

Non è ammessa la consultazione degli appunti e dei compiti precedenti. Si possono consultare i data sheet, anche su PC. Per lo svolgimento dei calcoli è possibile usare, oltre alla solita calcolatrice, anche il PC con applicativi numerici (es.: Matlab, Excel, ...).

ESERCIZIO N°1

5 punti

Progettare una rete combinatoria, facente uso di blocchi noti (porte logiche elementari, multiplexer, full-adder, ecc.) in grado di eseguire la somma di 4 numeri relativi a 8 bit, rappresentati rispettivamente come MS, Traslazione, Traslazione e C2 e il cui risultato (per il risultato scegliere il numero di bit minimo che garantisce in ogni caso la rappresentabilità) sia rappresentato in C1.

ESERCIZIO N°2

8 punti

Realizzare un sottoprogramma per il microcontrollore AVR XMEGA256A3BU che sostituisce i valori (senza segno) contenuti nelle locazioni di memoria comprese tra gli indirizzi 0x2200 e 0x24FF, compresi gli estremi, con la media tra 16 valori: il valore stesso e i 15 nelle celle successive (il risultato deve approssimare nel modo migliore il valore esatto, usando se necessario l'arrotondamento al numero pari più vicino).

ESERCIZIO N°3

5 punti

Disegnare il diagramma di flusso (a scelta della studente) di un sequenziatore a 8 stati. La parte operativa è gestita con codici operativi a 4 bit e ha in uscita 3 flag. Dopo aver codificato gli stati e i flag, realizzare lo schema logico del sequenziatore proposto, mostrando il contenuto della sua memoria.

ESERCIZIO N°4

5 punti

Lo studente converta in BCD M e usi la successione dei bit da d_0 a d_{23} e poi a ritroso da d_{23} a d_{16} come tabella di verità di una funzione combinatoria delle 5 variabili di ingresso x_4, x_3, x_2, x_1, x_0 . Sintetizzare la funzione in forma SP ottima (minimo numero di letterali), indicando in modo esplicito e motivato quali sono gli implicati essenziali (non è richiesto il disegno dello schema logico, ma solo l'espressione della forma ottima).

ESERCIZIO N°5

5 punti

Lo studente realizzi la funzione dell'esercizio precedente esclusivamente con multiplexer 4:1, evitando l'uso di elementi palesemente ridondanti.

ESERCIZIO N°6

5 punti

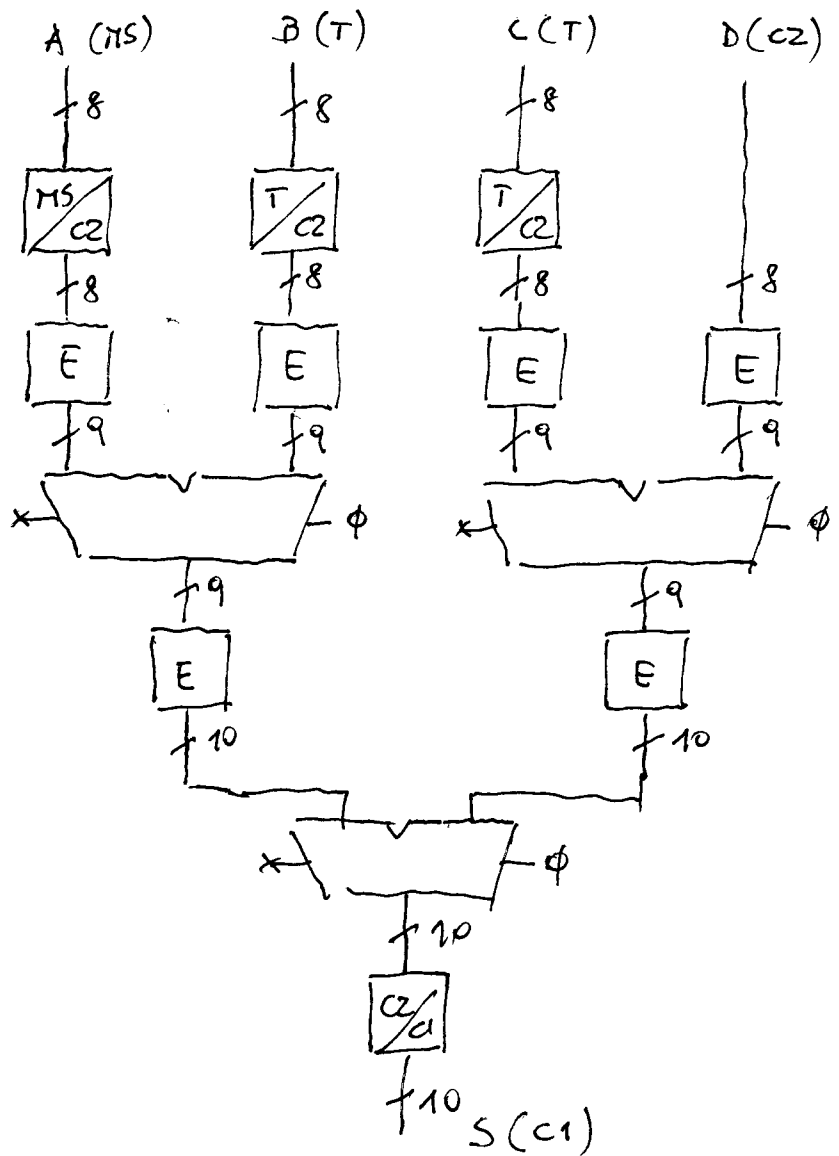
Determinare il grafo secondo Moore di un riconoscitore per le seguenti sequenze comunque interallacciate costituite dai 4 bit delle diverse cifre decimali della propria matricola (a partire dal bit più significativo, per esempio 5 corrisponde alla sequenza 0101). Determinare il numero minimo di D-FF necessari per la sintesi della rete e disegnare l'architettura (la sintesi completa delle reti non è richiesta).

① Determiniamo come prima cosa il range (e il n° di bit) del risultato. 4 valori siano A, B, C, D

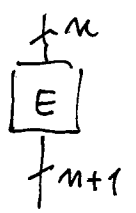
$$\left. \begin{aligned} -127 \leq A \leq 127 \\ -128 \leq B, C, D \leq 127 \end{aligned} \right\} \text{somma } S$$

$$-511 \leq S \leq 508$$

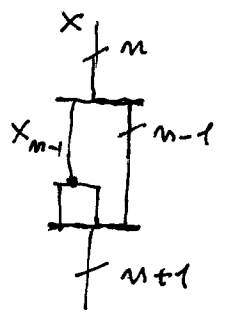
Il risultato è quindi SEMPRE rappresentabile in C1 su 10 bit. Per la rete, convertiamo tutto in C2, eseguiamo le somme e riconvertiamo in C1.



Vediamo lo schema dei singoli blocchi

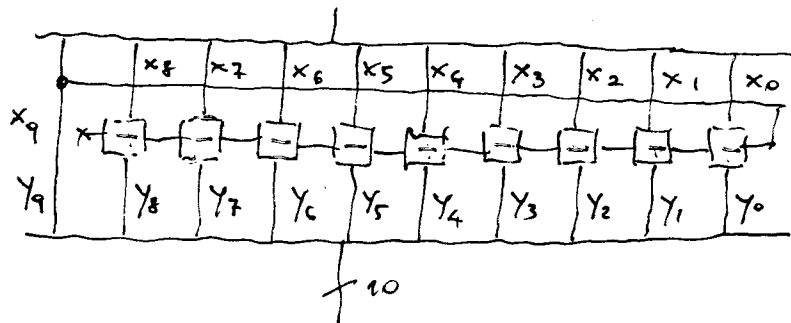
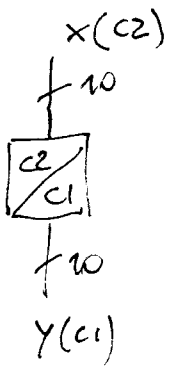
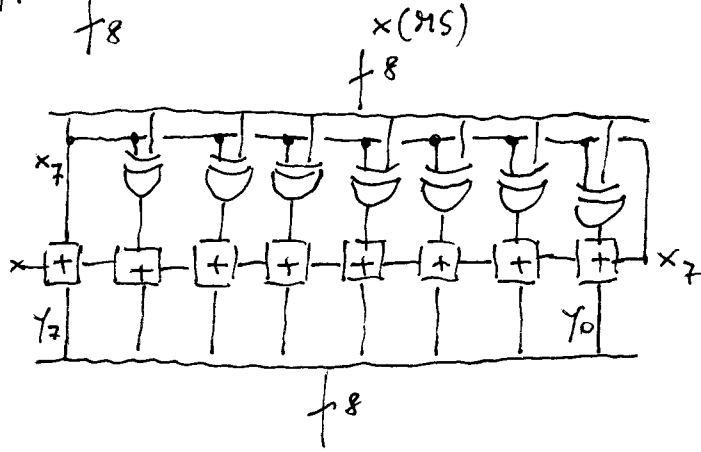
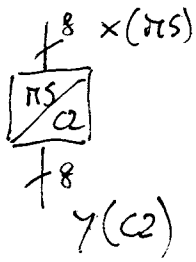
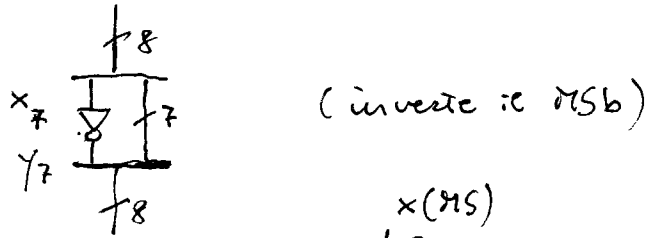
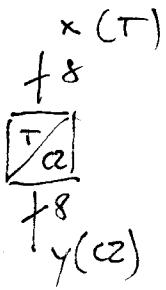


estensione in C2

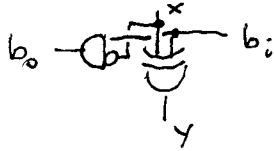


(replica il MSB)

x_{esteso}



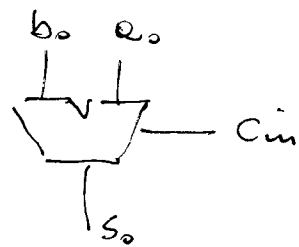
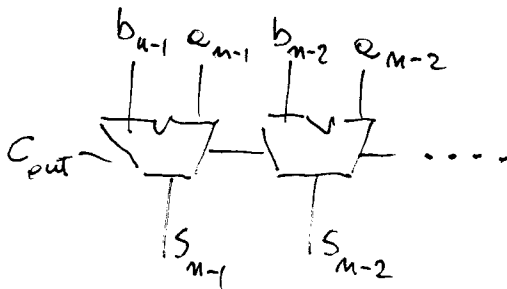
half subtractor



$y(c_1)$

nel nostro caso non ci può essere 0V

Per i commutatori si possono usare normali ripple-carry



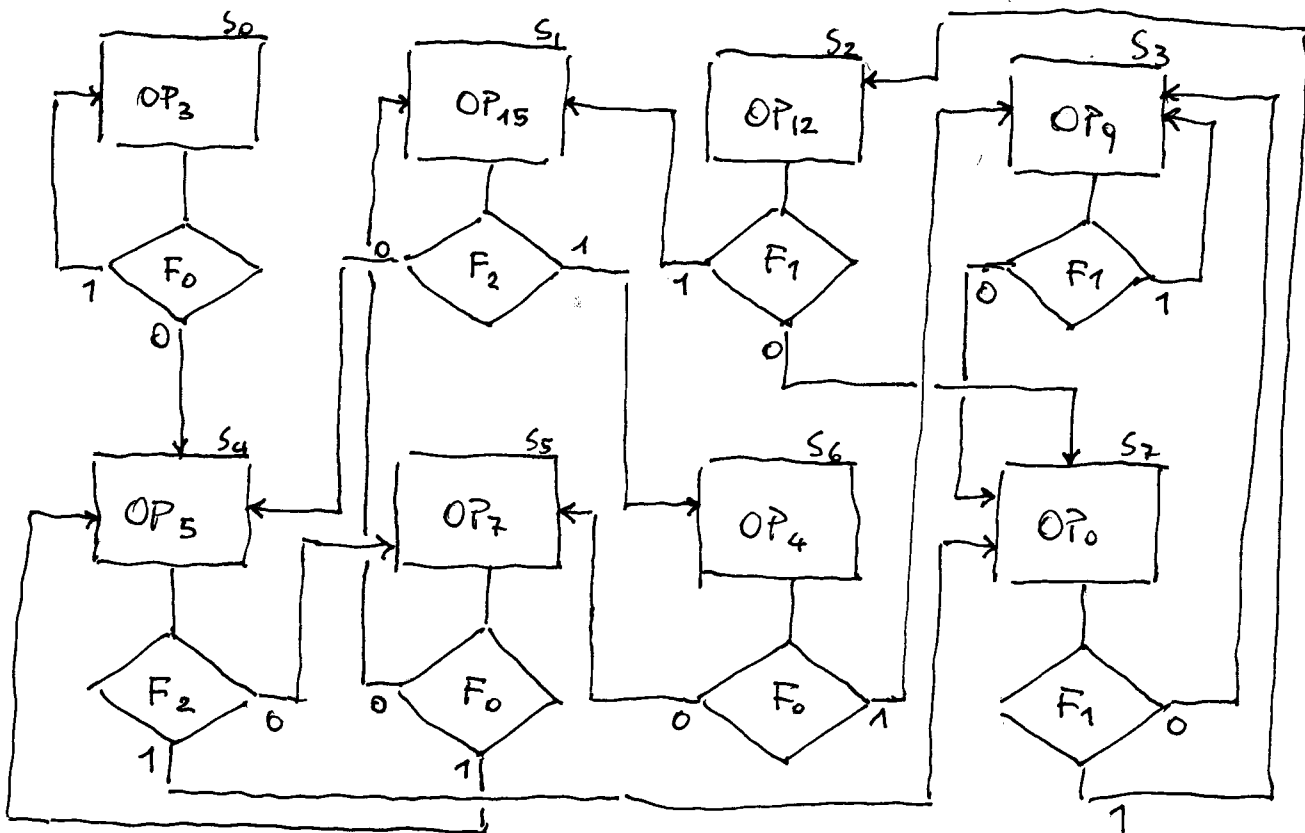
2

Realizzare un sottoprogramma per il microcontrollore AVR XMEGA256A3BU che sostituisce i valori (senza segno) contenuti nelle locazioni di memoria comprese tra gli indirizzi 0x2200 e 0x24FF, compresi gli estremi, con la media tra 16 valori: il valore stesso e i 15 nelle celle successive (il risultato deve approssimare nel modo migliore il valore esatto, usando se necessario l'arrotondamento al numero pari più vicino).

```
/* Realizziamo prima un sottoprogramma media16 che esegue la media tra 16 valori
consecutivi puntati da X lasciando in R16 il risultato arrotondato al pari più vicino */
media16:
    push R20
    push R21
    push R24
    push R25
    clr R21
    clr R24
    clr R25
    ldi R16,16
l0: //somma 16 valori consecutivi
    ld R20,X+
    add R24,R20
    adc R25,R21
    dec R16
    brne l0
    ldi R16,4
l1: //moltiplica per 16 il risultato (in R25 c'è il valore diviso per 16)
    lsl R24
    rol R25
    dec R16
    brne l1
    cpi R24,8 //se maggiore incr, se minore lascia come è, se uguale incr e testa pari
    breq pari
    brcc end
    inc R25
    rjmp end
pari: inc R25
    andi R25,0b11111110 //se l'incremento lo avesse reso dispari...
end: mov R16,R25
    sbiw XH:XL,16 //ripristina X
    pop R25
    pop R24
    pop R21
    pop R20
    ret

/* Ora il sottoprogramma richiesto */
media_mob:
    push R16
    push XL
    push XH
    ldi XL,low(0x2200)
    ldi XH,high(0x2200)
loop:
    rcall media16
    st X+,R16
    cpi XL, low(0x24FF+1)
    brne loop
    cpi XH, high(0x24FF+1)
    brne loop
    pop XH
    pop XL
    pop R16
    ret
```

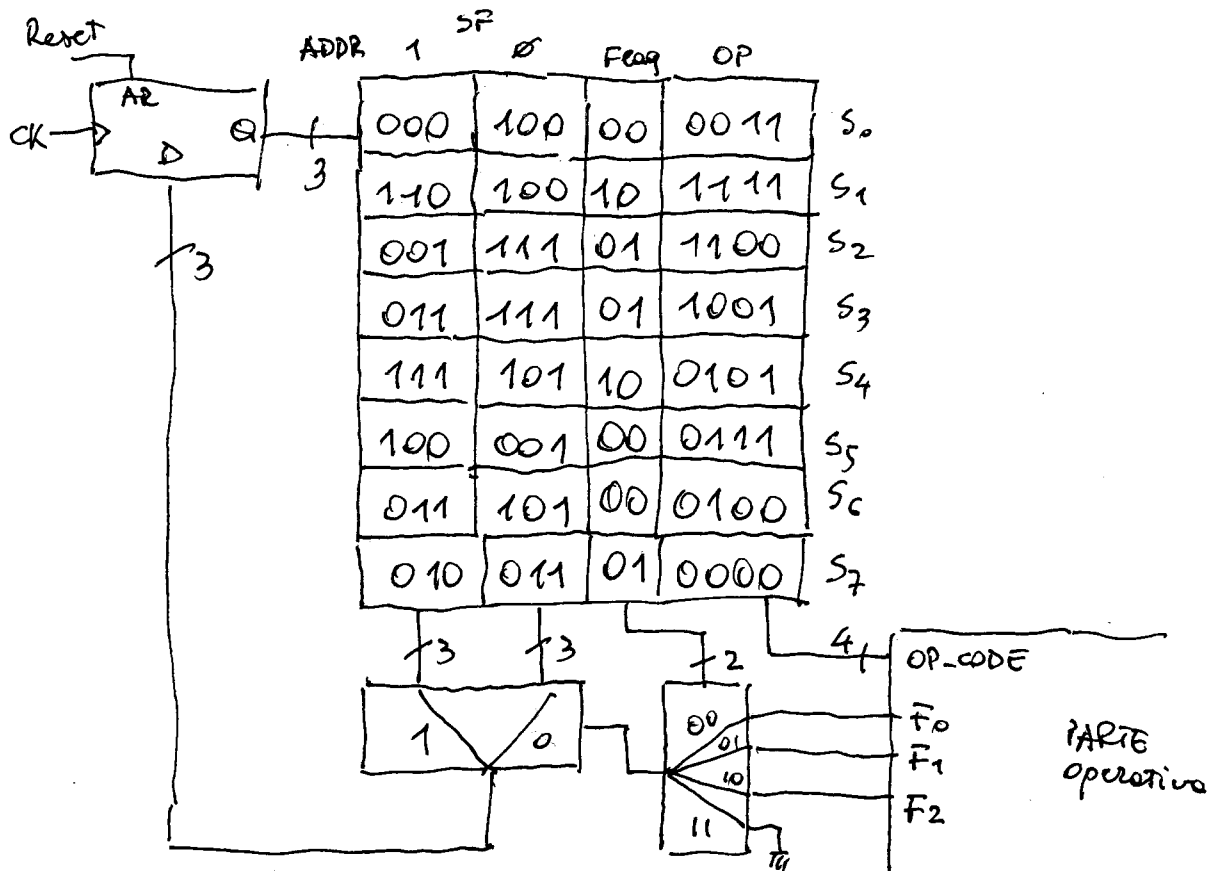
3



Codifico gli stati con il valore binario corrispondente al pedice e ugualmente per i flag.

I codici operativi sono espressi dal valore binario del pedice

Schemo



PARTI operativa

④

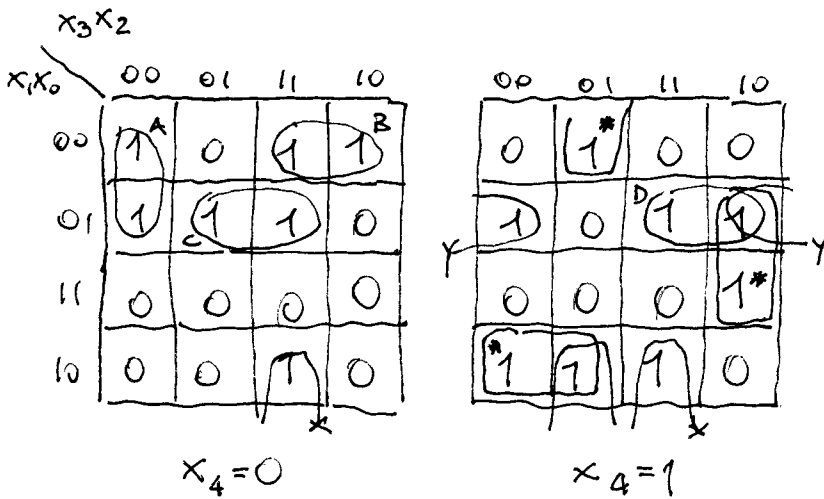
Prendiamo come esempio
 La tabella di verità è:

$M = 567123$

BCD: 010101100111000100100011

do... $d_{23} d_{23} \dots d_{16}$
 1100.0100.1000.1110.0110.1010.0101.0110

ritto in mappa



Con * sono indicati
 i MINTERMINI che
 rendono ESSENZIALE
 l'implicante
 principale a cui
 appartengono

$$\begin{aligned}
 U = & x_4 \bar{x}_3 \bar{x}_2 \bar{x}_0 + x_4 x_3 \bar{x}_2 \bar{x}_0 + x_4 \bar{x}_3 x_1 \bar{x}_0 + \\
 & + \bar{x}_4 \bar{x}_3 \bar{x}_2 \bar{x}_1 + \bar{x}_4 x_3 \bar{x}_1 \bar{x}_0 + \bar{x}_4 x_2 \bar{x}_1 \bar{x}_0 + x_3 x_2 x_1 \bar{x}_0 + \\
 & + x_4 \bar{x}_2 \bar{x}_1 \bar{x}_0 + x_4 x_3 \bar{x}_1 \bar{x}_0
 \end{aligned}$$

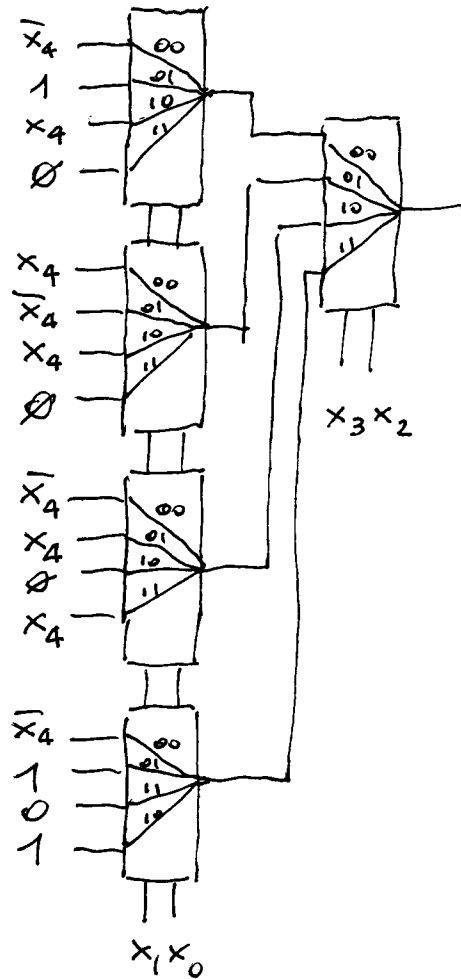
5) Ripeto la mappa

x_1, x_0 \ $x_3 x_2$	00	01	11	10
00	1	0	1	1
01	1	1	1	0
11	0	0	0	0
10	0	0	1	0

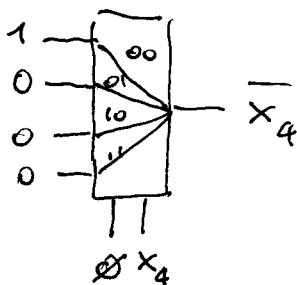
$x_4 = 0$

x_1, x_0 \ $x_3 x_2$	00	01	11	10
00	0	1	0	0
01	1	0	1	1
11	0	0	0	1
10	1	1	1	0

$x_4 = 1$



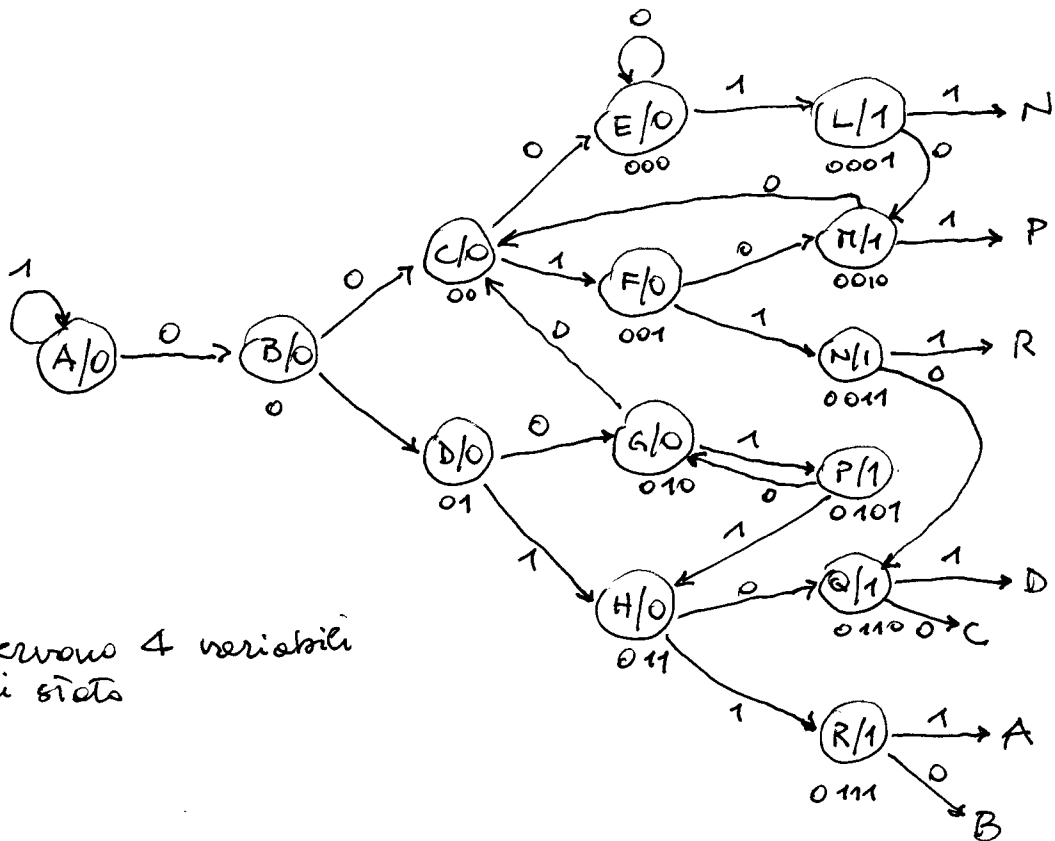
Per ottenere \bar{x}_4 si può usare un ulteriore MUX



6) Usiamo ancora per $M = 567123$

la rete deve riconoscere, interdicciati

0001
 0010
 0011
 0101
 0110
 0111



Servono 4 variabili
 di stato

