2, X_1$  e  $X_0$ ) e una uscita che indica con 1 i casi (e solo quelli) in cui  $X$  soddisfa almeno uno dei seguenti requisiti: è un multiplo di 4, di 6, di 9, di 13, un quadrato o un cubo perfetto.

Indicare tutti gli **implicati essenziali** della funzione, evidenziando un maxtermine che giustifica l'indicazione di essenziale.

## **ESERCIZIO N°4**

4 punti

Realizzare la rete combinatoria dell'esercizio precedente facendo uso di mux 2:1, cercando di ridurre il numero, evitando l'uso di blocchi non necessari (duplicati, mux con ingressi identici, ecc.).

## **ESERCIZIO N°5**

6 punti

Disegnare lo schema logico di una rete sequenziale sincrona che, quando abilitata, pone in uscita una sequenza Gray a 4 bit. Progettare la macchina in modo che la variazione dell'uscita non presenti glitch.

## **ESERCIZIO N°6**

5 punti

Disegnare lo schema di un contatore Johnson modulo 5, con uscite one-hot, usando flip-flop con valore di accensione garantito nullo. Nel caso invece in cui il valore iniziale dei flip-flop sia casuale, determinare se esiste un numero di cicli di clock finito entro il quale il funzionamento del contatore è quello atteso.

Codici in UTF:

sono ASCII i byte che hanno b7=0

i byte con b7=1, b6=1, b5=0 iniziano un codice a 2 byte

i byte con b7=1, b6=0 sono di prosecuzione codice

I codici sono i bit tra quadre

C3 A0	Codice a 2 byte	110[0011] 10[100000]
C3 A9	Codice a 2 byte	110[0011] 10[101001]
C3 AC	Codice a 2 byte	110[0011] 10[101100]
C3 B2	Codice a 2 byte	110[0011] 10[110010]
41	Codice ASCII	"A"
20	Codice ASCII	<SP>
42	Codice ASCII	"B"
39	Codice ASCII	"9"
30	Codice ASCII	"0"
C4 8D	Codice a 2 byte	110[0100] 10[001101]
C3 AD	Codice a 2 byte	110[0011] 10[101101]
20	Codice ASCII	<SP>
5A	Codice ASCII	"Z"
5A	Codice ASCII	"Z"
5A	Codice ASCII	"Z"
0A	Codice ASCII	<LF>

16 caratteri

/\* Realizzare un sottoprogramma per il microcontrollore AVR XMEGA256A3BU che inserisce il valore contenuto in R16 in un vettore di 64 byte, al posto del valore il cui indice è il contenuto di R0 (modulo 64). Il vettore ha 64 elementi in sequenza, e la prima componente (di indice 0) è puntata da Z.

\*/

insert:

```
push R17
push ZL
push ZH
mov R17,R0
andi R17,0x3F //esegue modulo 64, eliminando i bit di peso 64 e 128
add ZL,R17 //posiziona Z sul valore da modificare
ldi R17,0 //annulla R0 senza toccare il flag C
adc ZH,R17 //inserisce eventuale riporto della prima somma
st Z,R16 //inserisce il valore
pop ZH
pop ZL
pop R17
ret
```

Realizzare in forma PS ottima una rete combinatoria a 5 ingressi ( $X_4, X_3, X_2, X_1$  e  $X_0$ ) e una uscita che indica con 1 i casi (e solo quelli) in cui  $X$  soddisfa almeno uno dei seguenti requisiti: è un multiplo di 4, di 6, di 9, di 13, un quadrato o un cubo perfetto.

Indicare tutti gli implicati essenziali della funzione, evidenziando un maxtermine che giustifica l'indicazione di essenziale.

		$X_3, X_2$							
		00	01	11	10	00	01	11	10
$X_1, X_0$	00	1 <sub>0</sub>	1 <sub>4</sub>	1 <sub>12</sub>	1 <sub>8</sub>	1 <sub>16</sub>	1 <sub>20</sub>	1 <sub>28</sub>	1 <sub>24</sub>
	01	1 <sub>1</sub>	0* <sub>5</sub>	1 <sub>13</sub>	1 <sub>9</sub>	0* <sub>17</sub>	0 <sub>21</sub>	*0 <sub>29</sub>	1 <sub>25</sub>
	11	0 <sub>3</sub>	0 <sub>7</sub>	0 <sub>15</sub>	0 <sub>11</sub>	0 <sub>19</sub>	0 <sub>23</sub>	0 <sub>31</sub>	1 <sub>27</sub>
	10	0* <sub>2</sub>	1 <sub>6</sub>	0* <sub>14</sub>	0 <sub>10</sub>	1 <sub>18</sub>	0* <sub>22</sub>	1 <sub>30</sub>	1 <sub>26</sub>
		$X_4=0$				$X_4=1$			

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

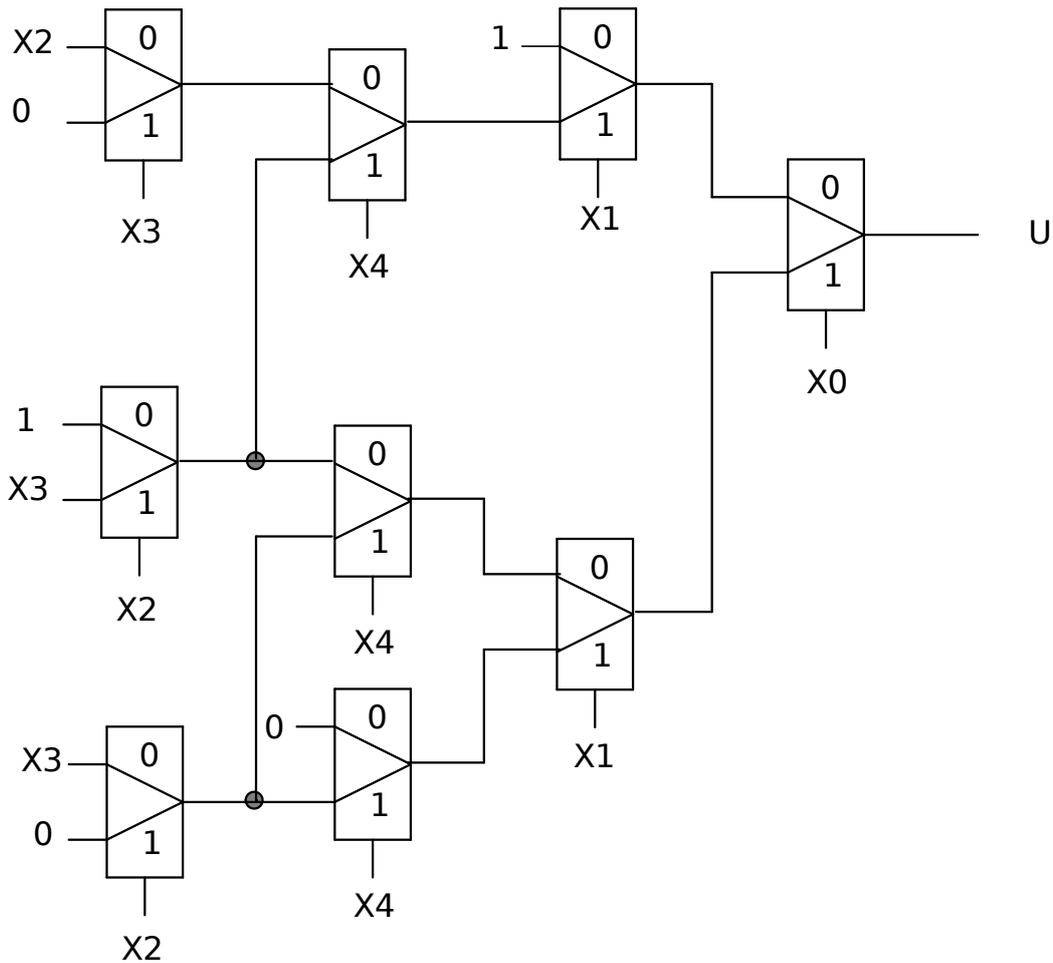
L'asterisco (\*) evidenzia il maxtermine che rende essenziale l'implicato che lo contiene. Tutti gli implicati considerati sono essenziali, quindi la sintesi ottima è univoca.

4

Realizzare la rete combinatoria dell'esercizio precedente facendo uso di mux 2:1 cercando di ridurre il numero, evitando l'uso di blocchi non necessari (duplicati, mux con ingressi identici, ecc.).

X3,X2		X4=0				X4=1			
		00	01	11	10	00	01	11	10
X1,X0	00	1 <sub>0</sub>	1 <sub>4</sub>	1 <sub>12</sub>	1 <sub>8</sub>	1 <sub>16</sub>	1 <sub>20</sub>	1 <sub>28</sub>	1 <sub>24</sub>
	01	1 <sub>1</sub>	0 <sub>5</sub>	1 <sub>13</sub>	1 <sub>9</sub>	0 <sub>17</sub>	0 <sub>21</sub>	0 <sub>29</sub>	1 <sub>25</sub>
	11	0 <sub>3</sub>	0 <sub>7</sub>	0 <sub>15</sub>	0 <sub>11</sub>	0 <sub>19</sub>	0 <sub>23</sub>	0 <sub>31</sub>	1 <sub>27</sub>
	10	0 <sub>2</sub>	1 <sub>6</sub>	0 <sub>14</sub>	0 <sub>10</sub>	1 <sub>18</sub>	0 <sub>22</sub>	1 <sub>30</sub>	1 <sub>26</sub>

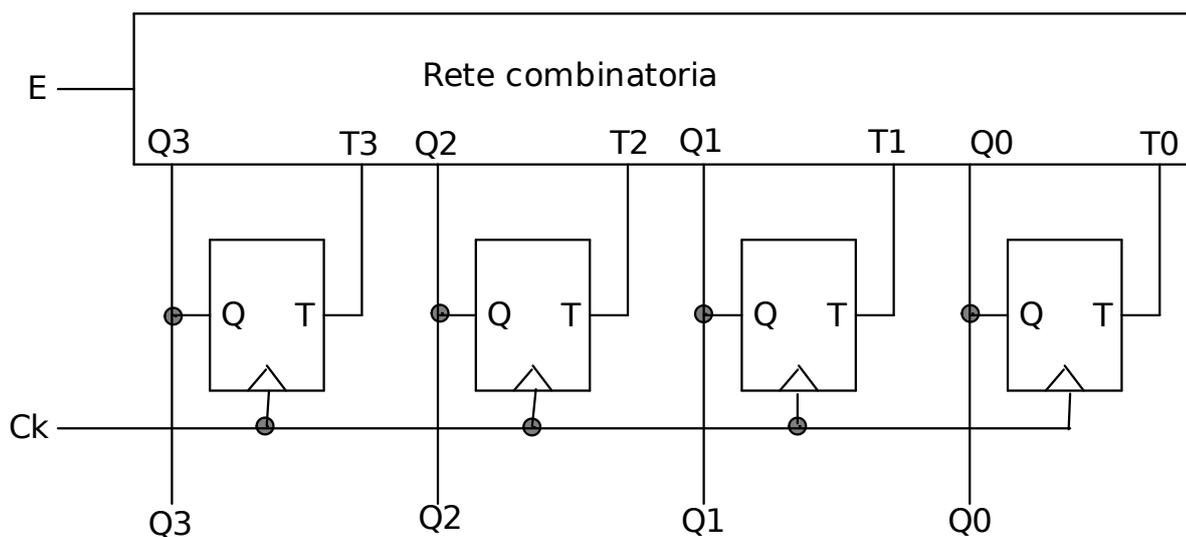
Una possibile soluzione è:



5

Disegnare lo schema logico di una rete sequenziale sincrona che, quando abilitata, pone in uscita una sequenza Gray a 4 bit. Progettare la macchina in modo che la variazione dell'uscita non presenti glitch.

Per evitare glitch usiamo una architettura di Mealy sincrona, costituita da 4 T-FF e una rete combinatoria con uscita one-hot (o nulla), che impone il cambiamento di un solo T-FF (o nessuno, se  $E=0$ )



Sequenza Gray e bit  $T_i$  da porre a 1

0000	T0
0001	T1
0011	T0
0010	T2
0110	T0
0111	T1
0101	T0
0100	T3
1100	T0
1101	T1
1111	T0
1110	T2
1010	T0
1011	T1
1001	T0
1000	T3

		G3,G2			
		G1,G0	00	01	11
T0	00	1		1	
	01		1		1
	11	1		1	
	10		1		1

$$T_0 = E \overline{(G_3 \oplus G_2 \oplus G_1 \oplus G_0)}$$

		G3,G2			
		G1,G0	00	01	11
T1	00				
	01	1		1	
	11		1		1
	10				

$$T_1 = E G_0 \overline{(G_3 \oplus G_2 \oplus G_1)}$$

		G3,G2			
		G1,G0	00	01	11
T2	00				
	01				
	11				
	10	1		1	

$$T_2 = E G_1 \overline{G_0} \overline{(G_3 \oplus G_2)}$$

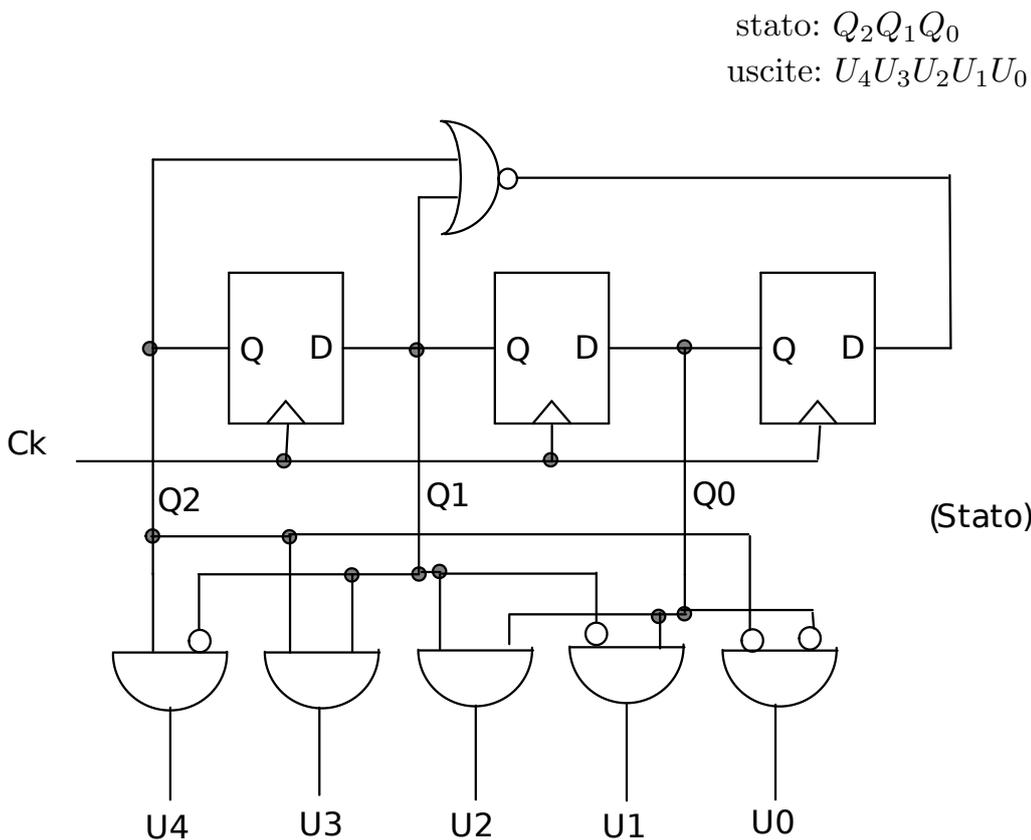
		G3,G2			
		G1,G0	00	01	11
T3	00				1
	01				
	11				
	10				

$$T_3 = E G_3 \overline{G_2} \overline{G_1} \overline{G_0}$$

6

Disegnare lo schema di un contatore Johnson modulo 5, con uscite one-hot, usando flip-flop con valore di accensione garantito nullo. Nel caso invece in cui il valore iniziale dei flip-flop sia casuale, determinare se esiste un numero di cicli di clock finito entro il quale il funzionamento del contatore è quello atteso.

Si useranno 3 D-FF con reazione invertente tramite NOR (per eliminare uno stato). L'uscita one-hot sarà ottenuta con una rete combinatoria.



Valori dello stato validi

000 0  
001 1  
011 3  
110 6  
100 4

Evoluzione da stati diversi a caso

010 2  
100 4ok  
  
101 5  
010 2ok (vedi sopra)  
  
111 7  
110 6ok

Si torna sempre in uno stato della sequenza