

BINARIO

BIN → C1 : NEG. BIT A BIT

BIN → C2 : C1 + 1 OPPURE BIN_{MS} PIÙ PROCEDIMENTO CLASSICO

MS: BIN NORMALE, NEGLI UNO DAVANTI UN BIT: 0 SE POSITIVO, 1 SE NEGATIVO

COD. BINARI ANALIZO BIN A GRUPPI DI 2 BIT. ULTIMO BIT CONFRONTATO CON 0.

- SE UGUALI: 0
- 01 : +1
- 10 : -1

COD. VIRGOLA FISSA (IEEE 754)

32bit = 1 8 23
 S E M

S=0 POSITIVO
 S=1 NEGATIVO

- LA PARTE INTEGRA DEL NUMERO DA CODIFICARE DEVE ESSERE $1 \leq X < 2$

SE NON È COSÌ MOLTIPLICA PER POTENZA DI 2

$B = 0,16219375 \rightarrow B = 1,296875 \cdot 2^{-3}$

$E = 127 + (-3) = 124 = 01111100$ PER VERIFICARE $-3 = E - 127 = 124 - 127 = -3$

LO SCARTO

0	296875	-2
0	59375	-2
1	1875	-2
0	375	-2
0	75	-2
1	5	-2
1		

$B = 0 \ 01111100 \ 01001100$
 FINO 23 BIT

SOMMA/SOTTRAZIONE NORMALIZZATA VIRGOLA FISSA

1. MA +/- - DEVO SPOSTARE A DESTRA IL NUMERO CON ESPONENTE PIÙ PICCOLO DI N BIT QUANTO LA DIFFERENZA DI ESPONENTI
1. MS SOTTRAENDO $n = e_A - e_B$

SI PONE SEMPRE LA PARTE INFERIORE.

- IL RISULTATO DEVE ESSERE PER TIPO 1. NORMALMENTE NORMALIZZATO
- IL RISULTATO ANCHE L'ESPONENTE DEL NUMERO PIÙ GRANDE, O L'ESPONENTE NORMALIZZATO A PARTIRE DA QUELLO PIÙ GRANDE.

MOLTIPLICAZIONE VIRGOLA FISSA

1- $e_{A \times B} = e_A + (e_B - 127) +$ [EVENTUALE MOLTIPLICAZIONE PERMANENTE DA CANCELLO PERMANENTE]

2- 1. MA *
 1. MB

3- SEGNO RISULTATO SE A SX DEL . NON HO 1. DEVO NORMALIZZARE

DIVISIONE VIRGOLA FISSA

- 1- $e_{A/B} = e_A - (e_B + 127)$
- 2,3- COME MOLTIPLICAZIONE MA CURIAMENTE DIVIDENDO AL PASSO 2.

MOLTIPLICAZIONE BOOTH

$A \times B \quad A = 13 \quad B = -9$

$A_{MS} = 01101 \quad A_{C2} = 10011 = -13$

$B_{MS} = 10111 \quad B_{BOOTH} = -1 + 100 - 1 = -9_{BOOTH}$

$01101 \times$
 $-1100-1 =$

QUANDO MOLTIPLICO PER -1 NEI ARZUALI SCRIVO A_{C2} ED ESTENDO IL BIT A SX.

- IL BIT A SX LO ESTENDO SEMPRE.

111110011

0001101

110011

$1110001011 \rightarrow P = -117$

$-2^8 + 2^8 + 2^7 + 2^3 + 2^1 + 2^0 = -117$

MOLTIPLICAZIONE TRAFICI

13: $01101 \times$
 -9: -10111

BIN IN MS CLASSICO PONGO DAVANTI SOLO IL SEGNO -

DIVISO IN DUE PASSAGGI:

- 1 PASSAGGIO: MOLTIPLICO TUTTI GLI 1 DEL MOLTIPLICANDO TRALFÈ QUELLO A SX: -1 (TENDO SOLO IL BIT A SINISTRA)
 - 2 PASSAGGIO: MOLTIPLICO SOLO 1 A SX FACENDO PUNA CHE STA POSITIVO, OGNI LO ALL'INCONTRARIO (TENDO SOLO IL BIT A SINISTRA)
- TOTALE: 1 PASSAGGIO + 2 PASSAGGIO IN C2

1-PASSAGGIO:

01101
 01101
 01101
 0
 0

2-PASSAGGIO

0
 0
 0
 0

91: 0001011011

208: 0011010000 #BIT
 208: 0011010000 #BIT
 208: 0011010000 #BIT

TOTALE:

91: $0001011011 +$

208: 1100110000

COMPLEMENTATO IN C2, COME DA REGOLE

$-117: 1110001011$

PLA TUTTO PROGRAMMABILE - SI CONDIVISIONE TRATTINI.



SE UNA DELLE PORTE OR È USATA NON PUÒ ESSERE UNA COSA DEL TIPO $\rightarrow \Delta_i = 0$ NO
 DOPO FINE MEMBRI IL CAVO



SOLTANTO SULLA LINEA DELLA RETTORAZIONE C'È IL FLIP-FLOP.

PLA OR NON PROGRAMMABILE, NO CONDIVISIONE TRATTINI, SOLTANTO VINCOLO SUGLI INGRESSI DELLE PORTE.
 QUANDO SCRIVO LE ESPRESSIONI DELLE USCITE DELLE PORTE OR, NON POSSO USARE PER DUE PORTE OR DIVERSE LA STESSA PORTA ANZO COME INGRESSO. SE MI DIVERSE SCRIVERE LA STESSA PORTA ANZO, DEVO CREARNE UNA IN PIÙ ANCHE SE IDENTICA.

FUNZIONI

1° FORMA CANONICA: SOP - SUM OF PRODUCT: $abc + \bar{a}bc$ 2° FORMA CANONICA: POS - PRODUCT OF SUM: $(a+b)(\bar{a}+b)$

CRITERI DI COSTO MINIMIZZAZIONE: # PRODOTTI NELLA SOLUZIONE MIN # DI LETTERALI = # LETTERALI NELLA SOLUZIONE

ALGEBRA DI BOOLE

- $\bar{x} + \bar{y} = \overline{xy}$ DE MORGAN
- $[(xy)']_{wz} = (xy)wz$ IDENTITÀ (a.a=a) (a+a=a)
- $x + \bar{x} = 1$ INVERSO
- $abcd + \bar{a}b'cd = acd(b+b')$ DISTRIBUTIVA
- $a+1 = 1$ ELEMENTO NEUTRO
- $x+(xy) = x$ ASSORBITIVITÀ $x(x+y) = x$

TH. ESPANSIONE BOOLE/SHANNON

$$f(x_1, x_2, \dots, x_n) = x_1 \cdot f(0, x_2, \dots, x_n) + x_1' \cdot f(1, x_2, \dots, x_n)$$

MINIMIZZAZIONE EURISTICA DI RETTI SU PIÙ LIVELLI

- NODI DI INGRESSO: LE VARIABILI INDIPENDENTI DELLA RETTE x_0, \dots, x_n .
- NODI DI USCITA: VAR. DI USCITA, OGNUNA FUNZIONE f_0, \dots, f_m SOLTANTAMENTE Y
- NODI INTERNI: VAR. TEMPORANEE INTERNE ALLA RETTE. ANNOTATI CON ESPRESSIONI BOOLEANEE
- TRASFORMAZIONI VENGONO REALIZZATE ED ESSEGUITE SOLO SE NON RISOLVONO IL COSTO DELLA RETTE
- ELIMINATE IN GENERE AUMENTA IL # LETTERALI. $\Delta l = n(l-1) - l$ $l = \#$ LETTERALI EXPR. ELIMINATE $n = \#$ NODI IN CUI ESSA È SOSTITUITA
- COSTO: # LETTERALI (SI CONTA TUTTI ANCHE LE RIPETIZIONI)
- SE IMPOSTATA SULLA Δl_{max} , ACCETTO SOLO SE: $\Delta l < \Delta l_{max}$

QUINE - MCCUSKEY

I DC VENGONO CONSIDERATI COME 1 NODI INTERNI DI IMPLEMENTAZIONE NELLE FUNZIONI.
 MAX DISTANZA DI HANMING IN ESPANSIONE = 1 NEI CONFRONTI I DC DEVONO COINCIDERE, MAX 1 SOLA VARIABILE DI 1

- ESTENSIONE: FLAG NINTERINE SE PRECEDUTA DA ESPANSIONE, ALTERNATIVE QUELLE SENZA FLAG SONO IMPLICANTI PRIMI SE LO STESSO IMPLICANTE APPARE PIÙ VOLTE, DEVO CONSIDERARLO SOLO UNA VOLTA, (FLAG CONTIENE I GENERANTI)
- COBERTURA: IMPLICANTI (SOLO ON-SET) UN IMPLICANTE PRIMO È ESSENZIALE SE È L'UNICO A COPRIRE UN INTERESTING
- REGOLANDO UN IMPLICANTE PRIMO NELLA COBERTURA CATTURO TUTTE LE COLONNE COPRITE DALL'IMPLICANTE E LA RIGA CORRESPONDENTE ALL'IMPLICANTE.
- DOMINANZA DI RIGA: LA RIGA I DOMINA LA RIGA J, SE L'IMPLICANTE P_i COPRE TUTTI I NODI INTERNI DI P_j PIÙ ALTREVO UNO.
- DOMINANZA DI COLONNA: LA COLONNA I DOMINA J SE IL NODI INTERNE m_j È COPERTO DALLI STESSI IMPLICANTI DA CUI È COPERTO m_i PIÙ ALTREVO UNO.
- GLI IMPLICANTI SELETTI CON I NODI DI DOMINANZA SONO IMPLICANTI ESSENZIALI SECONDAI.
- LETTERALI CHE COPRONO UN IMPLICANTE PRIMO: $A \begin{matrix} a & b & c \\ 0,1 & 0,0 \end{matrix} = a'b'$ COSTO IN LETTERALI: 2
- COSTO IN IMPLICANTI PRIMI NELLA SOLUZIONE: # IMPLICANTI PRIMI IN SOLUZIONE SOLUZIONE: \sum IMPLICANTI PRIMI = f DA COBERTURA

FUNZIONI A PIÙ USCITE

ASSOCIO UNA MASCHERA DI AFFIDABILITÀ AGLI FUNZIONI (2 APPARTIENE, \emptyset NON APPARTIENE)

ESPANSIONE: FUNG SECONDO LA REGOLA:

MASCHERA	RACCOLTAMENTO	FRAGMERTA
$t_{ab} = 0...0$	NON VALIDO	NESSUNA
$t_{ab} = t_a$	VALIDO	SOLO PA
$t_{ab} = t_b$	VALIDO	SOLO PB
$t_{ab} = t_a = t_b$	VALIDO	PA E PB

COBERTURA: $\{a, b\}$ IMPLICANTI PRIMI: $\{a, b\}$ NEI NODI INTERNI SOLO ON-SET SIMBOLO DI COBERTURA (X) PER IL NODI INTERNE SOLO SE APPARTIENE A QUELLE FUNZIONE DOMINANTE: VALEGNO SOLO SE IL COSTO È INFERIORE O UEGUALE.

COSTO IN IMPLICANTI: SE UN IMPLICANTE FA PARTE DI GIÙ DELLA COBERTURA DI UN'ALTRA FUNZIONE E LO SI VUOLE UTILIZZARE, IL SUO COSTO È ZERO.

BRANCH & BOUND: LO UTILIZZO PER RISOLVERE TABELLE CIRCULARI (DOVE LE DOMINANZE NON RISOLVONO) (È UN ALGORITMO CHE SCEGLIE INDICANDO SU OGNI FORK LA IMPLICANTE E PESO TOTALE FIN LI)

FSM

1- DIAGRAMMA DEGLI STATI

2- TAB. STATI



3- RIDUZIONE



COMPAT. SPEC.: X, N
NON COMP. SPEC.: X, V
ESPANNO TUTTI I VINCOLI, CHE POSSONO ESSERE N
ESPANNO SOLO X.

X SE LE USCITE SONO DIVERSE, INTERVENIVA VINCOLO. V O N SE VADE IN UNO DEI 2 STATI COMPATIBILI.

3.1 - SE NON È NUNCA (OVVERO VI SONO DEI VINCOLI) (O STATI COMPATIBILI, V, N)

CLASSI MASSIMA COMPATIBILITÀ:

SI FANNO QUANDO HO ANALIZZATO IL PRIMO STATO DELLA ROOT, CIO' LE QUANDO OTTENGO UNO STATO UGUALE ALLA ROOT

ALGORITMO EURISTICO:

(SOLUZIONE GENERALE LE CLASSI PIÙ GRANDI PER COPRIRE TUTTI GLI STATI)

- 1- LISTA L1 VUOTA
- 2- b: ORDINARE CLASSI MASSIMA COMPATIBILITÀ SUL GRADO PER DIM. b: CLASSI PIÙ GRANDI c: INSERIRE IN LISTA L1 TUTTI I VINCOLI DELLA CLASSE d: ELIMINARE DALLA LISTA L1 E DAL GRUPO I VINCOLI SODDISFATTI e: ELIMINARE DALLA LISTA L1 E DAL GRUPO TUTTI I NODI, CHE RISULTANO INACQUIRI, CHE NON RAPPRESENTANO A NESSUN VINCOLO PRESENTE NELLA LISTA L1 E/O NEL GRUPO.
- 3- PROCEDO FINO A GRUPO VUOTO, LE CLASSI ANALIZZATE SI OTTENGONO IL VUOTO SENZA LA COPERTURA INDIVIDUATA, SE NON OTTENGO VUOTO, RENTRO CONVENIENDO SOTTO INIZIALE

USO I 2 CRITERI PER FLIP FLOP D. ALTAMENTE ASSEGNAIMENTO BINARIO. (CHE HOI QUASI MAI UTILIZZABO)

3.2 - TAB. MACHINA BINARIA

X u/o 0/1
SE C'È UNO STATO, CHE NON VIENE RACCOMUNDO A NESSUN STATO PRECEDENTE LO POSSO SELEZIONARE COME STATO DI PART.

4 - ASSEGNAMENTO STATI

- ALTA PRIORITÀ:

I° CRITERIO: SE DUE STATI HANNO, PER LO STESSO INGRESSO, LO STESSO STATO FUTURO

- BASSA PRIORITÀ:

II° CRITERIO: SE DUE STATI, SONO STATI PRECEDENTI DELLO STESSO STATO, CORRISPONDONO A INGRESSI ADIACENTI

TAB. STATI:	0	1	Assegnamenti
S0	S1/0	S2/0	S1S2
S1	S0/1	S3/0	S2S4
S2	S3/0	S5/0	S3S5
S3	S2/1	S4/0	S4S1
S4	S5/1	S1/1	S5S1
S5	S4/0	S0/1	

NOTA: QUANDO SI HA UN SOLO BIT IN INGRESSO, NON SI CANTANO DUE VOLTE LE COPIE, QUINDI SE C'È S1S2 NON C'È S2S1.

KARNAUGH

- RAGGRUPPARE IN POTENZE DI 2.
 - EXP. LOGICA RAGGRUPPAMENTO È IL NUMERO DEI LETTERALI CHE RINGRUPPO IN UNO, NESSUNO QUELLO CON VALORE ESISTE.
 - EXP. LOGICA MAPPA SONO EXP. DI COPERTURA.
- 1- INDICANTO PARTI: TUTTI I RAGGRUPPAMENTI POSSIBILI
 - 2- INDICANTO PARTI ESS.: SOLO QUELLO CHE COPRE I NON COPERTI DA ALTRI. (NON DEVONO PER FORZA COPRIRE L'INTERA MAPPA)
 - 3- COPERTURA MINIMA: INDICANTO PARTI ESS + INDICANTO PARTI NECESSARI A COPRIRE LA MAPPA
 - 4- CIRCUITO EVENTUALE ALTRE COPERTURE.

- BASSA PRIORITÀ: NEL CASO DI MACCHINA DI NEALY SE DUE STATI HANNO USCITE IDENTICHE PER QUALCHE INGRESSO, E OPURTUNO CHE ABBIAMO CODIFICHE ADIACENTI.

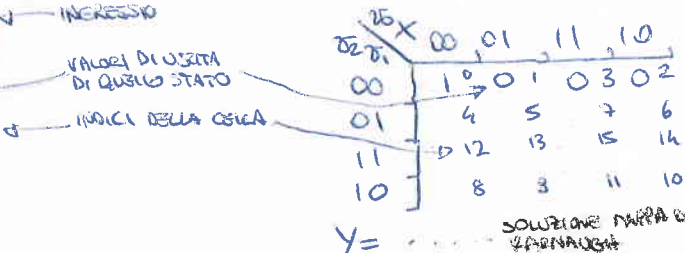
5 - TABELLA DELLE TRANSIZIONI

DALLA TAB. DEGLI STATI DEL FLIP FLOP D SOSTITUISCO A SUI LE CODIFICHE

6 - SINTESI DELLA FUNZIONE

RIPRODUCO LA TAB. DELLE TRANSIZIONI SECONDO LE CONDIZIONI BUNNIEF

Q2 Q1 Q0	0	1
0 0 0	0	0
0 0 1	0	0
0 1 0	0	0



CONTATORE

SONO MACCHINE DI MOORE, L'USCITA DIPENDE SOLO DALLI STATI CORRENTI, NON NECESSITANO DI AZIONI O ASSEGNAMENTI

1- Q2 Q1 Q0

0	0	0
0	1	0
1	0	0
0	0	1
0	1	1

TAB. STATI PROSSIMI:

0	0	0	0
0	1	0	0
1	0	0	0
0	0	1	0
0	1	1	0

PS = PRESENT STATE
NS = NEXT STATE
Y = USCITA

2 - ORDINO GLI STATI E RISPONDO L'USCITA

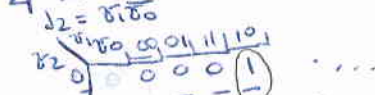
PS (a b c)	NS (a b c)	Y
0 0 0	0 1 0	0 1 0
0 0 1	0 1 1	0 1 1
0 1 0	1 0 0	1 0 0
0 1 1	0 0 0	0 0 0
1 0 0	0 0 1	0 0 1
1 0 1	-	x x x
1 1 0	-	x x x
1 1 1	-	x x x

TAB. eccitazione bistabile da usare, in questo caso JK

Q	Q+1	J	K
0	0	0	-
0	1	1	-
1	0	-	1
1	1	-	0

IL CIRCUITO DI JK CON Q+1 DA UOCCO A JK

4 - KARNAUGH: PER OGNI COLONNA DI PS UNA MAPPA



5 - SE RAPPRESENTA SU PLA/PAL, LE PARTI AND SONO I VALORI PRODOTTI DALLI STATI Q1 Q0, CIO' LE COPERTURE DI NS, POICHE SONO USCITA.

3 - TAB. eccitazione contatore

PS	NS	Y
0 0 0	0 1 0	0 1 0
0 0 1	0 1 1	0 1 1
0 1 0	1 0 0	1 0 0
0 1 1	0 0 0	0 0 0
1 0 0	0 0 1	0 0 1
1 0 1	-	-
1 1 0	-	-
1 1 1	-	-

POTREBBE ESSERE NECESSARIO CODIFICARE ANCHE L'USCITA. IN QUESTO CASO NON ERA RICHIESTA.

BISTABILITÀ

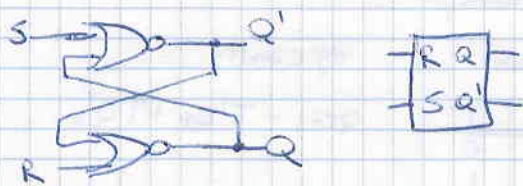
- ASINCRONI
- SINCRONI
 - A LIVELLO = LATCH
 - SUL FRONTI = FLIP-FLOP

→ FAMIGLIE:

- SR : LATCH, FLIP-FLOP, ASINCRONI
- D : LATCH, FLIP-FLOP
- JK : FLIP-FLOP
- T : FLIP-FLOP

TABELLA DELLE TRANSIZIONI DELLO STATO: CON SI CONVIENE
TABELLA DELLE EQUAZIONI: INDICA QUALE AZIONE DEVE ESSERE
APPLICATA AGLI INGRESSI PER PRODURRE
UNA DETERMINATA TRANSIZIONE DI STATO.

BISTABILE ASINCRONO SR (SET-RESET)



TAB. TRANSIZIONI STATO:

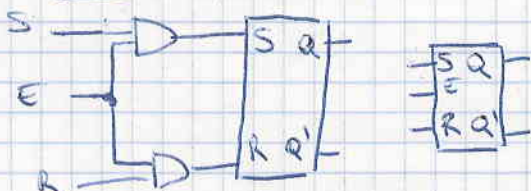
S	R	Q	Q'	
0	0	Q	Q'	
0	1	0	1	STATO DI RESET
1	0	1	0	STATO DI SET
1	1	0	0	NON UTILE

TAB. EQUAZIONI:

Q _T	Q _{T+1}	S	R
0	0	0	-
0	1	1	0
1	0	0	1
1	1	-	0

ES. LOGICA: $Q_{T+1} = S + R'Q_T$

LATCH SR SINCRONO



CON $E=0$ LO STATO RESTA
INVAIATO, INDIPENDENTEMENTE
DA S E R.

BISTABILITÀ TIPO D (DEADLY 0 DATA) (SOLUZIONE)

tab stato:

D	Q	Q'
0	0	1
1	1	0

Sono
Uscite

tab equazioni = tab transizioni

Q _T	Q _{T+1}	D
0	0	0
0	1	1
1	0	0
1	1	1

$$\text{eq caratt: } Q_{T+1} = D$$

Può essere ricavato da SR
Pensando:

$$S = D = R = D$$

BISTABILITÀ TIPO JK

tab stato:

J	K	Q	Q'
0	0	Q	Q'
0	1	0	1
1	0	1	0
1	1	Q'	Q

tab equazioni

Q _T	Q _{T+1}	J	K
0	0	0	-
0	1	1	-
1	0	-	1
1	1	-	0

eq caratt:

$$Q_{T+1} = JQ'_T + K'Q_T$$

BISTABILITÀ TIPO T (TOGGLE o TRIGGER)

tab stato:

T	Q	Q'
0	Q	Q'
1	Q'	Q

tab equazioni

Q _T	Q _{T+1}	T
0	0	0
0	1	1
1	0	1
1	1	0

eq caratt:

$$Q_{T+1} = T'Q_T + TQ'_T$$