

Cognome

Nome

ESERCIZIO N°1

8 punti

Realizzare una subroutine per il microcontrollore AVR XMEGA256A3BU, in grado di inizializzare un blocco di memoria da N byte, con indirizzo iniziale A , con un valore ottenuto a partire dall'indirizzo stesso di ciascuna locazione. La subroutine trova N nella coppia di registri R1:R0 e l'indirizzo A nel puntatore Y. Il valore da scrivere in ciascun byte è ottenuto moltiplicando tra loro parte alta e parte bassa dell'indirizzo e sommando quindi la parte bassa del risultato con il complemento della parte alta del risultato. Si ha la garanzia che tutti gli indirizzi del blocco da inizializzare sono validi.

ESERCIZIO N°2

6 punti

Sintetizzare una rete sequenziale sincronizzata secondo il modello di Moore, con in ingresso IN e una uscita U , in grado di riconoscere ponendo U a 1 per un ciclo di clock, le due sequenze, non interallacciate in alcun modo, 00110 e 01110. Minimizzare il numero di flip-flop usati.

ESERCIZIO N°3

3 punti

Disegnare lo schema logico di un sequenziatore con contatore sincrono (dotato della possibilità di caricamento parallelo e reset asincrono) che implementi microcodice specificato nel seguito.

```
A:  IF J THEN C ELSE F; OP = 101
B:  IF K THEN F ELSE G; OP = 100
C:  IF M THEN D ELSE H; OP = 010
D:  IF M THEN E ELSE C; OP = 111
E:  IF L THEN H ELSE D; OP = 110
F:  IF L THEN A ELSE B; OP = 000
G:  IF J THEN B ELSE A; OP = 011
H:  IF K THEN G ELSE G; OP = 001
```

ESERCIZIO N°4

6 punti

- a) Determinare la mappa di Karnaugh di una funzione logica $DV = f(X_4, X_3, X_2, X_1, X_0)$ dove X_3, X_2 e X_1, X_0 rappresentano due cifre consecutive in codifica GRAY, mentre X_4 è un bit di parità (logica parità **pari**)
 DV (Data Valid) vale 1 se se si verificano entrambe queste condizioni: la codifica GRAY è rispettata; la regola di parità è corretta (DV vale 0 altrimenti).
- b) Realizzare con circuito a porte logiche AND, OR, NOT e 2 livelli di logica la funzione del punto a) scegliendo tra diverse possibili soluzioni quella che minimizza numero di porte logiche.
- c) Se porte logiche elementari (AND, OR, NOT) a K ingressi hanno $T_{pd} = 0,05 \text{ ns} + 0,1 K \text{ ns}$, quale è il T_{pd} massimo del circuito di cui al punto b)? Se ingressi e uscite del circuito combinatorio di cui al punto b) sono registrati con registri aventi $T_{co} = 0,15 \text{ ns}$, $T_{hold} = 0,05 \text{ ns}$ e $T_{setup} = 0,1 \text{ ns}$ quale è la massima frequenza di lavoro possibile?

ESERCIZIO N°5

4 punti

Un circuito elettronico per telecomunicazioni che connette 32 unità che trasmettono e 8 che ricevono è composto dalla cascata di un multiplexer e di un demultiplexer:

- a) Realizzare il circuito usando decoder e porte three-state.
- b) Realizzare il circuito a partire da mux 2 to 1 e demux 1 to 2.

ESERCIZIO N°6

6 punti

Dati i numeri $X = -\pi$, $Y = \sqrt{2}$ e $Z = -10/3$ e ipotizzando che sia accettabile un errore minore o uguale a 10^{-1} sulla parte frazionaria

- a) Determinare il numero minimo di bit per raggiungere la specifica e la loro rappresentazione in virgola fissa e MS, C2, C1 e Traslazione.
- b) Determinare la loro rappresentazione in virgola mobile formato standard IEEE 754 singola precisione.
- c) Se si usa il microcontrollore AVR XMEGA analizzato durante il corso, quale errore si commette nel rappresentare i numeri di cui sopra in un registro?

```

/*
 * ASE1605.asm
 *
 * Created: 26/07/2016 7.50.55
 * Author: Roberto Roncella
 * Compito del 04/07/2016
 * Realizzare una subroutine per il microcontrollore AVR XMEGA256A3BU,
 * in grado di inizializzare un blocco di memoria da N byte, con indirizzo iniziale A,
 * con un valore ottenuto a partire dall'indirizzo stesso di ciascuna locazione.
 * La subroutine trova N nella coppia di registri R1:R0 e l'indirizzo A nel puntatore
 * Y.
 * Il valore da scrivere in ciascun byte è ottenuto moltiplicando tra loro parte alta
 * e parte bassa dell'indirizzo e sommando quindi la parte bassa del risultato
 * con il complemento della parte alta del risultato.
 * Si ha la garanzia che tutti gli indirizzi del blocco da inizializzare sono validi.
 */
.EQU N=500
.EQU A1=0x2100

test: //sezione di test della subroutine
    ldi R16,low(N)
    mov R0,R16
    ldi R16,high(N)
    mov R1,R16
    ldi YL,low(A1)
    ldi YH,high(A1)
l11:
    rcall Initialize
    rjmp l11

Initialize:
    push R0 //salva i registri usati
    push R1
    push R16
    push XL
    push XH
    push YL
    push YH

    clr R16
    cp R0,R16
    cpc R1,R16
    breq end //non c'è da fare nulla

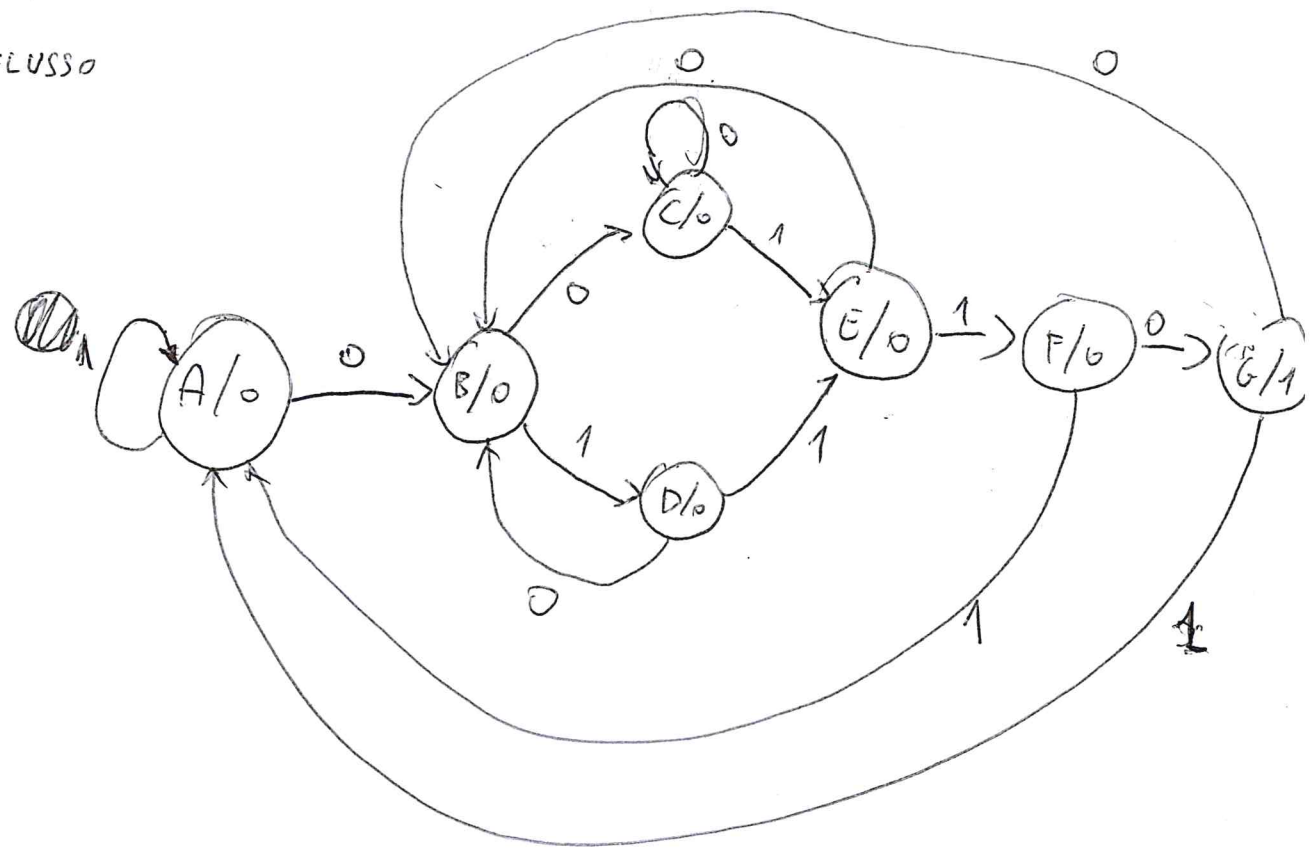
    movw XH:XL,R1:R0
loop:
    ld R16,Y
    mul YH,YL //moltiplica parte alta e bassa dell'indirizzo
    com R1 //complementa parte alta del risultato
    add R0,R1 //somma parte bassa e parte alta complementata
    st Y+,R0 //si salva il valore trovato (ignorando il riporto; non si potrebbe fare
    altro)
    sbiw XH:XL,1
    brne loop

```

```
end:  
pop YH  
pop YL  
pop XH  
pop XL  
pop R16  
pop R1  
pop R0 //ripristina i registri usati  
ret
```

ESERCIZIO N°2

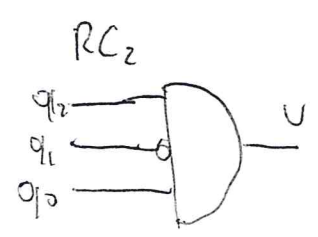
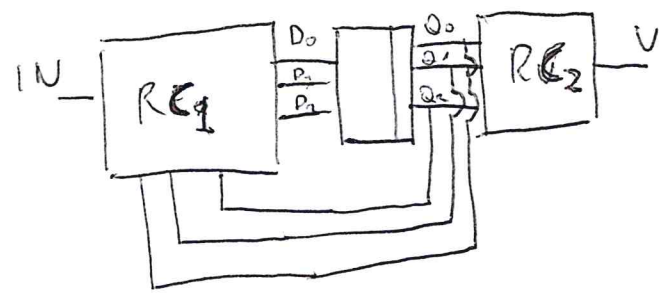
GRAFO DI FLUSSO



OCCORRENZA 3 FF

Codifica degli stati

	Q_2	Q_1	Q_0
A	0	0	0
B	0	0	1
C	0	1	1
D	0	1	0
E	1	1	0
F	1	1	1
G	1	0	1
(H)	1	0	0



Rete per lo stato futuro D_2, D_1, D_0

$Q_1, Q_0 \backslash IN, Q_2$	00	01	11	10
00	001	-	-	000
01	011	000	000	010
11	011	101	000	110
10	001	001	111	110

D_2

$Q_1, Q_0 \backslash IN, Q_2$	00	01	11	10
00	0	-	-	0
01	0	0	0	0
11	0	1	0	1
10	0	0	1	1

D_1

$Q_1, Q_0 \backslash IN, Q_2$	00	01	11	10
00	0	-	-	0
01	1	0	0	1
11	1	0	0	1
10	0	0	1	1

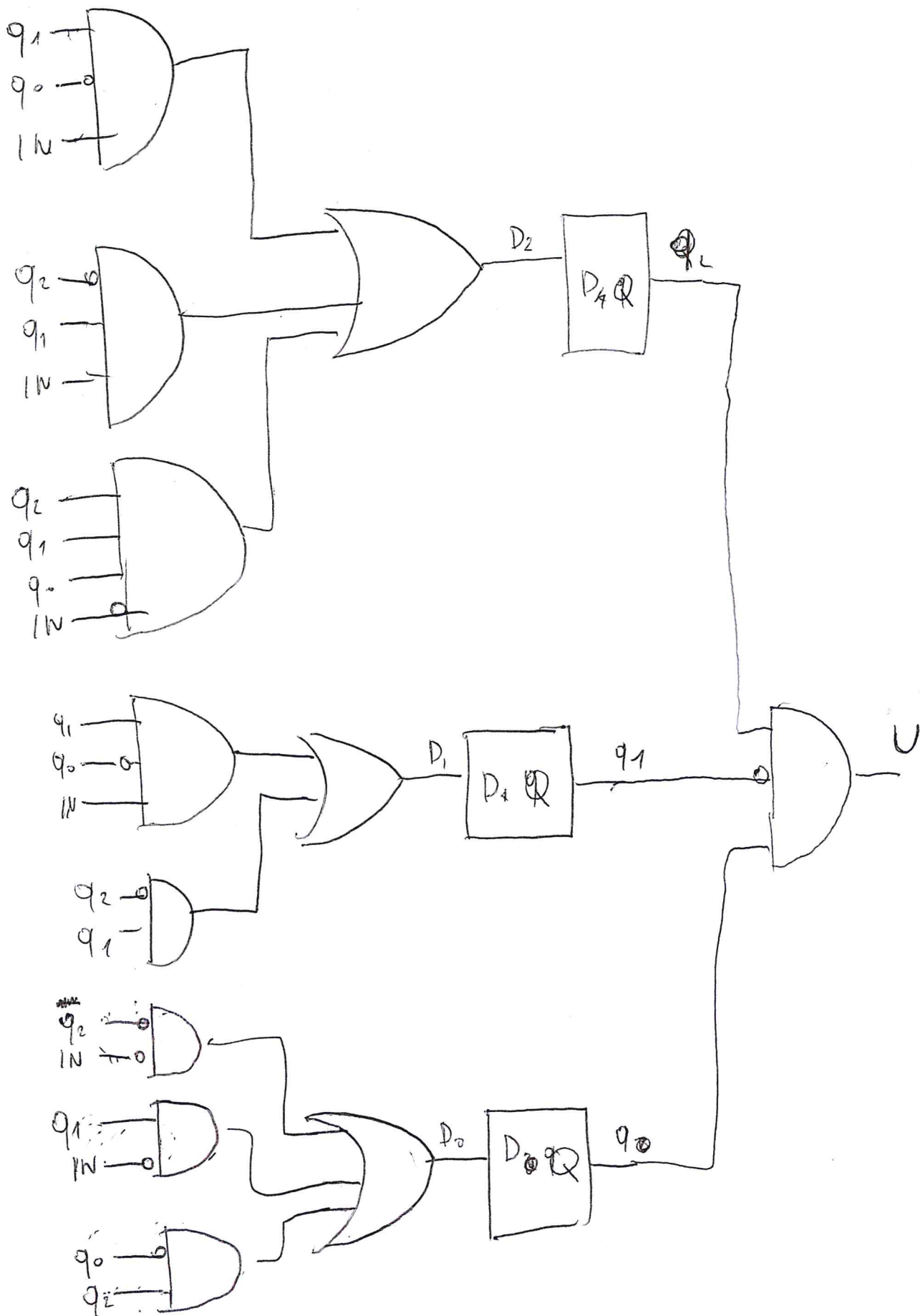
D_0

$Q_1, Q_0 \backslash IN, Q_2$	00	01	11	10
00	1	0	-	0
01	1	0	0	0
11	1	1	0	0
10	1	1	1	0

$$D_2 = Q_1 \bar{Q}_0 IN + Q_1 \bar{Q}_2 IN + Q_0 Q_1 Q_2 \bar{IN}$$

$$D_1 = Q_1 \bar{Q}_0 IN + Q_0 \bar{Q}_2$$

$$D_0 = \bar{Q}_2 \bar{IN} + Q_1 \bar{IN} + \bar{Q}_0 Q_2$$



ESERCIZIO N. 3

ESISTE UNA SEQUENZA CICLICA completa (ver)

A, E, D, E, H, G, B, F, A...

CODIFICO gli stati della sequenza con valori consecutivi.

- A 000
- C 001
- D 010
- E 011
- H 100
- G 101
- B 110
- F 111

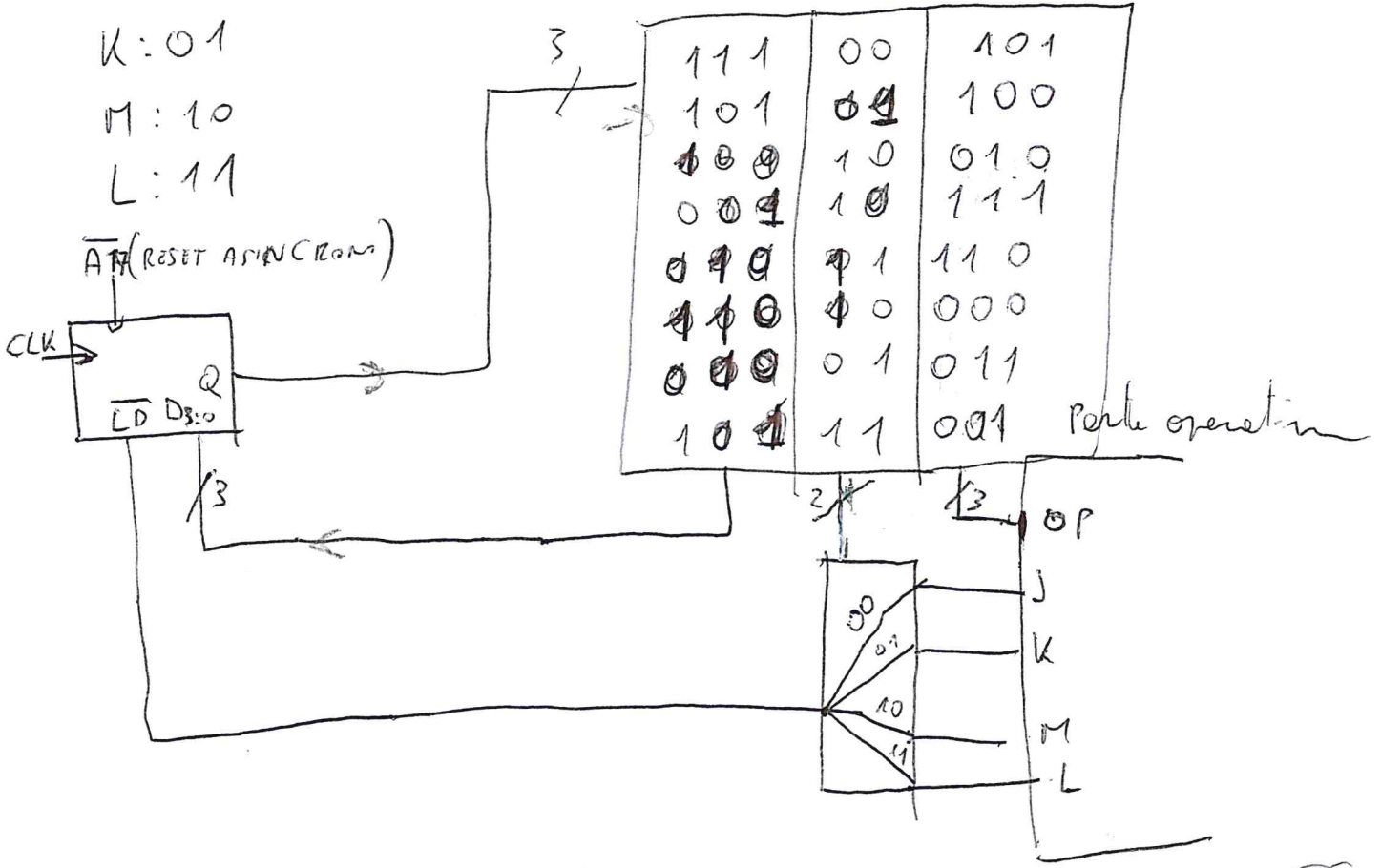
~~XX~~ CODIFICO

J = 00

K = 01

M = 10

L = 11



ESERCIZIO 4

a) MAPPA DV

$X_1 \backslash X_3, X_2$	00	01	11	10
00	0	0	0	0
01	0	0	0	0
11	0	0	0	0
10	0	0	0	0

$X_1 \backslash X_3, X_2$	00	01	11	10
00	0	1	0	1
01	1	0	1	0
11	0	1	0	1
10	1	0	1	0

$X_4 = 0$

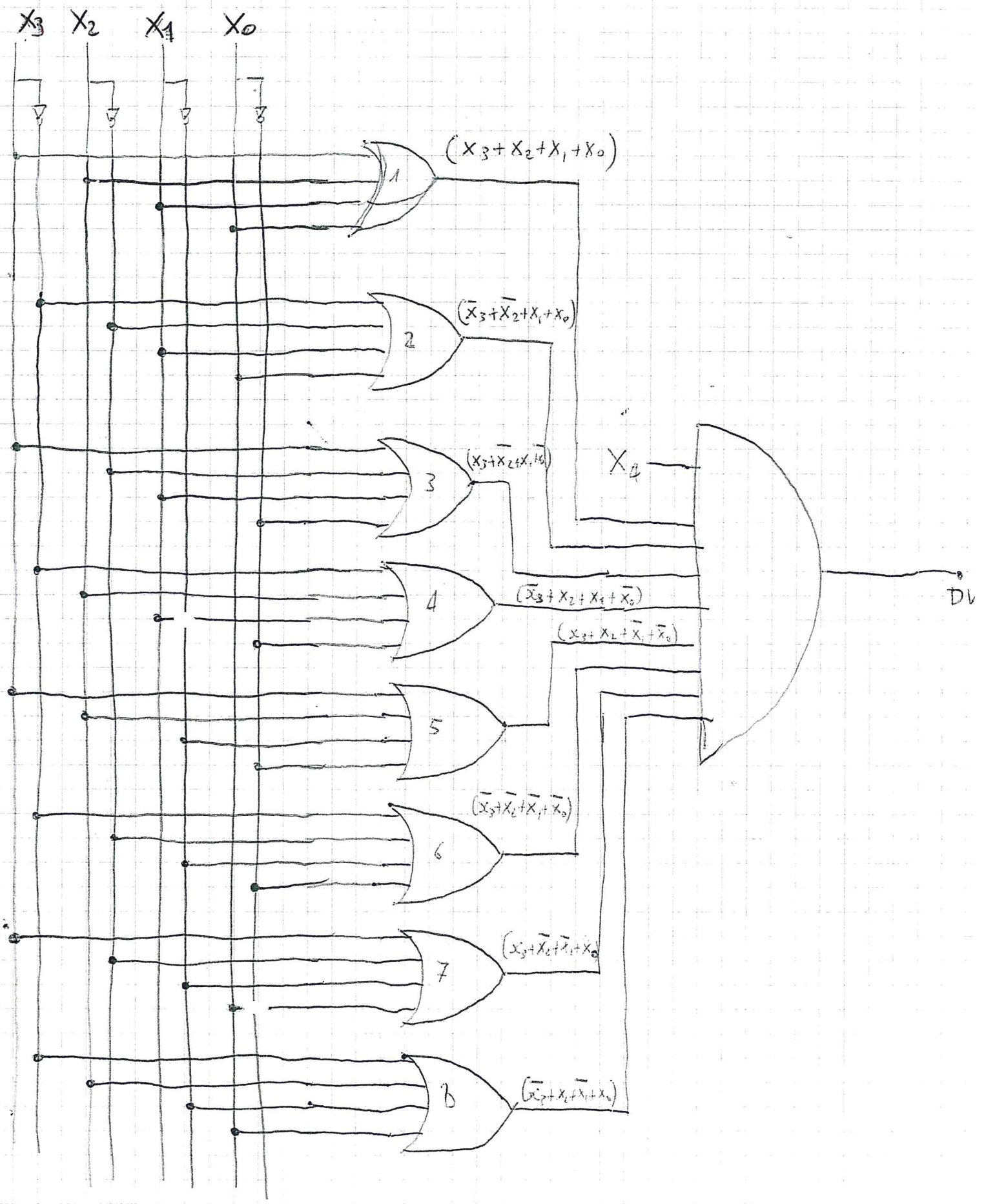
b) SINTESI

SP Servono 8 AND e 5 ingressi
1 OR e 8 ingressi

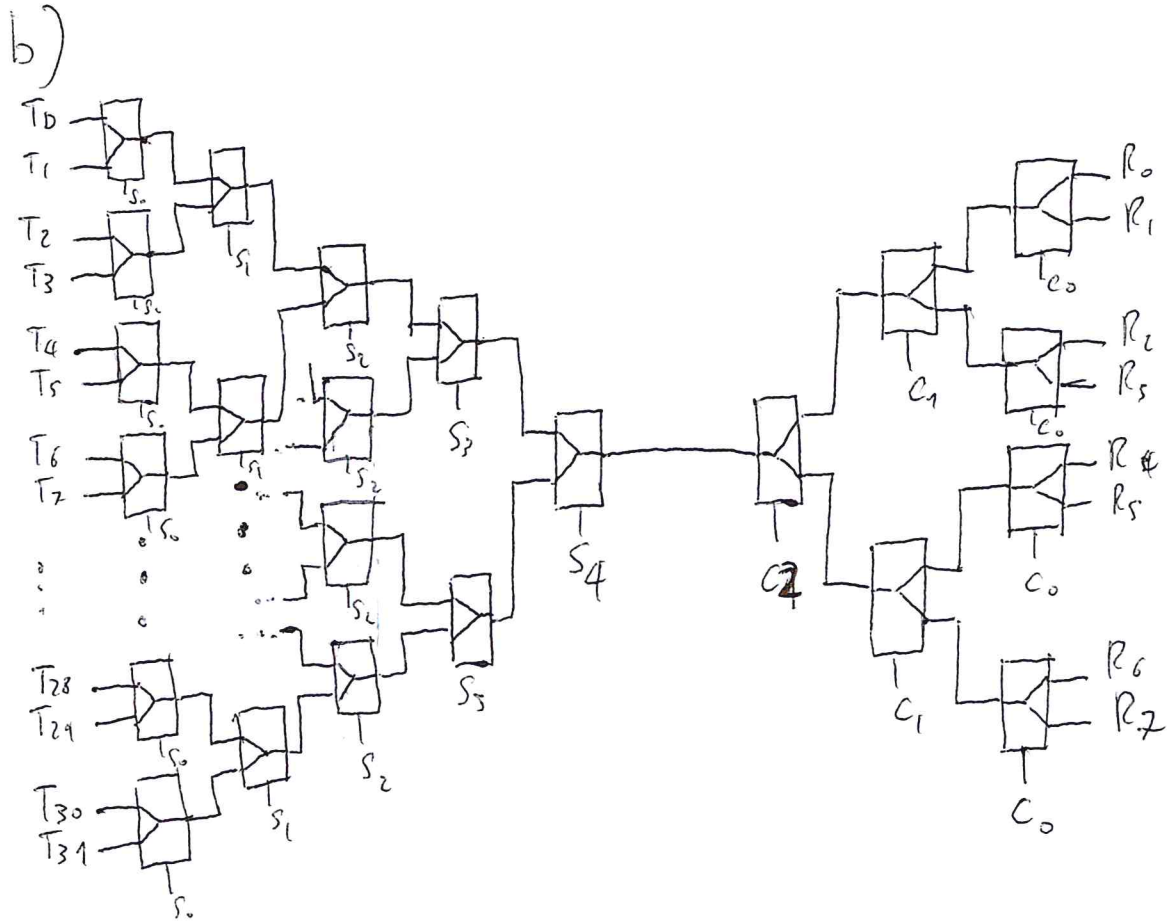
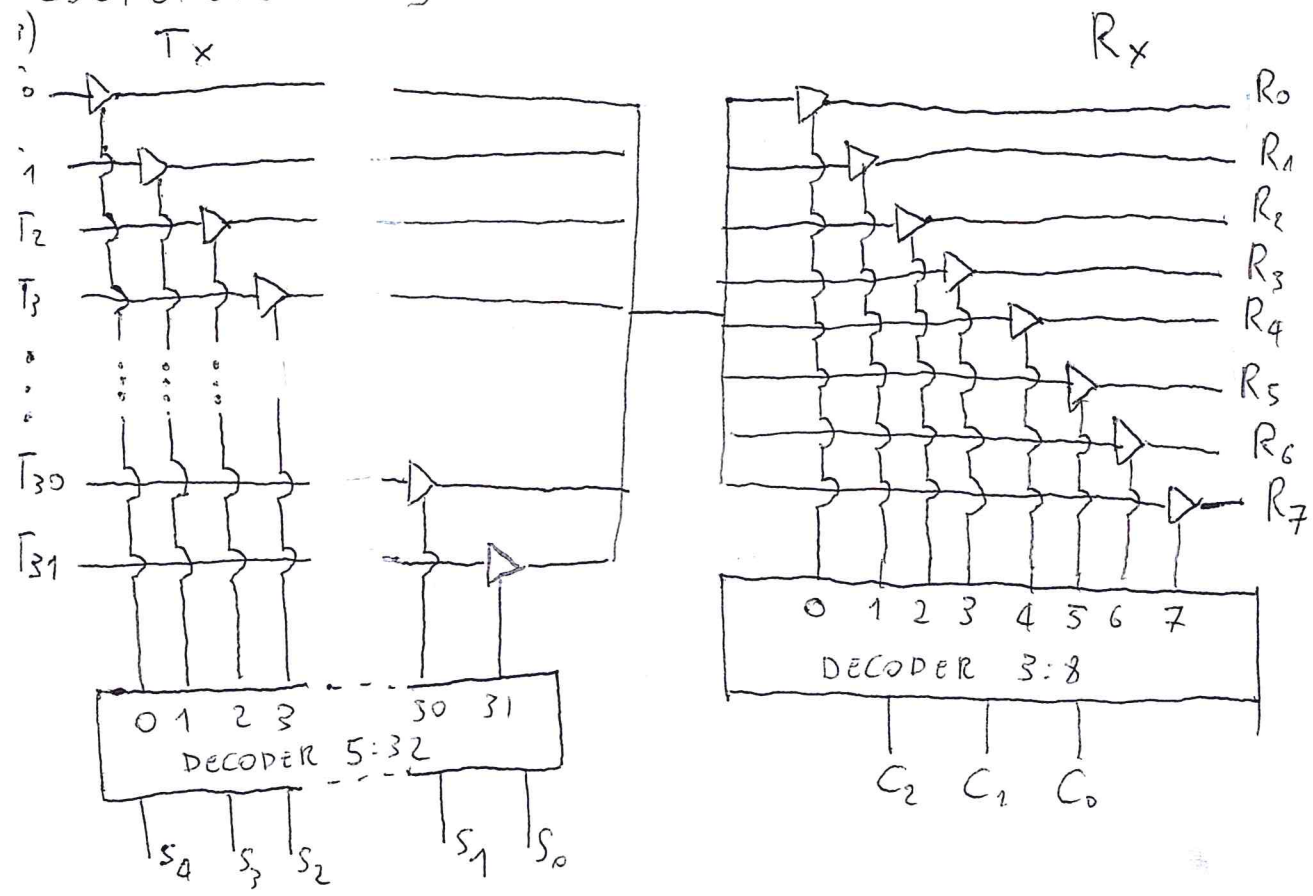
PS Servono 8 OR e 4 ingressi
1 AND e 9 ingressi

Le due soluzioni sono equivalenti dal punto di vista del numero di porte (9)

Scelgo di implementare PS che ha meno letterali.



ESERCIZIO N. 5



ESERCIZIO NUM. 6

a)
 IN GENERALE, utilizzare 3 bit per la parte frazionaria garantisce un errore massimo di 0,125 che è maggiore di quello richiesto nelle specifiche (0,1).

Si può però notare che per i tre numeri richiesti dall'esercizio usare 3 bit per la parte frazionaria garantisce un errore minore di 10^{-1} .
 Infatti utilizzando 3 bit per la parte frazionaria i numeri vengono approssimati come nel seguito:

$$\begin{aligned}
 -\pi &= -3,141\dots \rightarrow -3,125 & |\epsilon| &< 0,017 \\
 \sqrt{2} &= 1,414\dots \rightarrow 1,375 & |\epsilon| &< 0,04 \\
 -10/3 &= -3,33\dots \rightarrow -3,375 & |\epsilon| &< 0,042
 \end{aligned}$$

La parte intera con segno richiede 3 bit
 In totale occorrono perciò 6 bit

$$\begin{array}{l}
 X = -\pi \rightarrow -3,125 \\
 \text{MS } 111,001 \\
 \text{C1 } 100,110 \\
 \text{C2 } 100,111 \\
 \text{T } 000,111
 \end{array}$$

$$\begin{array}{l}
 Y = \sqrt{2} \rightarrow 1,375 \\
 \text{MS } 001,001 \\
 \text{C1 } 001,001 \\
 \text{C2 } 001,001 \\
 \text{T } 101,001
 \end{array}$$

$$\begin{array}{l}
 Z = -10/3 \rightarrow -3,375 \\
 \text{MS } 111,011 \\
 \text{C1 } 100,100 \\
 \text{C2 } 100,101 \\
 \text{T } 000,101
 \end{array}$$

b) La rappresentazione in formato IEEE 754, ha un numero di cifre significative pari a 23 e quindi consente di rappresentare senza errori i numeri precedentemente approssimati (e quindi mantenendo l'errore precedentemente calcolato)

$$x = (-1)^s \cdot 2^{e-127} \left\{ 1 + \sum_{i=1}^{23} b_{23-i} \cdot 2^{-i} \right\}$$

$$-3,125 = (-1)^1 \cdot 2^1 \left(1 + \cancel{7185} \cdot 2^{-23} \right)$$

$$\left[\begin{array}{l} s \quad e=123 \\ 1 \quad 10000000 \quad 10000000000000000000000000000000 \end{array} \right]$$

$$1,375 = (-1)^0 \cdot 2^0 \left(1 + 3145728 \cdot 2^{-23} \right)$$

$$\left[\begin{array}{l} s \\ 0 \quad 01111111 \quad 01100000000000000000000000000000 \end{array} \right]$$

$$-5,375 = (-1)^1 \cdot 2^2 \left(1 + 5767168 \right)$$

$$\left[\begin{array}{l} s \quad e=122 \\ 1 \quad 10000000 \quad 10110000000000000000000000000000 \end{array} \right]$$

3) Avendo a disposizione un registro a 8 bit possiamo rappresentare i numeri con lo stesso ~~precisione~~ ~~precisione~~ calcolati al punto e), utilizzando a due bit aggiuntivi per rappresentare la parte intera.

Oppure possiamo usare i 2 bit nella parte frazionaria per aumentare la precisione

$$-\pi \rightarrow -3,1525$$

$$111.01101 \quad |\varepsilon| < 0,016$$

$$\sqrt{2} \rightarrow 1,40625$$

$$001,01101 \quad |\varepsilon| < 0,008$$

$$-10/3 \rightarrow -3,33375$$

$$111.01011 \quad |\varepsilon| < 0,011$$