RETI LOGICHE

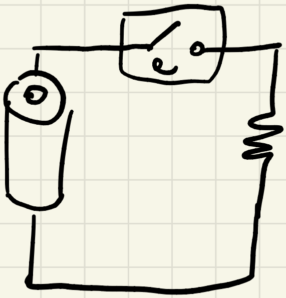
A. BOGLIOLO

28/09

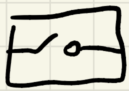
$y = f(x)$

y ↓ VARIABILE DIPENDENTE

x ↓ VARIABILE INDIPENTE



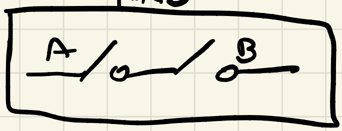
$\longrightarrow f(e) = e$



1 VARIABILE

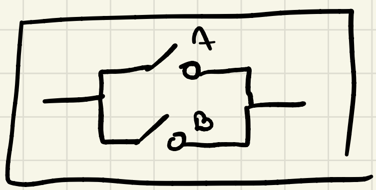
2 STATI

AND



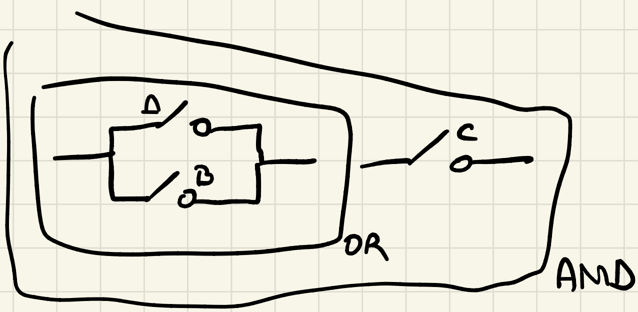
2 VARIABILI

4 STATI



2 VARIABILI

OR

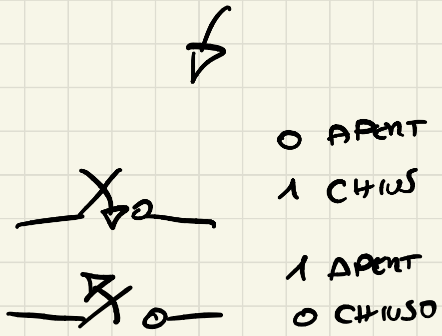


$f(A,B,C) = (A \text{ OR } B) \text{ AND } C$

↓

$(A \times B) \cdot C$

NEGAZIONE $\Rightarrow$ convenzione nel passaggio o non passaggio corrente



OVVERO IL NOT

TEORIA DELLE INFORMAZIONI

OGNI INFORMAZIONE SI PUÒ RAPPRESENTARE CON UN NUMERO FINITO DI VALORE 0, 1

OGNI CIFRA BINARIA SI CHIAMA BIT

ALGEBRA DI BOOLE

AND, OR, NOT NELLA LOGICA BOOLEANA $B = \{0, 1\}$

$$f = B^n \to B$$

OGNI ESPRESSIONE PUÒ ESSERE RAPPRESENTATA CON LA CONCATENAZIONE DI AND, OR, NOT

A	B	h
0	0	0
0	1	1
1	0	1
1	1	0

$$h = a'b + ab'$$

$$a' = \overline{a} = \text{NOT } a$$

5/10

(ESSERE DETTATO)

DIGITALE → TUTTO CIÒ CHE SI PUÒ RAPPRESENTARE IN CIFRE

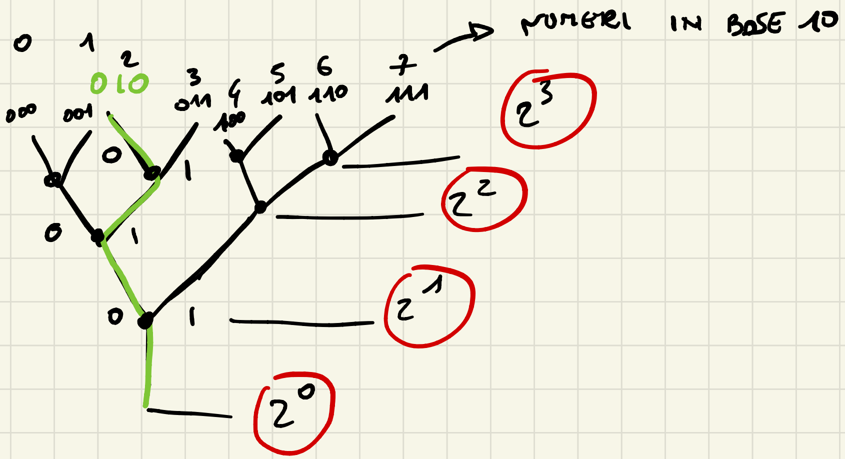
È TUTTO CIÒ CHE PUÒ ESSERE RAPPRESENTATO CON UNA SEQUENZA FINITA DI SEGNI DA UN ALFABETO FINITO

RAPPRESENTAZIONI DIGITALE

- IN BINARIO

INFORMAZIONE È RIDUZIONE DI INCERTEZZA

CONTENUTO INFORMATIVO (BIT) (0,1)



FINITO ALFABETO S SIMBOLI, N CARATTERI

S^N DIFFERENT WORDS

S^N DIFFERENT CONFIGURATIONS

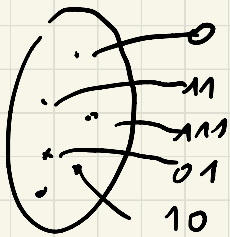
BYTE = 8 BITS $2^8 = 256$

CODICE
- INSIEME DI PAROLE
- INSIEME DI REGOLE

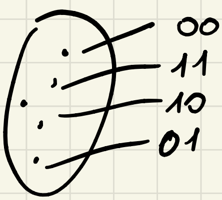
PAROLE DI CODICE

CODIFICA → ATTRIBUZIONE DI UN SIGNIFICATO AL CODICE SECONDO DETERMINATE REGOLE

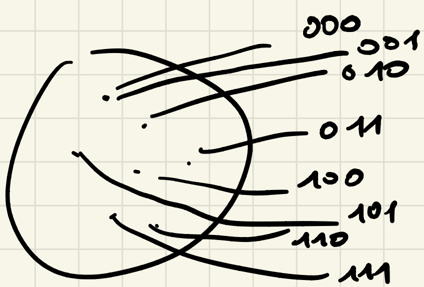
CODIFICA NON RIDONDANTE → UN ELEMENTO A UN SOLO SIGNIFICATO



LUNGHEZZA COSTANTE → STESSO NUMERO DI CARATTERI



ESATTO → UNA CODIFICA ESATTO È UN ASSEGNAZIONE DI TUTTI GLI ELEMENTI HANNO UNA RAPPRESENTAZIONE NON AMBIGUA



MINIMO NUMERO DI CIFRE PER RAPPRESENTARE TUTTI I SIGNIFICATI

$$N = \lceil \log_s M \rceil$$ → APP. PER ECCESSO ALL' INTERO

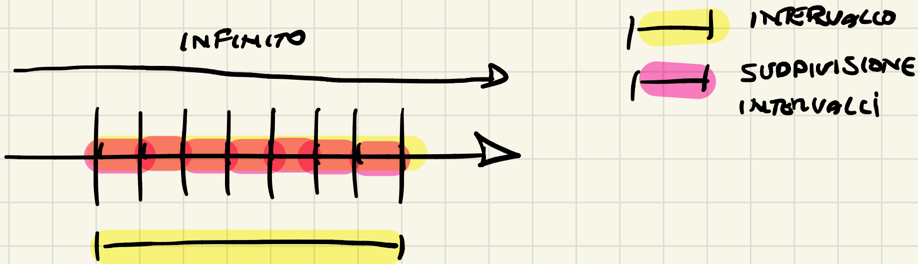
$$s^N \geq M$$

10 / 10

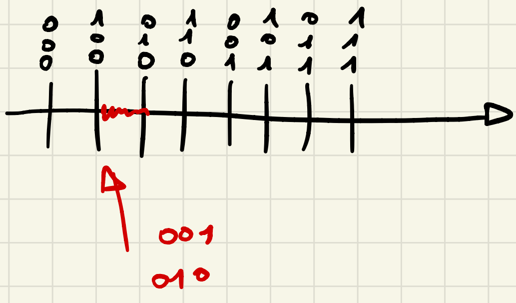
RAPPRESENTAZIONI SET INFINITI (CODIFICA)

↳ DATO CHE NOI LAVORIAMO CON NUMERI LIMITATI DOBBIAMO LIMITARE ANCHE I SET INFINITI
DIMENSIONANDO LA PROPRIA MACCHINA, PER CUI LA RAPPRESENTAZIONE DI QUELLO CHE VOGLIAMO SIA ESATTO

SET INFINITI INFINITAMENTE DENSO, INFINITAMENTE CONTINUO SONO PIÙ PROBLEMATICI, QUINDI BISOGNA APPLICARE LA TECNICA DI DISCRETAZIONE, LO SUDDIVISIAMO IN TANTI INTERVALLI, LA CUI SOMMA PRODUCE L'INTERO



SUDDIVISIONE INTERVALLI (SOTTOINSIEME DELL'INSIEME)



OGNI SUDDIVISIONE È UNA APPROSSIMAZIONE DELL'INSIEME

RAPPRESENTAZIONE DEI NUMERI INTERI SENZA SEGNO

$527_{10} \rightarrow \begin{array}{ccc} 100 & 10 & 1 \\ 5 & 2 & 7 \\ 10^2 & 10^1 & 10^0 \end{array}$ IN BASE 10

$5 \times 100 \quad 2 \times 10 \quad 7 \times 1 \quad = 527$

$\begin{array}{ccccc} 16 & 8 & 4 & 2 & 1 \\ 1 & 0 & 0 & 1 & 1 \\ 4 & 3 & 2 & 1 & 0 \end{array}$

$$2^4 \cdot 1 + 2^1 \cdot 1 + 2^0 \cdot 1 = 16 + 2 + 1 = 19$$

MAX NUM RAPP. IN BIN

perché c'è lo 0

$$2^8 = 2^8 - 1 = 255$$

$$c_{m-1} \cdot 2^{m-1} + c_{m-2} \cdot 2^{m-2} + \dots c_2 \cdot 2^2 + c_1 \cdot 2 + c_0 =$$

$$= 2\left(c_{m-1} \cdot 2^{m-2} + c_{m-2} \cdot 2^{m-3} + \dots \, c_2 \cdot 2 + c_1\right) + c_0$$

8 4 2 1

13 ... $\underbrace{00001101}$ 8 bit

6 1 0
3 0 1
1 1 2
0 1 3
0 0
0 0
0 0
0 0
0 0

$8 + 4 + 1 = 13$

0,625

101

1,25 $\times 2$

0,5 $\times 2$

1,0 $\times 2$

$$\overbrace{\overset{n}{\boxed{\quad\quad\quad}}\,,\,\overset{m}{\boxed{\quad\quad}}}^{n+m}$$

$$\frac{2^{n+m}-1}{2}$$

OGNI VOLTA CHE SPOSTO LA VIRGOLA DIVIDO PER 2

0,9

1,8

1,6

1,2

0,4

0,8

1,6

1,2

0,4

0,8

1,6

1,2

PERIODICO

12 | 10

ARITMETICA BINARIA

$$\begin{array}{r} {\scriptstyle 1\,1\,1\;} \\ 0111+ \\ 0001= \\ \hline 1000 \end{array} \qquad 7+1=8$$

$$\begin{array}{r} 0110\,\cdot \\ 11= \\ \hline 0110 \\ 0110+ \\ 0000++ \\ 0000+++ \\ \hline 0010010 \end{array} \qquad 6\cdot 3 = 18 \qquad 16+2=18$$

SEGNO NUMERICO

0 → POSITIVO
1 → NEGATIVO

CONVENIENZA DEL NON MODIFICARE UN NUMERO SENZA SEGNO DA UNO CON

1001101 → CIFRA PIÙ SIGNIFICATIVA
0011011

$$(-1)^S \qquad S=(0,1)$$

1° CONV.

-3 1 0011 3 0 0011

2° CONV.

3 0 0011 -3 1 1100 COMPLEMENTO A 3

3° CONV.

COMPLEMENTO A 3

3 0 0011 -3 1 1101 = $(0\ 0011)' + 1$

```
3,5      011,1
2,5      010,1
        ------
          0111
         00000+
        0111++
       0000+++
      --------
       01000,11
```

```
  35
  25
 ----
 175
 70+
 ----
 8,75
```

RAPP.

8 bit

3

0	0	0	0	0	0	1	1

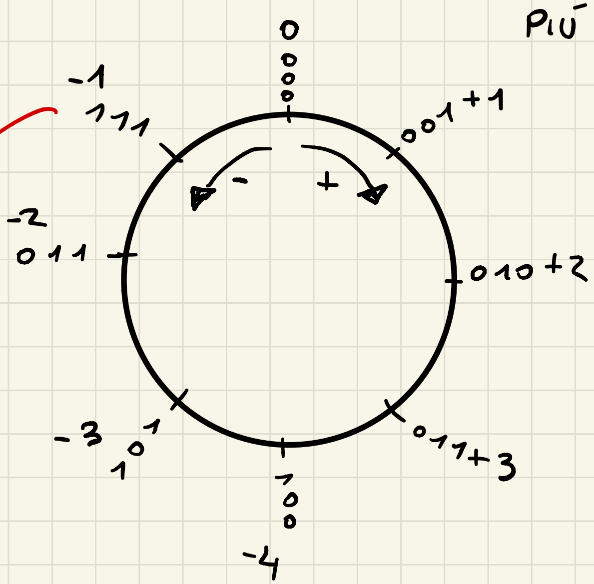
1 0 0 0 0 0 1 1 - 1° CONV
1 1 1 1 1 1 0 0 - 2° CONV
1 1 1 1 1 1 0 1 - 3° CONV

4

0	0	0	0	0	1	0	0

1 0 0 0 0 1 0 0 - 1° CONV
1 1 1 1 1 0 1 1 - 2° CONV
1 1 1 1 1 1 0 0 - 3° CONV

PIÚ USATA 3° CONVENZIONE



IL NUMERO CHE BISOGNA SOMMARE 1 PER FARLO ARRIVARE A 0

-2 3 CONV

110

negato

101 +

1 +1

110

3 CONV

$2^n - V$

n = numero di bit

V = valore modulo del numero

ES

$n = 3$

$V = 2$

$2^3 - 2 \rightarrow 6 \rightarrow 110$

DIFF. TRA DUE NUMERI

$a - b$

$a + (-b)$ $\quad a + (2^m - b) = 2^m + a - b = 2^m - (b-a)$

$\downarrow$

$2^m - b$

$-(b-a) = a-b$

CI PERMETTE DI FARE LE ADDIZIONI PER CALCOLARE LE SOTTRAZIONI

$3-2$

COMP. 3

$011 \quad 010 \rightarrow \boxed{110 \quad 2^3 - 2}$

$$\begin{array}{r} 011 \; + \\ 110 \; = \\ \hline 1|001 \end{array}$$

$2-3$ COMP 3

$010 \quad 101$

$$\begin{array}{r} 010 \; + \\ 101 \\ \hline 011 \end{array} \; +$$

BIT PIÙ SIGNIFICATIVO NEGATIVO

000 SI RIAPPLICA COMP 3

-001

$$\begin{array}{r} 010 \; + \\ 010 \; = \\ \hline 100 \end{array}$$

17/10

NUMERI RAPPRESENTATI IN VIRGOLA MOBILE

$$s\ 0.M * b^{se\ E}$$

peso della mantissa

S	M	se	E

n di bit

s → segno
M → mantissa
b → base
se → segno esponente
E → esponente

$n = 1 + m + 1 + e$

n = numero di bit

ESEMPIO

3,25

1	4	1	2
S	M	se	E

= 8 bit

0 1 1 0 1 0 1 0 = 3,25 FLOATING POINT

11,01

M = 1101

0.1101

S = 0

Se = 0

E = 10

ESEMPIO 2

6,5

0.1101

110,1

0 1 1 0 1 0 1 1 = 6,5 FLOATING POINT

1,625

1,101 0 1 1 0 1 0 0 1 = 1,625 FLOATING POINT

0,625

0,101 0 1 0 1 0 0 0 0 = 0,625 FLOATING POINT

0,25

0,01 0 1 0 0 0 1 0 1 = 0,25 FLOATING POINT

0 1 0 1 0 0 1 0

0,1010 → 10,10 = 2,5

0 1 1 1 0 0 1 1

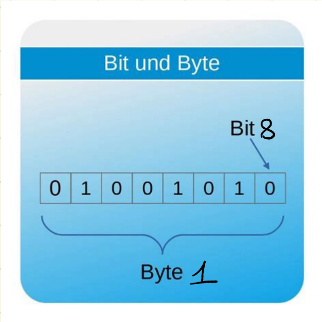
IEEE 754

19/10

INFORMAZIONI TESTUALI

ASCII 7-bit → 128 configurazioni

ASCII 8-bit esteso → 256 conf.



INFORMAZIONI

IMMAGINE DIGITALE (MATRICE DI PIXEL COLORI)

m_x

m_y

MATRICE INSCRITTA DISCRETIZZATA

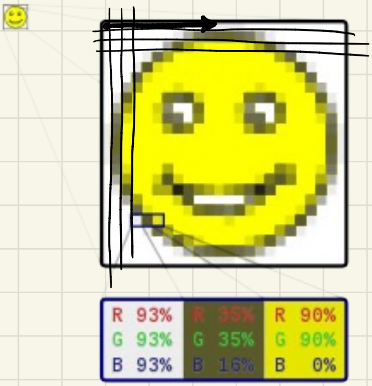
INTERSEZIONE NULLA (L'INSIEME È TUTTO IL RIQUADRO)

$m_x \times m_y$

COLORE (PROFONDITÀ)

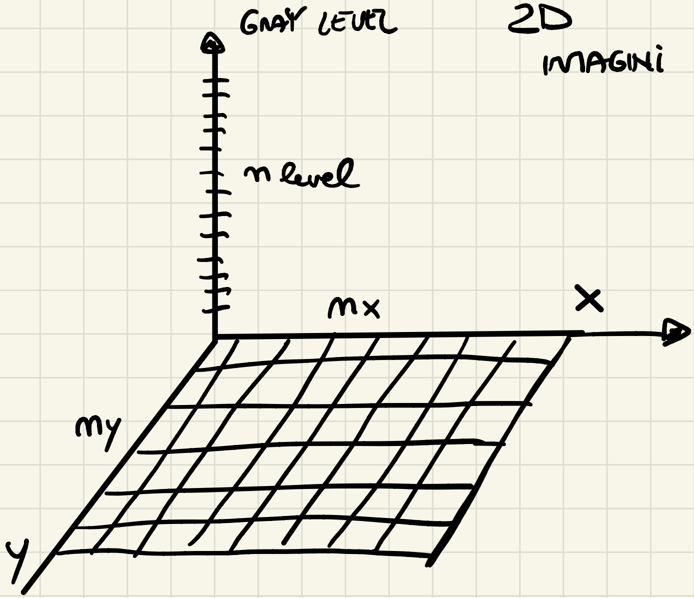
APPLICHIAMO DISCRETIZZAZIONE

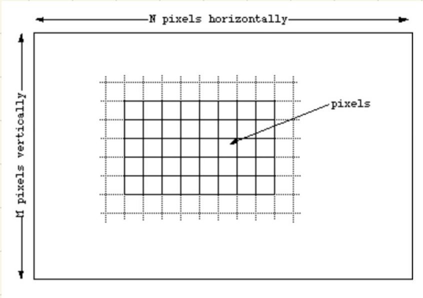
SI CODIFICA DA DESTRA VERSO SINISTRA



$$\boxed{m_x \cdot m_y \cdot \log_2 n_{liv}}$$

2D IMAGINI





Dimensione immagini	profondità colore	dimensione file
	1 bit	2 KB
128 x 128	8 bit	16 KB
	24 bit	48 KB
	1 bit	8 KB
256 X 256	8 bit	64 KB
	24 bit	192 KB
	1 bit	128 KB
1K X 1K	8 bit	1 MB
	24 bit	3 MB

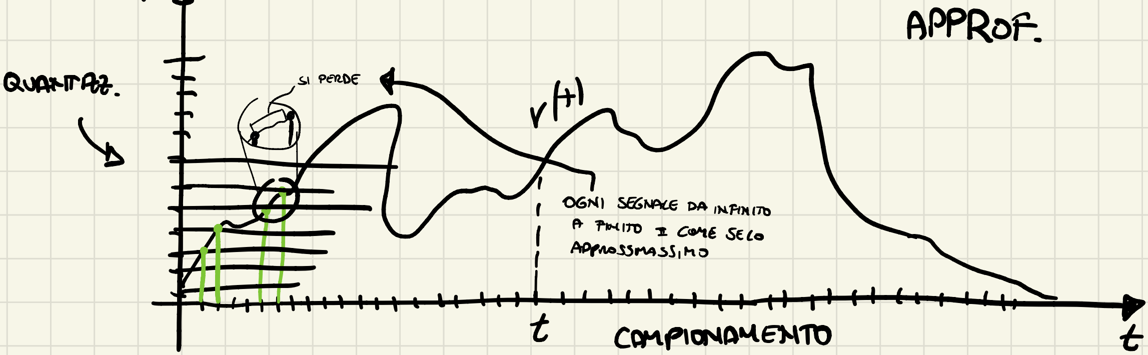
http://paulbourke.net/dataformats/bitmaps/index_it.html

APPROFONDIMENTO RGB

Segnali → grandezze fisiche che variano nel tempo

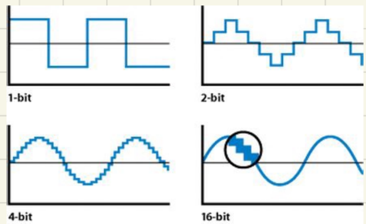
https://it.wikipedia.org/wiki/Teoria_dei_segnali

APPROF.



QUANTIZ. → DA INFINITO A FINITO

CAMPIONAMENTO → IDEALE CLOCK



DIMENSIONE SEGNALE

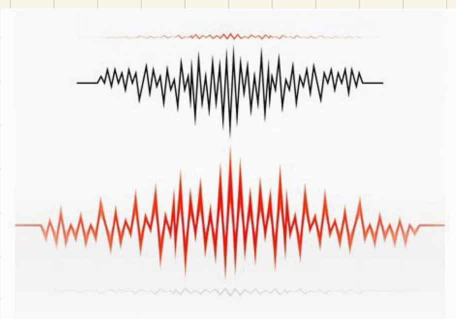
$$SIZE = S_{RATE} \cdot T \cdot S_{SIZE}$$

S_{SIZE} → $\log_2 n$ liv

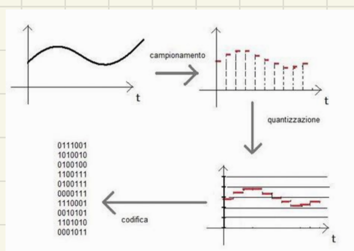
S_{RATE} → FREQ. DI CAMP.

T → DURATA

S_{SIZE} → n° DI BIT DEDICATO A OGNI BIT



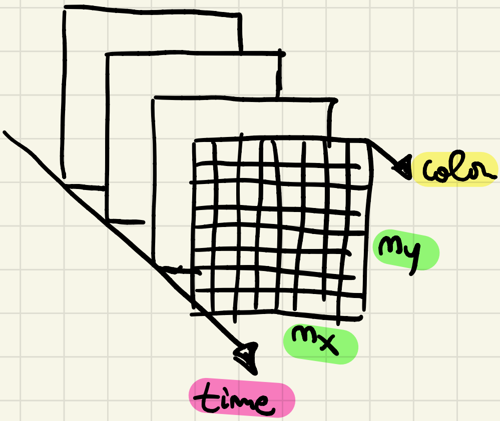
TEORIA DEL CAMPIONAMENTO



https://it.wikipedia.org/wiki/Teorema_del_campionamento_di_Nyquist-Shannon

VIDEO

$$size = s_{rate} \cdot T \cdot \log_2 n_{col} \cdot n_x \cdot n_y$$



s_{RATE} = frame rate
n_{col} = number of color
$n_x n_y$ = frame size

COMPRESSIONE https://it.wikipedia.org/wiki/Compressione_dei_dati

NALLENT . ENCODING

https://it.wikipedia.org/wiki/Compressione_dell%27immagine

IMMAGINE

VIDEO VENGONO TRASMESSI SOLO I FRAME CHE CAMBIANO

24/10 RIDONDANZA (MODELLO DEGLI ERRORI)

ERRORE bit ricevuto il suo opposto, gli errori sono indipendenti

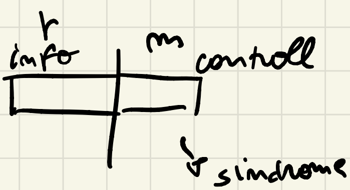
PROBABILITÀ DI ERRORE

$p_{\text{errore}} = 1-(1-p)^n \cong np$ (0,1) 0 sbagliato, 1 corretto

p la percentuale di volte in cui eseguo il test, divido tutti i risultati

https://it.wikipedia.org/wiki/Distanza_di_Hamming

NUMERO DI BIT CHE DIFFERISCONO



$2^m \geq r+m+1$

26/10 COSTRUZIONE CODICE HAMMING (CODIFICA E DECODIFICA)

1	2	3	4	5	6	7	8	9	10	11	12	13

ES 9

BIT CONTROLLO POTENZE MAGGIORI DI 2 (4)

BIT INFORMAZIONE (5)

r	m	n
1	2	3
2	3	5
3	3	6
4	3	7
ES 5	4	9
6	4	10

⋮

1 0 0 1 0

1 0 0 1 1

1 0 1 0 0 ← 1

1 0 1 1 0 ← 2

1 0 1 1 1

1 1 0 0 0

⋮

CODICE DI HAMMING

①

1	2	3	4	5	6	7	8	9	10
1	0	1	1	0	1	0	0	0	
×		×		×		×		×	
	×	×			×	×			
			×	×	×	×			
							×	×	× × × × × ×

②

1	2	3	4	5	6	7	8	9	10
0	0	1	0	0	1	1	0	0	
×		×		×		×		×	
	×	×			×	×			
			×	×	×	×			
							×	×	× × × × × ×

DECODIFICA

1	2	3	4	5	6	7	8	9					
1	0	1	1	1	1	0	0	0					
x		x		x		x		x					1
	x	x			x	x							0
			x	x	x	x							1
							x	x	x	x	x	x	0

PESO

0 1 0 1 = SINDROME

4 + 1 = 5

CODIFICA → 1 0 1 → 1 0 1 1 0 1

(CORREZIONE ERRORE SINGOLO

1	2	3	4	5	6	
1	0	1	1	0	1	
x		x		x		
	x	x		x	x	
			x	x	x	x

1	0	1	0	0	1	
x		x		x		x
	x	x			x	x
			x	x	x	x

1 0 0

2/11 (dispense 3.01 ep2) LABORATORIO

ALGEBRA DI BOOLE

- SISTEM. MAT. DEDUT.
- INSIEME DI ELEM.
- UN SET DI OPERAT.
- UNA SERIE DI ASSIOMI

OP. BINARI → OPERANO SU DUE OPERANDI

OR, AND

NOT → UNARIO → OPER. SU UN OPERATORE

$$A = \{ \underbrace{A}_{\text{INSIEME BOOLEANO}} ; + ; \cdot ; ' ; \underbrace{0 ; 1}_{\text{COSTANTI}} \}$$

ASSIOMI (in A)

$xy = yx \qquad x+y = y+x$ (COMMUTATIVA)

$x \cdot 1 = x \qquad x+0 = x$ (IDENTITÀ, ELEMENTO NEUTRO)

$x \cdot (y+z) = (x \cdot y) + (x \cdot z) \qquad x + (y \cdot z) = (x+y) \cdot (x+z)$ (DISTRIBUTIVA)

LEGGE DEL COMPLEMENTO

$\forall x \in A \; \exists x' \in A$:

$x \cdot x' = 0 \qquad x + x' = 1$

$A = \{a, a', b, b', \dots\}$

BOOLEAN (SWITCHING) ALGEBRA

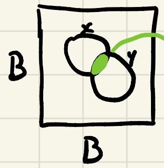
$$\mathcal{B} = \{B, +, \cdot, ', 0, 1\}$$

UGUALE AD A in più le funzione booleana

- OPERATORI

AND $\quad B \times B \to B \quad z = x \cdot y = xy$

x	y	xy
0	0	0
0	1	0
1	0	0
1	1	1

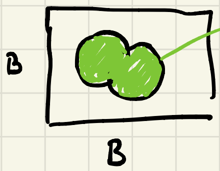


insieme x AND y

$x \cap y$

OR $\quad B \times B \to B \quad z = x + y$

x	y	x+y
0	0	0
0	1	1
1	0	1
1	1	1



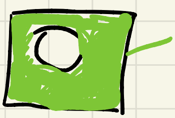
insieme x OR y

$x \cup y$

NOT $\quad B \to B$

$z = x' = \bar{x}$

x	x'
0	1
1	0



complemento di x

x'

- FUNZIONI BOOLEANE $f : B^n \to B$

TABELLA DELLA VERITÀ (QUANDO QUESTA FUNZIONE È VERA?)

n° variabili conf poss 2^n

n $2^3 = 8$ var

a	b	c	f (0,1)
0	0	0	1
0	0	1	0
0	1	0	
0	1	1	1
1	0	0	0
1	0	1	
1	1	0	1
1	1	0	1

2^n = funz finit 2^{2^n}

variabili indipend

var. dipend

f di una sola variabile

$f(x) = x$

$f(x) = x'$

$f(x) = xx' = 0$

$f(x) = x + x' = 1$

LEGGE DEL COMPLEMENTO

ORDINE DI RISOLUZIONE

ESPRESSIONI BOOLEANE → (PARENTESI → UNARIE → AND → OR)

$f = a \cdot b + c' = ab + \bar{c} = (ab) + \bar{c}$ le espressioni si verificano con la tabella della verità

Ci interessa di più partire dalla tabella della verità e trovare un espressione

PROPRIETÀ NON ASSIOMATICHE

- ASSOCIATIVA

$$x \cdot y \cdot z \xrightarrow{=} (x \cdot y) \cdot z = x \cdot (y \cdot z)$$

- ELEMENTI FORZANTI

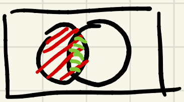
$$X \cdot 0 = 0 \qquad X + 1 = 1$$

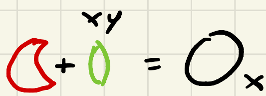
- IDEMPOTENZA

$$x \cdot x = x \qquad x + x = x$$

- LEGGE ASSORBIMENTO

$x + xy = x$





$x(x+y) = x$

- DE MORGAN (PRINCIPIO DI DUALITÀ)

$$(x \cdot y)' = x' + y' \qquad (x+y)' = x' \cdot y'$$

ESEMPI DIMOSTRAZIONI SLIDE

- INDUZIONE PERFETTA (FORZA BRUTA)
- MANIPOLAZIONE BOOLEANA (RIDOTIONE A LA FORMA PIÙ RIDOTTA delle espress. equivalente all'altra)

- FORME CANONICHE (forme di espressione ridotte alla minima configurazione equivalenti a tutte le espressioni equivalente ad essa)

Somme di min termini

- letterale ogni valore dell'espressione $x + yx = \{x \ yx\}$ letterali
- min termini $f(x, y, z)$ $\quad x, y', z \quad x, y, z'$ $\quad x', y, z$
- stabiliamo un'ordine

7/11

IMPORTANTE

OGNI ESPRESSIONE BOOLEANA SI PUÒ RAPPRESENTARE CON SOMMA DI MIN TERMINI

ESEMPIO SOMMA MIN TERMINI

a	b	c	f
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	0

minor numero di letterali

$$f = a'b'c + ab'c + abc'$$

$$= b'c(a'+a) + abc'$$

$$= b'c + abc'$$

ESPRESS. SLIDE (MINIMIZZAZIONE BOOLEANA)

$$f = a'b'c' + a'bc' + ab'c' + abc' + abc$$

$$= c'(a'b' + a'b + ab' + ab) + abc$$

$$\downarrow$$

$$b'(a'+a) + b(a'+a)$$

$$\downarrow$$

$$b' + b = 1$$

$$= c' + abc \longrightarrow = c' + abc' + abc = c' + ab(c'+c) = c' + ab$$

$$c' = c' + c'ab$$

$$x = x + xy$$

$$x = c'$$

$$y = ab$$

$$c' + abc$$

$$x + x'y = x + y$$

TEOREMA ESPANSIONE BOOLEANA

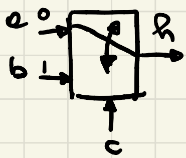
$f: B^n \rightarrow B$

SLIDE

MINIMIZZAZIONE BOOLEANA → RIDUZIONE LETTERALI, IL PIÙ POSSIBILE

CONDIZIONI DI INDIFFERENZA → CASELLE NON SPECIFICATE NELLA TABELLA DELLA VERITÀ
(DON'T CARE CONDITIONS)

SINTESI LOGICA → TAB VERITÀ → SOM MIN TERMINI → OTT. BOOLEANA



se $c=0 \rightarrow f=a$

$c=1 \rightarrow f=b$

a	b	c	f
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	1

$f = a'bc + ab'c' + abc' + abc$

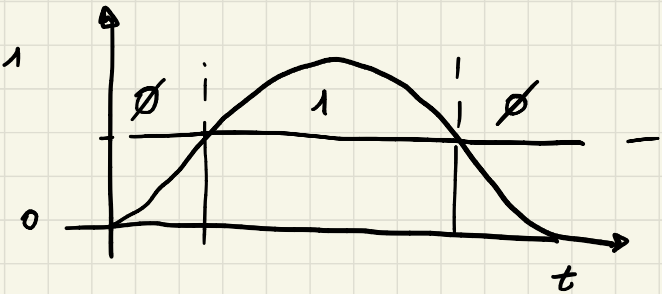
$= bc(a'+a) + ac'(b'+b) = bc + ac'$

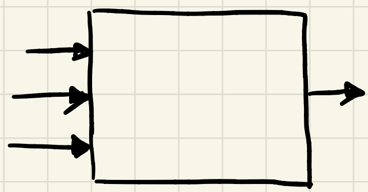
$c = a \quad f = 1\cdot a + \cancel{0\cdot b} = a$

$c = b \quad f = \cancel{0\cdot a} + 1\cdot b = b$

14/11

GATES → PORTE LOGICHE → Componente che prende in input uno o più segnali logici e restituisce in output uno o più segnali logici dipendenti dalle funzioni di configurazioni dei segnali di input

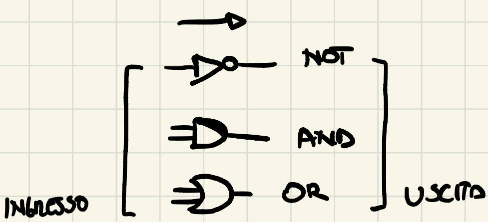




Valore minimo (0) → terra
Valore massimo (1) →

TRATTAZIONE ASTRATTA → TRATTAZIONE FISICA
ALGEBRA DI BOOLE → COMPONENTI ELETTRONICI

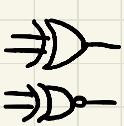
PORTE LOGICHE SPECIALI (VEDI 3.02 SLIDE)



$f = a' = \bar{a}$

$f = ab$

$f = a + b$



EXOR $\oplus$ $f = a \oplus b$

EXNOR $f = (a \oplus b)'$

TAB VERITÀ

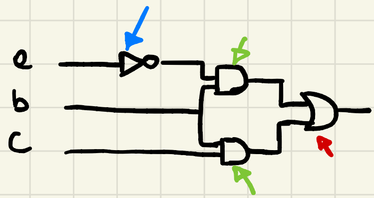
a	b	EXOR
0	0	0
0	1	1
1	0	1
1	1	0

PORTE LOGICHE COMPOSTE → RETI LOGICHE

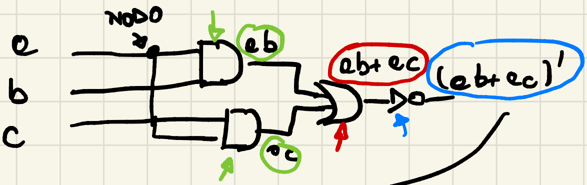
CORR BIUNIVOCA TRA → ESPRESSIONE BOOLEANA

$$h = a'b + b \cdot c$$

componenti $'$, $+$, $\cdot$



$$h = (ab + ac)'$$

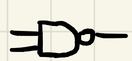


LIBRERIA → COMPONENTI CHE POSSO UTILIZZARE QUANDO SERVONO
(DAL TIPO NON DAL NUMERO) (AND, NOT)

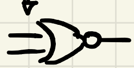
LIBRERIE FUNZIONALMENTE COMPLETE → RAPP DI QUALUNQUE FUNZIONE ESPRESSIONE

(AND, NOT, OR)

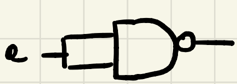
(NAND)



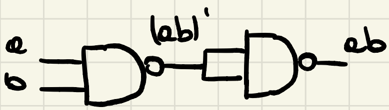
(NOR)



NAND DIM.

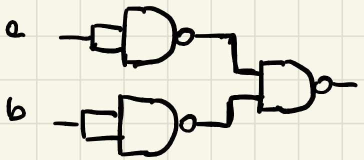


$(a \cdot a)' = a'$ → NOT



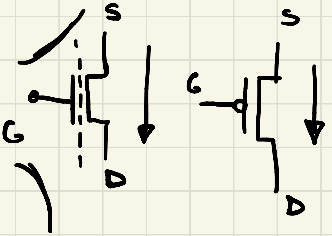
→ AND

$a+b = (a'b')' = ((aa)'(bb)')'$

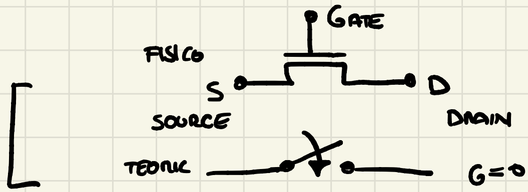


→ OR

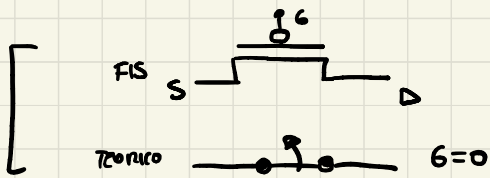
TRANSISTOR (FORMA FISICA) → STESSA NATURA DEGLI INTERRUTTORI A LIVELLO CONCETTUALE



NMOS



PMOS

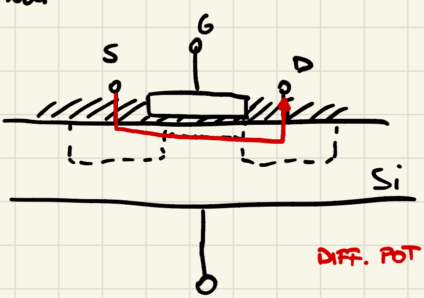


M O S FET

M → METAL

O → oxide

S → SILICON



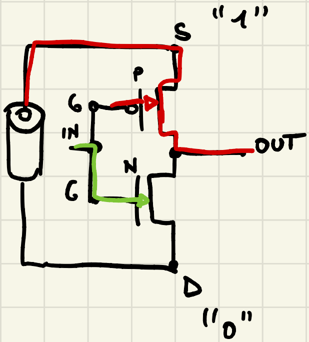
DIFF. POT TRA G B

← SOSTITUZIONE ATOMO DI SILICIO

REGIONI RICCHE DI ELETTRONI

SEMI CONDUTTORE

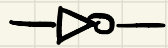
BULK

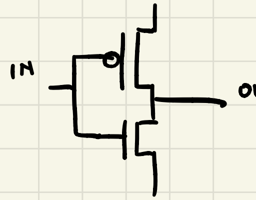


FCMOS → PORTE LOGICHE

in	out
0	1
1	0

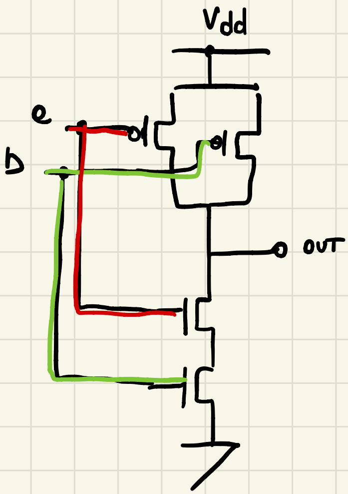
→ NOT

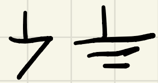




FCMOS

MAI ACCESI ENTRAMBI





MASSA
MESSA A TERRA

a	b	OUT
0	0	1
0	1	1
1	0	1
1	1	0

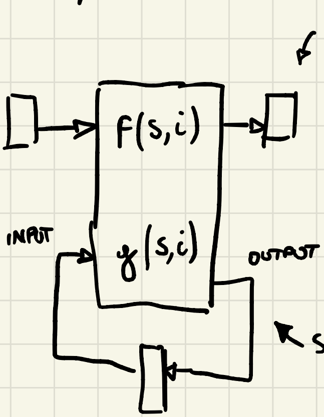
→ NAND

CORREZIONE VER.

Q.2 COMPLEMENTO A 2 PRIMO BIT SEGNO

111000

28/11 FLIP FLOP → REGISTRI



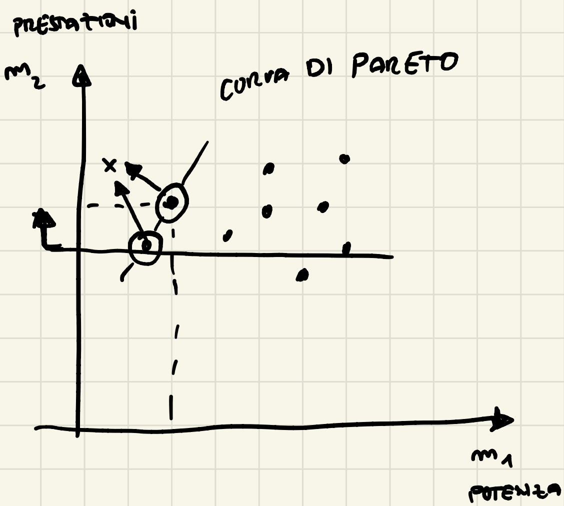
↖ SINCRONO → CIRCUITO CHE USA IN TUTTI I SEGNALI UN SEGNALE DI CLOCK E PRESENTE E MANTIENE IL SINCRONISMO

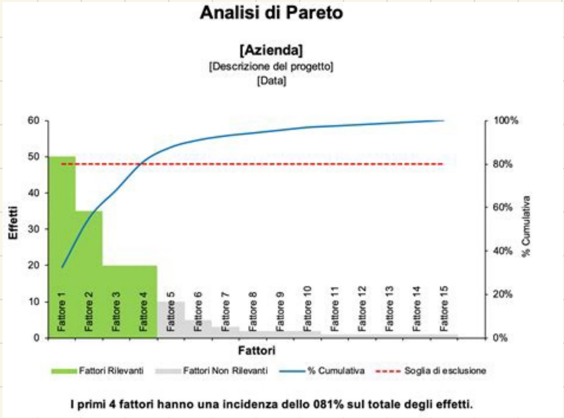
GENERAL

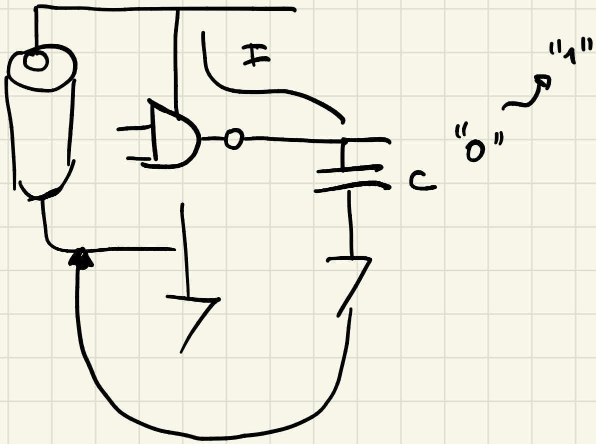
SISTEMI DIGITALI

METRICHE:

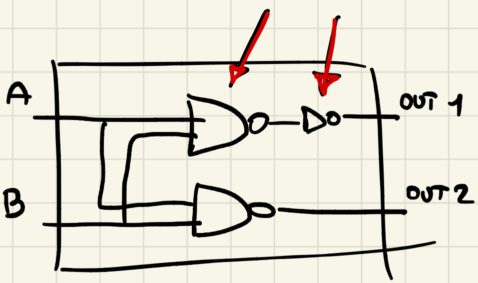
- AREA (A)
- PERFORMANCE → tempo propagazione
  → tempo contaminazione
- POTENZA (W)

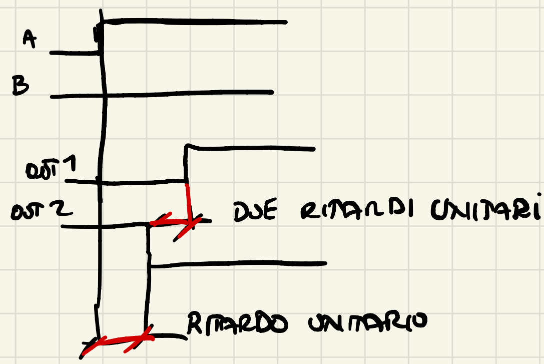






PRESTAZIONI





T_p 2 PIÙ GRANDE

T_c 1 PIÙ PICCOLO

T_p pin-to-pin

	OUT1	OUT2
A	2	1
B	2	1

PORTE LOGICHE CHE ATTRAVERSA

SINTESI LOGICA → SPEC FUNZ → TAB VERITÀ

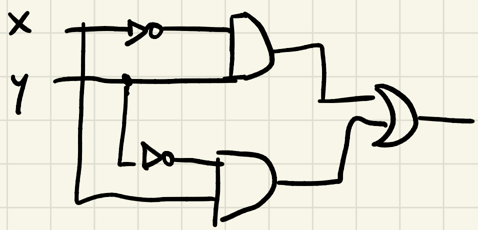
TOP-DOWN

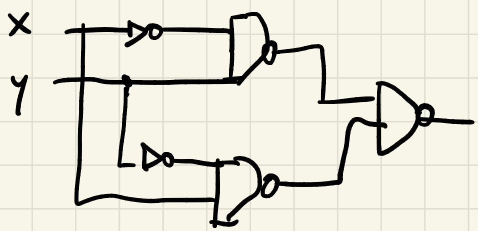
- MODULARI
- SCALABILI

FULL ADDER

S.01

EXOR



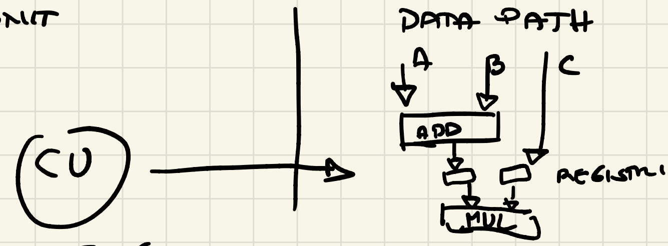


RTL

13 / 12

RTL → REGISTRI
→ MACRO
→ MACCHINA A STATI FINITI

CONTROL UNIT | DATA PATH

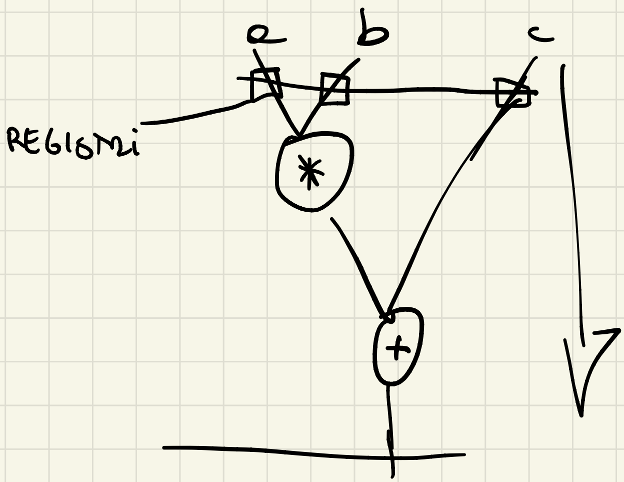


FMS

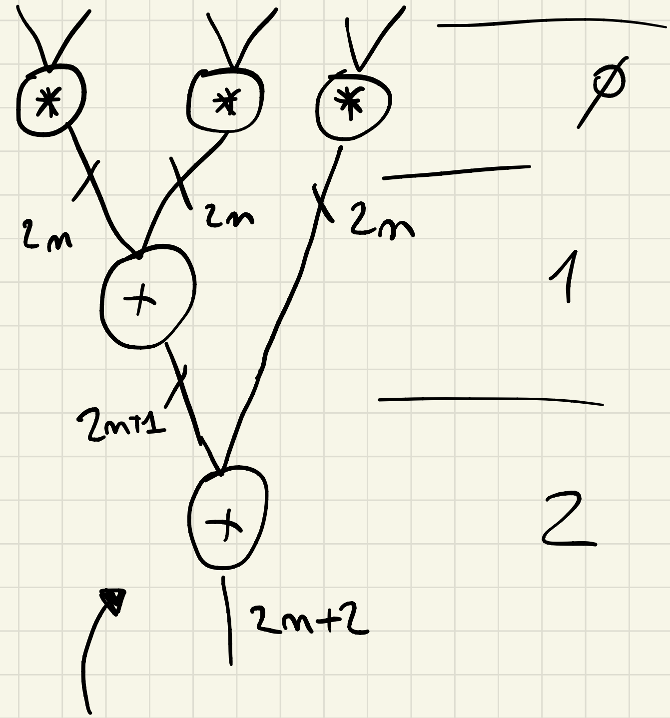
MACCHINA A STATI FINITI

$C * (A+B)$

$F = a * b + c$

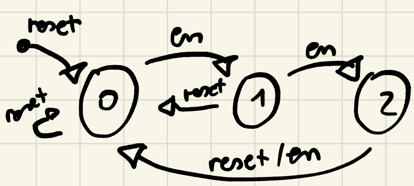


DATA FLOUGHT GRADE



$x_1 c_1 + x_2 c_2 + x_3 c_3$

DIAGRAMMA DEGLI STATI



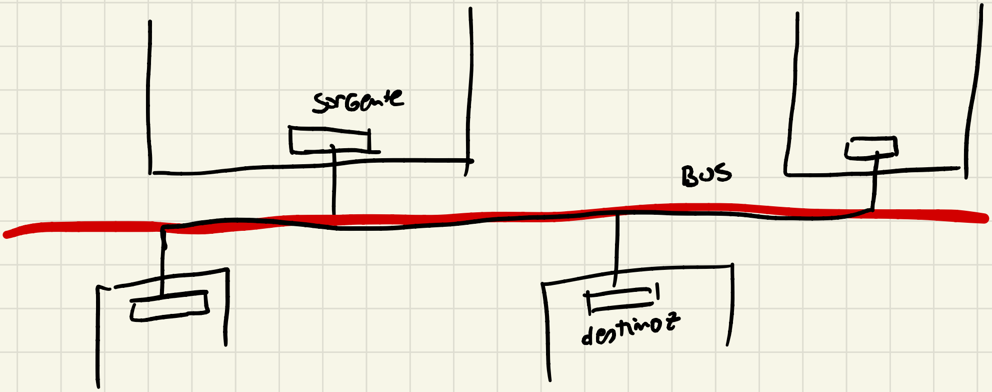
Controllo

TAB VERITÀ

$S_1 S_0$		a	b	c		next S_1 S_0
00	0	0	0	0	1	01
01	1	1	1	1	2	10
10	2	0	0	0	0	00
11		1	1	1		10

Controllo = S_0

BUS



11101.01

16 8 4   1

29 ,25

26	0
13	1
6	0
3	1
1	1
0	

11010

00101

00110

11,11

0,1111   010

0.11100   111   101

0.00101

110   101

24

$a\,e\,c = 1$ 101

$b'\,e\,c' = 1$ 100

e 0

c' 1

b 1

c 0

$\bar{0}\,\bar{1}\,1\;\bar{0}\,0\;1\,1\;\bar{0}\,1$

1
0
0
1

1001

1001