

12/2/2019

ES 1

x_3x_4					
x_1x_2		00	01	11	10
	00	1	0	1	1
	01	1	0	1	1
	11	1	0	1	1
	10	1	-	0	1

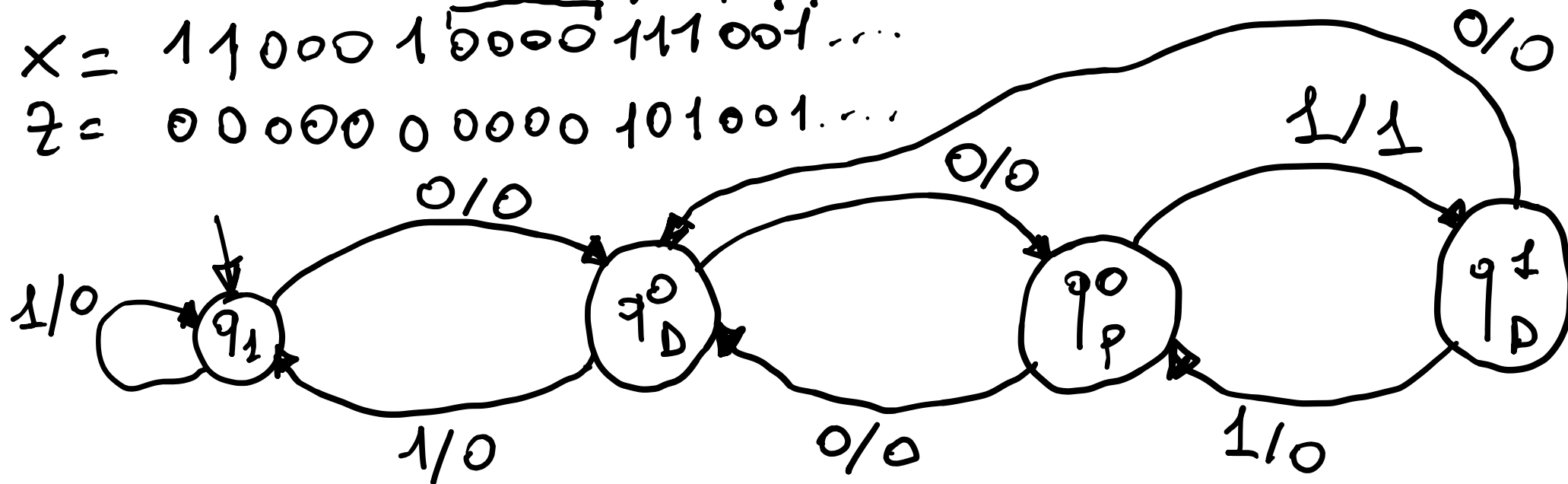
$$f = \overline{x_4} + \overline{x_1}x_3 + x_2x_3$$

Es. 2

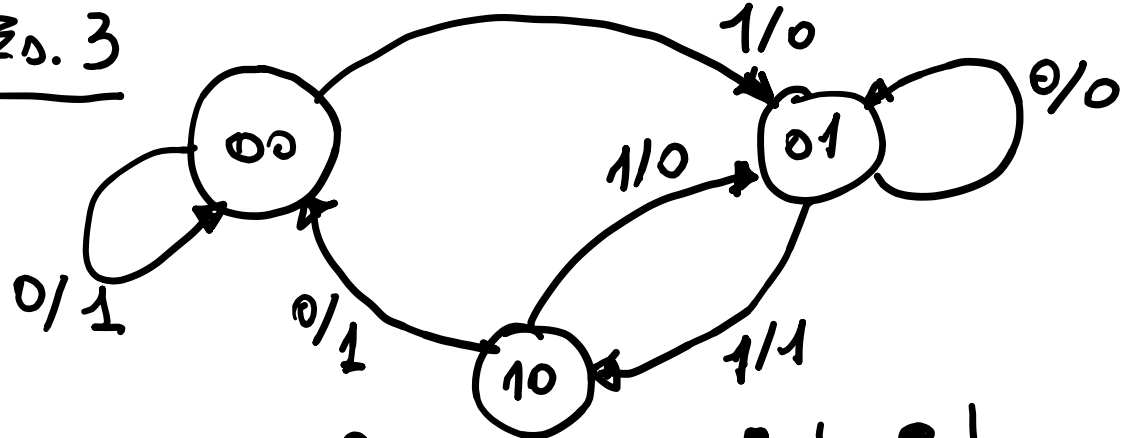
macchine a stati, ingresso x , uscita z ,
 in uscita fornisce 1 ogni volta che su x si è presentata
 una sequenza costituita da un num. pari (>0) di
 0 consecutivi, seguiti da un num. dispari (>0) di
 1 consecutivi. (es. 0, altrimenti)

$x = 1100010000111001\dots$

$z = 0000000000101001\dots$



Es. 3



x	y_1	y_2	z	y_1	y_2	t_1	t_2
0	0	0	1	0	0	0	0
1	0	0	0	0	1	0	1
0	0	1	0	0	1	0	0
1	0	1	1	1	0	1	1
0	1	0	1	0	0	1	0
1	1	0	0	0	1	1	1

flip-flop di tipo T
 t_1, t_2, z : ?

disegno rete: ?

y	\bar{y}	t
0	0	0
0	1	1
1	0	1
1	1	0

x	y_1	y_2	z	y_1	y_2	t_1	t_2
0	0	0	1	0	0	0	0
1	0	0	0	0	1	0	1
0	0	1	0	0	1	0	0
1	0	1	1	1	0	1	1
0	1	0	1	0	0	1	0
1	1	0	0	0	1	1	1

$$t_2 = x$$

$x \backslash y_1 y_2$	00	01	11	10
0	1	0	0	1
1	0	1	1	0

(2)

$$z = \bar{x} \bar{y}_2 + x y_2$$

$x \backslash y_1 y_2$	00	01	11	10
0	0	0	1	1
1	0	1	1	1

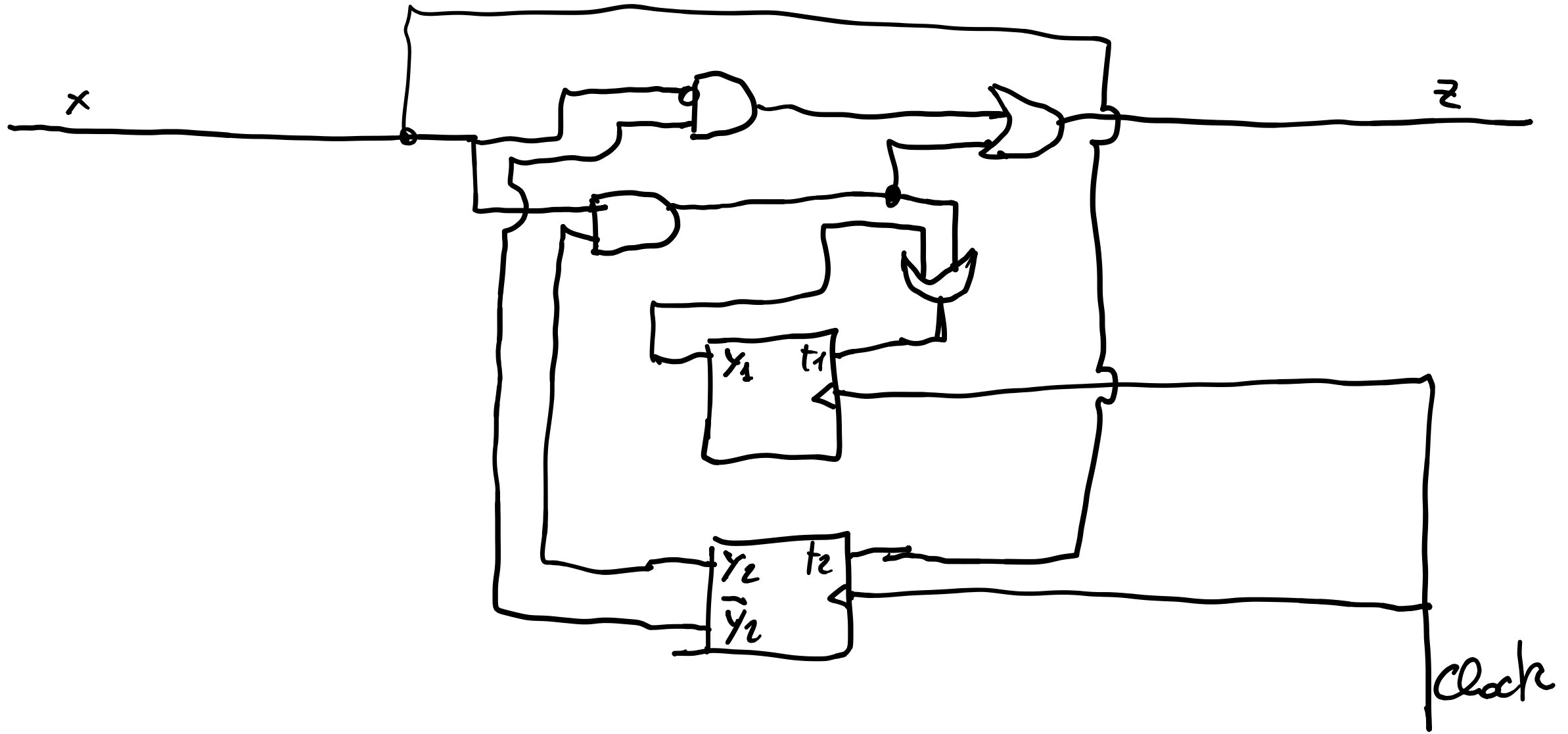
(1)

$$t_1 = y_1 + x y_2$$

$$t_2 = x$$

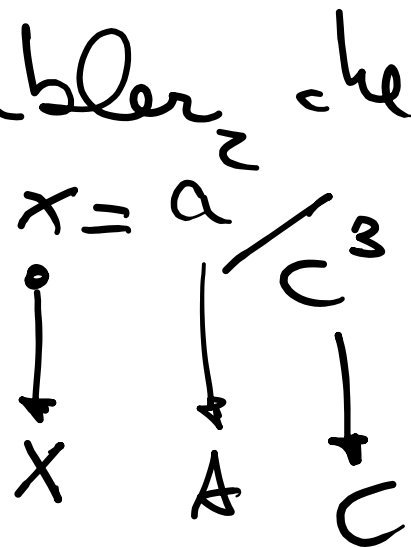
$$z = \bar{x} \bar{y}_2 + x y_2$$

$$t_1 = y_1 + xy_2$$



Es 4:

Determinare la seq. di istruzioni assembler che
realizzino lo statement di alto livello $x = a / c^3$
nel modello registro-registro.



LD R1, A

LD R2, C

MUL R1, R1, R1

MUL R3, R2, R2

MUL R3, R3, R2

DIV R3, R1, R3

ST X, R3

Ex 5:

$SEL1, SEL2, SEL3, SEL4, SEL5, SEL6 \text{ e } SEL7 : ?$

1 MB

16 bit
invariante

17 bit
invariante

19 bit
invariante

	A_{20}	A_{19}	A_{18}	A_{17}
SEL7	1	1	1	1
SEL6	1	1	1	0
SEL5	1	1	0	1
SEL4	1	1	0	0
---SEL3---	1	0	1	1
---	1	0	1	0
---SEL2---	1	0	0	1
---	1	0	0	0
---	0	1	1	1
---	0	1	1	0
---	0	1	0	1
---	0	1	0	0
---SEL1---	0	0	1	1
---	0	0	1	0
---	0	0	0	1
---	0	0	0	0

$$1 \text{ MB} = 1024 \text{ kb} =$$

$$= 1024 \cdot 1024$$

byte

$$= 2^{10} \cdot 2^6 \text{ byte}$$

$$= 2^{20} \text{ byte}$$

⇒ 20 bit
variante

$$SEL2 = \overline{A_{20}} + A_{19} + A_{18}$$

$$SEL1 = A_{20}$$

$$SEL3 = \overline{A_{20}} + A_{19} + \overline{A_{18}}$$

	A ₂₀	A ₁₉	A ₁₈	A ₁₇
SBL 7	1	1	1	1
SBL 6	1	1	1	0
SBL 5	1	1	0	1
SBL 4	1	1	0	0
- - - SBL 3 - - -	1	0	1	1
	1	0	1	0
- - - SBL 2 - - -	1	0	0	1
	1	0	0	0
- - - - -	0	1	1	1
- - - - -	0	1	1	0
- - - - -	0	1	0	1
- - - - -	0	1	0	0
- - - SBL 1 - - -	0	0	1	1
- - - - -	0	0	1	0
- - - - -	0	0	0	1
- - - - -	0	0	0	0

$$SBL 4 = \overline{A_{20}} + \overline{A_{19}} + A_{18} + A_{17}$$

$$SBL 5 = \overline{A_{20}} + \overline{A_{19}} + A_{18} + \overline{A_{17}}$$

$$SBL 6 = \overline{A_{20}} + \overline{A_{19}} + \overline{A_{18}} + A_{17}$$

$$SBL 7 = \overline{A_{20}} + \overline{A_{19}} + \overline{A_{18}} + \overline{A_{17}}$$

Es. 6:

Execute: ?

CP1 RA, V

Determinare la fase di *Execute* dell'istruzione CP1 (RA), V che ha l'effetto di copiare il contenuto della locazione di memoria di indirizzo simbolico V nella locazione di memoria il cui indirizzo è contenuto in RA, assumendo che nel formato in linguaggio macchina i 6 bit più significativi siano dedicati al codice operativo, i 5 bit successivi alla specifica del registro RA e i 21 bit meno significativi alla specifica dell'indirizzo V.

$$M[RA] \leftarrow M[V]$$

Ex. 6:

CP1 54Y RA, V 21 5.1r
66Y 66Y

$$M[RA] \leftarrow M[V]$$

Execute: ?

T5: $(0^{11} || IR[20:0])_{out}, MAR_{in} ; MAR \leftarrow V$

T6: MRD

T7: MRD, DTR_{in} ; DTR \leftarrow M[V]

T8: RA_{out}, MAR_{in} ; MAR \leftarrow RA

T9: MWR, DTR_{out}, SE L DTR_{in}

T10: " " " ; M[RA] \leftarrow M[V]