

Cognome

Nome

ESERCIZIO N°1

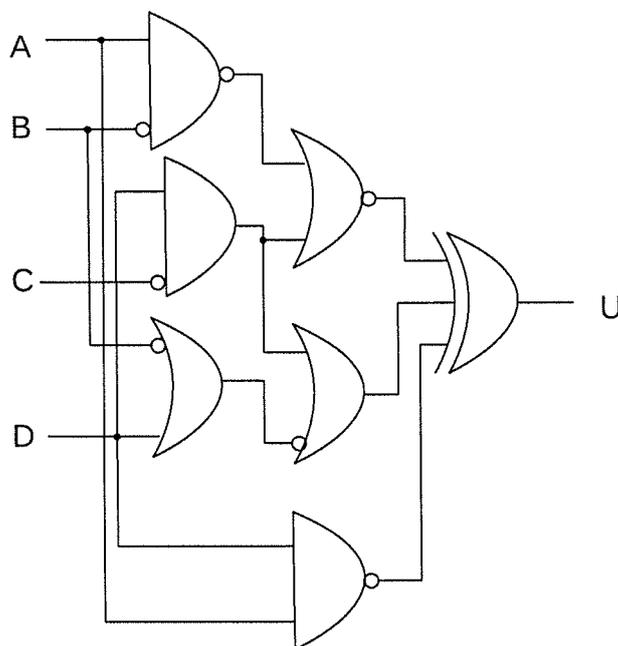
8 punti (2 per le verifiche di coerenza)

Realizzare una subroutine `substring_insert` per il microcontrollore AVR XMEGA256A3BU che inserisce la stringa puntata da `Y` dentro la stringa puntata da `X`, a partire dalla posizione successiva al carattere `N` (`N` contenuto in `R16`; se `N` è nullo la stringa inserita deve precedere tutti i caratteri; se `N` è uguale o maggiore della lunghezza `Nx`, la stringa viene inserita dopo l'ultimo carattere). 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). La stringa risultante, che deve essere aggiornata anche per quanto riguarda il nuovo valore della lunghezza che non potrà comunque superare il valore 255, finisce al posto di quella originale puntata da `X`.

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 descritta dallo schema seguente.



ESERCIZIO N°3

5 punti

Realizzare, se possibile con un approccio modulare, una rete combinatoria a 6 ingressi (2 numeri binari di 3 bit ciascuno) in grado di indicare con un valore alto in uscita quando uno dei due ingressi è maggiore o uguale all'altro.

ESERCIZIO N°4

4 punti

Disegnare lo schema logico di un registro universale a 4 bit in grado di implementare funzionalità simili ai comandi indicati di seguito, sulla base del valore di un segnale di controllo a 2 bit.

0: Load, 1: Shift dx, 2: Shift sx, 3: Clear

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 e viene posta a 1 (mantenendo poi tale valore fino alla condizione di azzeramento) se (e solo se) gli ingressi assumono immediatamente di seguito i valori 3, 2, 2, 1.

ESERCIZIO N°6

5 punti

Dati i numeri $X = \sqrt{7} + \pi$, $Y = -\sqrt{\pi}$, $Z = -23/15$ 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 `substring_insert` per il microcontrollore AVR XMEGA256A3BU che inserisce la stringa puntata da Y dentro la stringa puntata da X, a partire dalla posizione successiva al carattere N (N contenuto in R16; se N è nullo la stringa inserita deve precedere tutti i caratteri; se N è uguale o maggiore della lunghezza N_x , la stringa viene inserita dopo l'ultimo carattere). 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) La stringa risultante, che deve essere aggiornata anche per quanto riguarda il nuovo valore della lunghezza che non potrà comunque superare il valore 255, finisce al posto di quella originale puntata da X.

```
substring_insert:
/* Allocazione dei registri (senza ottimizzazione)
R2: 0   registro di appoggio nullo
R16: N  posizione di inserimento
R17: Nx lunghezza stringa X
R18: Ny lunghezza stringa Y
R19: Nt lunghezza stringa finale min(Nx+Ny,255)
R20: Ch carattere da trasferire
R21: N1 prima parte del risultato min(N,Nx)
R22: N2 seconda parte del risultato min(Ny,Nt-N1)
R23: N3 terza parte del risultato Nt-N1-N2
R25:R24 appoggio puntatore stringa Y
YH:YL  puntatore di lettura
ZH:ZL  puntatore di scrittura
*/
push R2 //valore nullo
push R17
push R18
push R19
push R20
push R21
push R22
push R23
push R24
push R25
push ZL
push ZH
clr R2
movw R25:R24,YH:YL //salva il puntatore Y
//calcola i parametri della nuova stringa
ld R17,X //in R17 Nx
ld R18,Y //in R18 Ny
mov R21,R16 //valuta N1=min(N,Nx)
cp R17,R21
brcc a1
mov R21,R17 //N1=Nx
a1:
mov R19,R17 //valuta Nt=min(Nx+Ny,255)
add R19,R18
brcc a2
ldi R19,255
a2:
st X,R19 //sistema la nuova lunghezza totale del risultato
mov R22,R19 //valuta N2=min(Nt-N1,Ny)
sub R22,R21
mov R23,R22 //pone R23=Nt-N1
```

```

cp R22,R18
brcs a3
mov R22,R18
a3:
sub R23,R22 //valuta N3=Nt-N1-N2

//calcola puntatori primo spostamento
movw ZH:ZL,XH:XL //puntatore di scrittura dal fondo della stringa finale
Z=X+1+Nt
adiw ZH:ZL,1
add ZL,R19
adc ZH,R2
movw YH:YL,ZH:ZL //puntatore di lettura Y=X+1+N1+N3=Z-N2
sub YL,R22
sbc YH,R2
//sposta la parte finale della prima stringa per far posto alla parte da
inserire
//dal fondo per non sovrascrivere caratteri utili
tst R23
breq next1
loop3:
    ld R20,-Y
    st -Z,R20
    dec R23
    brne loop3 //alla fine Z=(X+1+Nt-N3)
//sistema i puntatori per copiare la parte centrale
next1:
sub ZL,R22 //puntatore di scrittura Z=(X+1+Nt-N3)-N2
sbc ZH,R2
movw YH:YL,R25:R24 //puntatore di lettura Y=Y+1
adiw YH:YL,1
//inserisce la parte centrale, che può essere nulla
tst R22
breq next2
loop2:
    ld R20,Y+
    st Z+,R20
    dec R22
    brne loop2
//ripristino registri
next2:
movw YH:YL,R25:R24
pop ZH
pop ZL
pop R25
pop R24
pop R23
pop R22
pop R21
pop R20
pop R19
pop R18
pop R17
pop R2
ret

```

3

$$F = \bar{A} + B + D\bar{C}$$

$$G = D\bar{C} + B\bar{D}$$

$$H = AD$$

$$U = F \oplus G \oplus H = \bar{F} \oplus G \oplus \bar{H}$$

①

	AB	00	01	11	10
00		1	1	1	0
01		1	1	1	1
11		1	1	1	0
10		1	1	1	0

②

	AB	00	01	11	10
00		0	1	1	0
01		1	1	1	1
11		0	0	0	0
10		0	1	1	0

③

	AB	00	01	11	10
00		0	0	0	0
01		0	0	1	1
11		0	0	1	1
10		0	0	0	0

④

	AB	00	01	11	10
00		1*	0	0	0
01		0	0	1*	1
11		1*	1	0	1
10		1	0	0	0

Sintesi SP: 4 implicati di ordine 1 (di cui 3 essenziali)
12 letteri

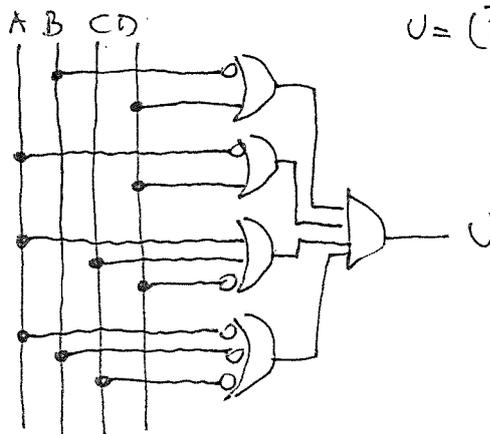
⑤

	AB	00	01	11	10
00		1	0	0	0*
01		0*	0	1	1
11		1	1	0*	1
10		1	0*	0	0

Sintesi PS: 2 implicati di ordine 2
2 implicati di ordine 1
TUTTI ESSENZIALI
(vedi * sul MAXTERMINE coperto in eccessivo)

10 letteri
SOLUZIONE OTTIMA

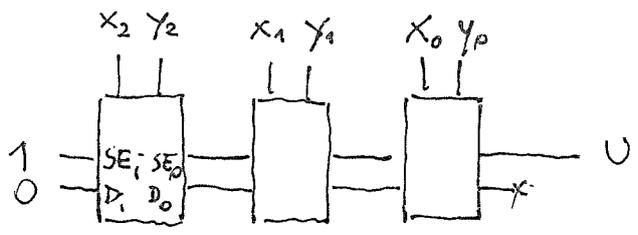
$$U = (\bar{B} + D)(\bar{A} + D)(A + C + \bar{D})(\bar{A} + \bar{B} + \bar{C})$$



3) $x \geq y$

il confronto si può fare con 3 blocchi uguali a 4 ingre

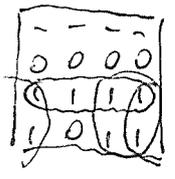
Soluzione MSB first



	x y			
SE, D	00	01	11	10
00	-	-	-	-
01	01	01	01	01
11	11	11	11	11
10	10	01	10	11

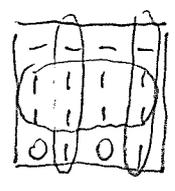
questo caso non può mai darsi

SE_0, D_0

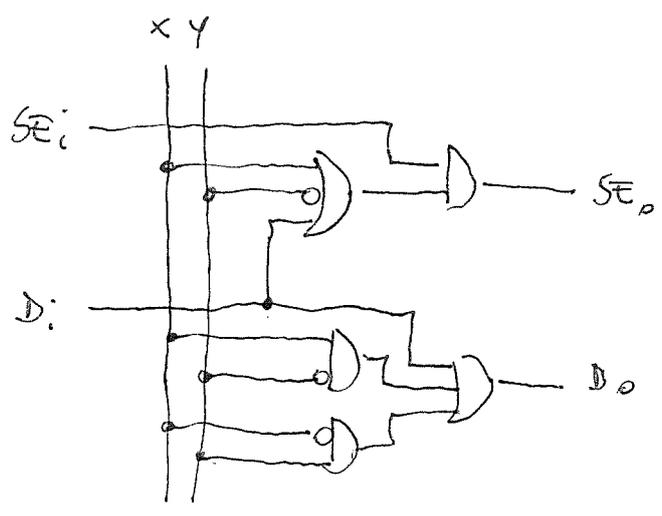


$$SE_p = SE_i D_i + SE_i x + SE_i \bar{y}$$

$$SE_i \cdot (D_i + x + \bar{y})$$

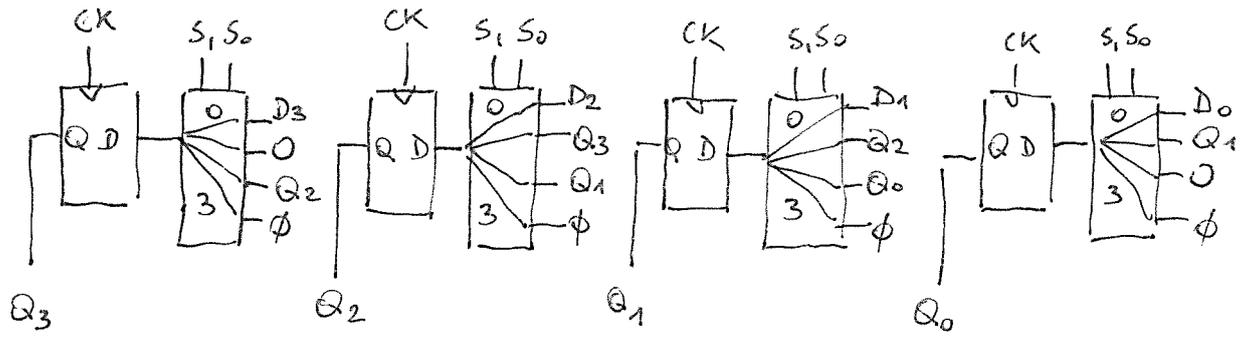


$$D_0 = D_i + \bar{x}y + x\bar{y}$$



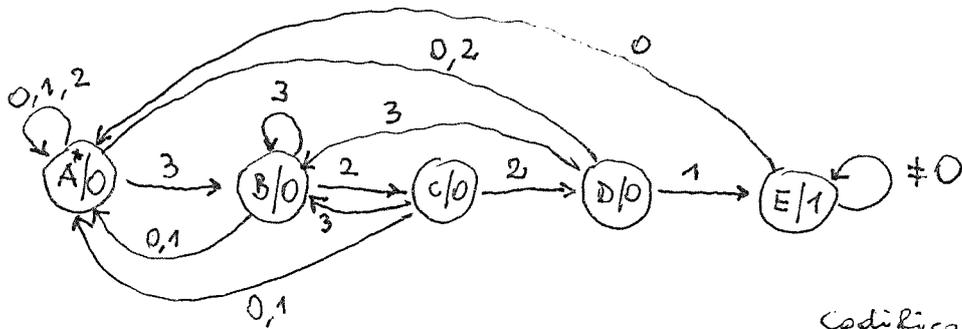
4

Registro UNIVERSALE



5

gruppo



Codifica	U
A	000
B	001
C	010
D	011
E	111

stato reset

$q_1 q_0$	$x_1 x_0$	00	01	11	10
0: 00	A	000	000	000	000
1: 01	B	000	000	111	000
3: 11	D	001	001	001	001
2: 10	C	000	010	000	011

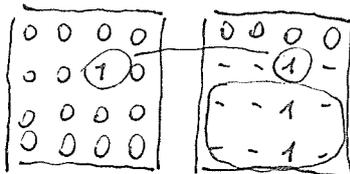
$q_2 = 0$

$q_2 q_1 q_0$	00	01	11	10
000	000	000	000	000
-	-	111	-	-
-	-	111	-	-
-	-	111	-	-

$q_2 = 1$

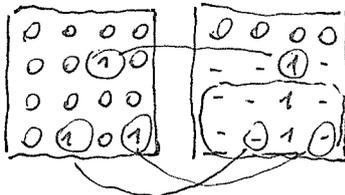
Anche gli stati non previsti vengono RESETTATI dall'ingresso 00 (questo elimina il rischio di blocco in caso di errore nello stato)

simboli SP

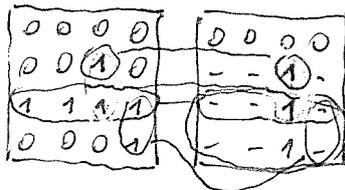


$$q_2^+ = q_1 q_0 \bar{x}_1 x_0 + q_2 x_1$$

(Va bene per U)

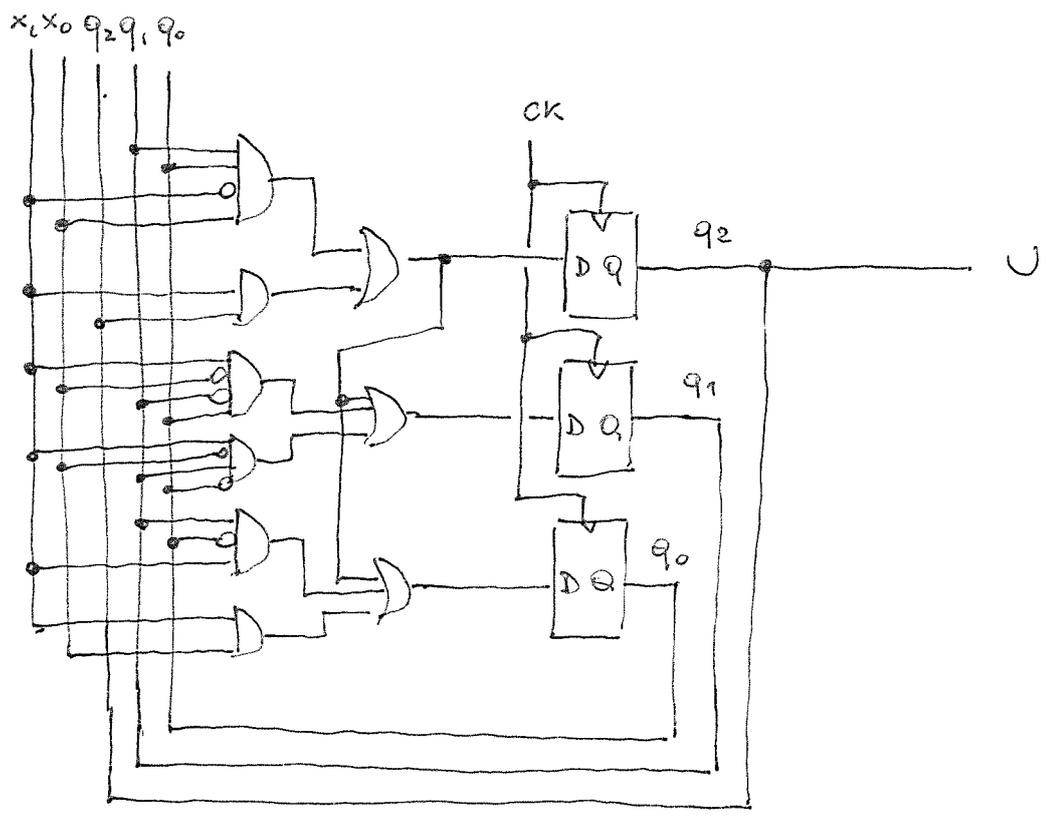


$$q_1^+ = q_1 q_0 \bar{x}_1 x_0 + q_2 x_1 + \bar{q}_1 q_0 x_1 \bar{x}_0 + q_1 \bar{q}_0 x_1 x_0$$



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

(non ottimale, ma nessuno impiccanti già voluti)



6

La domanda a) ha una risposta generale.
Per garantire l'errore richiesto servono (arrotondamenti) per la parte frazionaria:

n bit } con n minimo tale che $2^{-n-1} < 10^{-2}$
quindi $n=6$

[infatti $\frac{1}{128} < \frac{1}{100}$]

I numeri sono (5 cifre signif.)

$x \approx 5,7873$
 $y \approx -1,7725$
 $z \approx -1,5333$

Per la parte intera servono $m=3$ bit. 1 bit per il segno.
In totale servono 10 bit.

Per rispondere alle domande si trova la rappresentazione con 10 bit e poi si eliminano bit verificando che l'errore non rispetta la specifica.

X MS	0101.110010	} conviene pensare da MS
Y MS	1001.110001	
Z MS	1001.100010	

Rappres C2 su 10 bit

$x = 0101.110010$	5,78125	$-6,094 \cdot 10^{-3}$
$y = 1110.001111$	-1,765625	$6,829 \cdot 10^{-3}$
$z = 1110.011110$	-1,53125	$2,083 \cdot 10^{-3}$

Rappz. su 9 bit

$x = 0101.11001$	5,78125	$-6,094 \cdot 10^{-3}$	ok
$y = 1110.01000$	-1,75	$2,245 \cdot 10^{-2}$	non rispetta
$z = 1110.01111$	-1,53125	$2,083 \cdot 10^{-3}$	ok

Rappz. su 8 bit (registro XMEGA)

$x = 0101.1101$	5,8125	$2,516 \cdot 10^{-2}$
$y = 1110.0100$	-1,75	$2,245 \cdot 10^{-2}$
$z = 1110.1000$	-1,5	$3,333 \cdot 10^{-2}$