SCHEDA **ASE2402**	Data: **25 Gennaio 2024**

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, in questo caso, cosa rappresentano.
F0 9F 98 81 EF B8 8F C3 88 20 74 65 73 74 6F 20 63 6F 6E 20 C3 A0 C3 A9 C3 B9 F0 9F 98 90 EF B8 8F 0A

USASCII code chart

b_4	b_3	b_2	b_1	Row ↓ / Column →	0 (0 0 0)	1 (0 0 1)	2 (0 1 0)	3 (0 1 1)	4 (1 0 0)	5 (1 0 1)	6 (1 1 0)	7 (1 1 1)
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

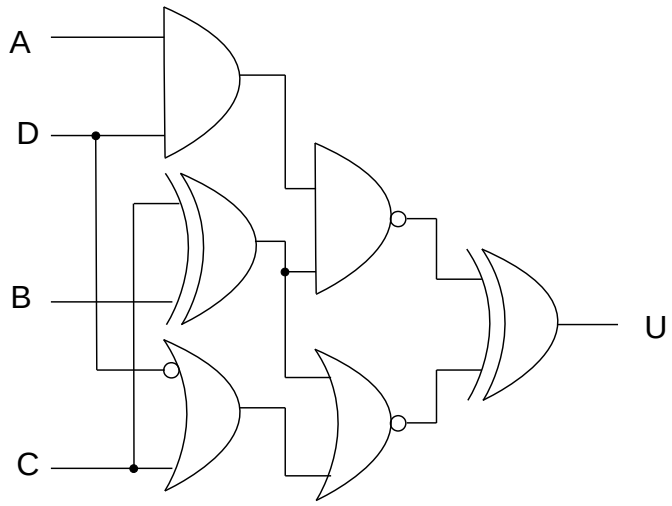
8 punti

Scrivere un sottoprogramma nel linguaggio assembly della famiglia XMEGA AVR che scrive nelle locazioni di memoria da 0x3000 a 0x33AB (compresi gli estremi) i valori in sequenza ottenuti dalla formula $|7^m|_{256}$ dove *m* è la differenza tra l'indirizzo della cella da modificare e quello della prima cella del blocco.

ESERCIZIO N°3

5 punti

Realizzare in forma PS ottima la seguente rete combinatoria a 4 ingressi e una uscita; indicare quali implicati sono essenziali, motivando la scelta.



ESERCIZIO N°4

5 punti

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

ESERCIZIO N°5

5 punti

Spiegare la differenza tra le istruzioni MUL, MULS e MULSU del linguaggio assembly della famiglia AVR ed esprimere come valore intero con segno in base 10 il valore del risultato delle tre diverse istruzioni (dove è contenuto?) quando gli argomenti sono due registri il cui contenuto è rispettivamente 0xBF e 0xFB.

ESERCIZIO N°6

5 punti

Progettare una rete di Moore a due ingressi (A e l'abilitazione E) e 3 uscite (Q_2, Q_1 e Q_0) in grado di generare (se abilitata) una sequenza in cui il valore successivo è pari al valore precedente più 5, se l'ingresso è 0, e invece è pari al valore precedente meno 6 se l'ingresso è 1. Le somme e le differenze si intendono modulo 8.

1

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,
in questo caso, cosa rappresentano.
F0 9F 98 81 EF B8 8F C3 88 20 74 65 73 74 6F 20 63 6F 6E
20 C3 A0 C3 A9 C3 B9 F0 9F 98 90 EF B8 8F 0A

```
F0       9F       98       81       Codice 0x1F601
11110000 10011111 10011000 10000001 00001111011000000001
EF       B8       8F                Codice 0xFE0F
11101111 10111000 10001111          1111111000001111
C3       88                        Codice 0xC8
11000011 10001000                  00011001000
20                                 ASCII <SP>
74                                 ASCII t
65                                 ASCII e
73                                 ASCII s
74                                 ASCII t
6F                                 ASCII o
20                                 ASCII <SP>
63                                 ASCII c
6F                                 ASCII o
6E                                 ASCII n
20                                 ASCII <SP>
C3       A0                        Codice 0xE0
11000011 10100000                  00011100000
C3       A9                        Codice 0xE9
11000011 10101001                  00011101001
C3       B9                        Codice 0xF9
11000011 10111001                  00011111001
F0       9F       98       90       Codice 0x1F610
11110000 10011111 10011000 10010000 00001111011000010000
EF       B8       8F                Codice 0xFE0F
11101111 10111000 10001111          1111111000001111
0A                                 ASCII <LF>
```

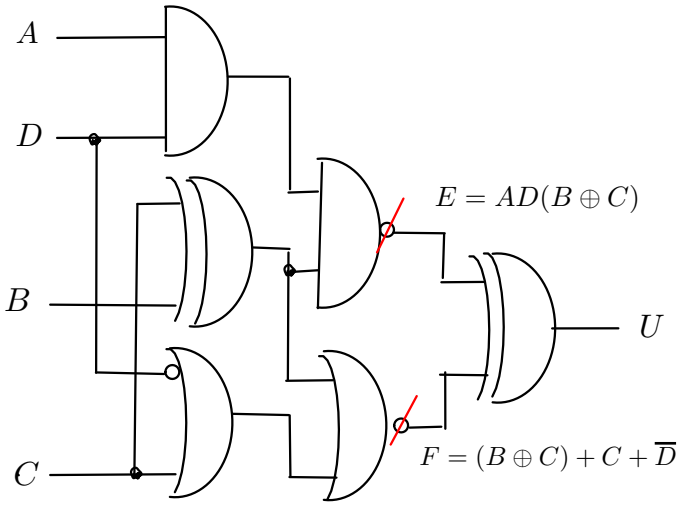
2

```
/* Scrivere un sottoprogramma nel linguaggio assembly della famiglia
   XMEGA AVR che scrive nelle locazioni di memoria da 0x3000 a 0x33AB
   (compresi gli estremi) i valori in sequenza ottenuti dalla formula
   7^m modulo 256
   dove m è la differenza tra l'indirizzo della cella da modificare e
   quello della prima cella del blocco.*/

write_7power:
 push R0                //risultato prodotto
 push R1
 push R16               //variabile di appoggio
 push XL
 push XH
 ldi XL,low(0x3000)
 ldi XH,high(0x3000)
 ldi R16,1              //mette in R0 il valore iniziale 1
 mov R0,R16
 ldi R16,7              //mette la base della potenza
 loop:
  st X+,R0
  mul R0,R16            //valore da caricare nella prossima cella
  cpi XL,low(0x33AB+1)
 brne loop
  cpi XH,high(0x33AB+1)
 brne loop
 pop XH
 pop XL
 pop R16
 pop R1
 pop R0
ret
```

3

Realizzare in forma PS ottima la seguente rete combinatoria a 4 ingressi e una uscita; indicare quali implicati sono essenziali, motivando la scelta.



Posso modificare lo schema, eliminando le NOT. Trovo le mappe di E ed F e quindi la mappa di U.

$E = AD(B \oplus C)$

BC \ AD	00	01	11	10
00	0	0	0	0
01	0	0	1	0
11	0	0	0	0
10	0	0	1	0

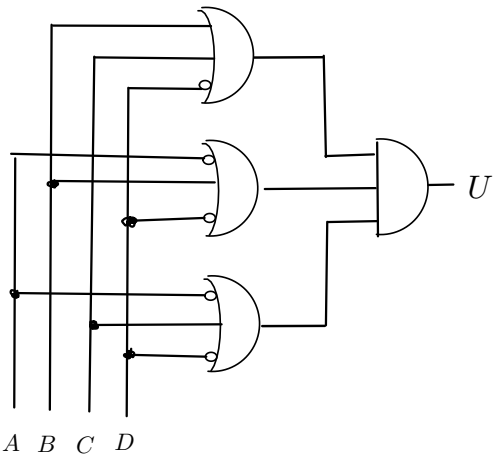
$F = (B \oplus C) + C + \overline{D}$

BC \ AD	00	01	11	10
00	1	0	0	1
01	1	1	1	1
11	1	1	1	1
10	1	1	1	1

$U = (E \oplus F)$

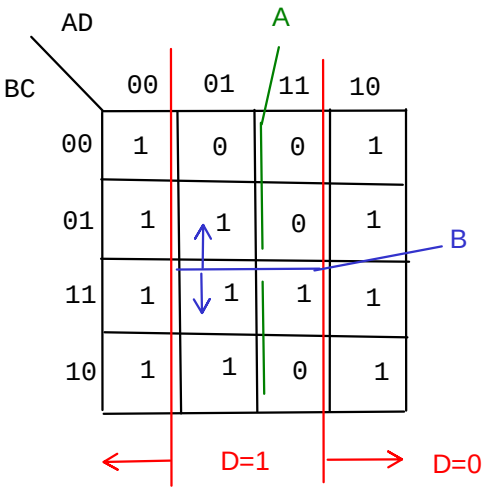
BC \ AD	00	01	11	10
00	1	*0	0	1
01	1	1	*0	1
11	1	1	1	1
10	1	1	0*	1

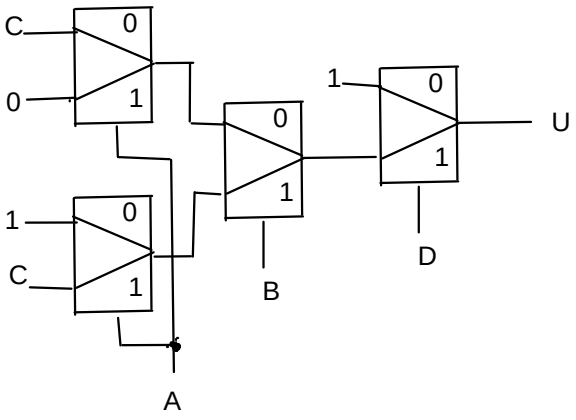
Forma PS



4

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





5

Spiegare la differenza tra le istruzioni MUL, MULS e MULSU del linguaggio assembly della famiglia AVR ed esprimere come valore intero con segno in base 10 il valore del risultato delle tre diverse istruzioni (dove è contenuto?) quando gli argomenti sono due registri il cui contenuto è rispettivamente 0xBF e 0xFB.

Le tre istruzioni eseguono la moltiplicazione tra grandezze contenute in due registri e pongono il risultato su 16 bit nella coppia di registri R1:R0.

Gli operandi vengono interpretati diversamente dalle 3 operazioni e presentano pure delle limitazioni.

MUL Rd,Rr
Gli operandi sono numeri binari senza segno e possono essere registri qualsiasi

MULS Rd,Rr
Gli operandi sono numeri con segno C2 e possono essere registri nel range R16-R31

MULSU Rd,Rr
Rd è un numero con segno (C2) e Rr senza segno.
Gli operandi possono essere registri nel range R16-R23

I valori assegnati, in base 10, corrispondono ai numeri
senza segno
Rd=0xBF=191; Rr=0xFB=251
con segno
Rd=0xBF=-65; Rr=0xFB=-5

quindi, in base 10:
MUL Rd,Rr // in R1:R0 viene 47941 (-17595 negativo, se considerato in C2)
MULS Rd,Rr // in R1:R0 viene 325
MULSU Rd,Rr // in R1:R0 viene -955

6

Progettare una rete di Moore a due ingressi (A e l'abilitazione E) e 3 uscite (Q2, Q1 e Q0)
in grado di generare (se abilitata) una sequenza in cui il valore successivo
è pari al valore precedente più 5, se l'ingresso è 0, e invece è pari al valore precedente meno 6
(ovvero più 2 considerando il modulo 8) se l'ingresso è 1.
Le somme e le differenze si intendono modulo 8.

Possiamo costruire la rete di Moore direttamente dalla definizione, considerando l'uscita uguale allo stato presente e valutando lo stato futuro come richiesto dal testo.
Per soddisfare il requisito sull'abilitazione, usiamo DE-FF.