

Diagrama Karnaugh

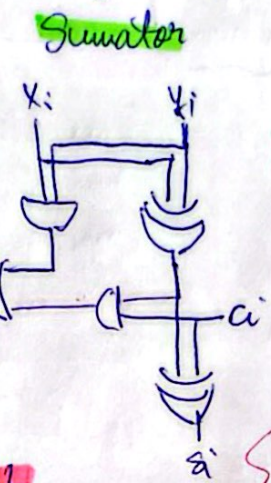
ab \ cd	00	01	11	10
00	0	1	3	2
01	4	5	7	6
11	12	13	15	14
10	8	9	11	10

$X + X' = 1$
 $X \cdot X' = 0$ } Complementul
 $X + X = X$
 $X \cdot X = X$ } Idempotenta
 $y \cdot X + X = X$
 $(y + X) \cdot X = X$ } Absorbția

$(x + y)' = x' \cdot y'$
 $(x \cdot y)' = x' + y'$ } De Morgan
 $x + 1 = 1$
 $x \cdot 0 = 0$
 $x \cdot (x + y) = x \cdot y$
 $x + x \cdot y = x + y$ } Absorbția inversă

Zecimal	Octal
1 → 7	1 → 7
8 → 15	+ 2

Inversor $x \rightarrow x'$ 1
AND $x, y \rightarrow x \cdot y$ 2, 4
NAND $x, y \rightarrow (x \cdot y)'$ 1, 4
OR $x, y \rightarrow x + y$ 2, 4
NOR $x, y \rightarrow (x + y)'$ 1, 4
XOR $x, y \rightarrow x \oplus y$ 4, 2
XNOR $x, y \rightarrow x \odot y$ 3, 2



SM	C1	C2
9 bits	S20 bits	S20 bits
	S20 bits	S20 bits

Vîrgulă flotantă
 $nr = \pm m.b^e$
 normalizăm
 $exp + 127$

BCD
 $45_{10} = 0100\ 0101_{2cd}$

IEEE

s	exp	mantință
0	16	8b 8b 23b

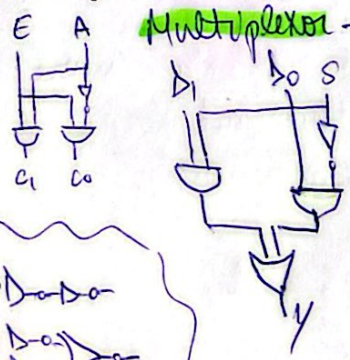
Minterm / Maxterm

$f = \sum (3, 5, 6, 7) \quad m_i = 1$
 $f = \prod (0, 1, 2, 4) \quad M_i = 0$
 $n \text{ variabile} \Rightarrow 2^{2^n} \text{ fct.}$

$B\bar{C} + \bar{B} \cdot C = B \oplus C$
 $BC + \bar{B}\bar{C} = B \odot C$

implicant prim - la o grupare pe diagramă nu mai poți grupa un minterm
esențial - dacă un implicant prim conține un minterm care nu apare în altul \Rightarrow esențial.

Decodificator $\rightarrow n$ intrări cu 2^n ieșiri, activează doar o ieșire

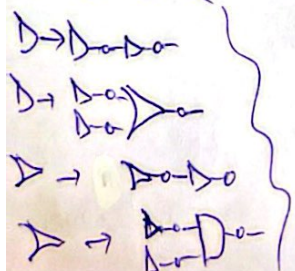


Codificator \rightarrow fct. inversă decodificatorului
 $\rightarrow 2^n$ intrări, 1 ieșire



ROM - utilizate doar pt. citirea informației

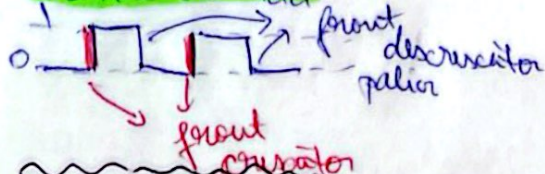
PROM - ca ROM dar pot fi programate de utilizator



Sequentials

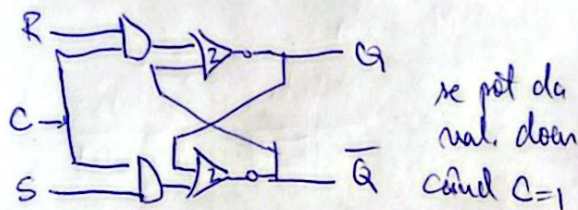
sincron: modifică starea și val. de ieșire în fun. de modificările variabilelor de la intrare, când se modifică aceste.
asincron: modifică val. în fun. de variabile de intrare la momente bine def. de timp dictate de clock.

Signal de tact



$T = \frac{1}{f}$ - frecvență periodică
 $\text{delay} = N \cdot T = \frac{N}{f}$ nr. cicluri

Gated S-R Latch



Tipuri FF

SR	$Q(\text{next}) = S + R' \cdot Q$
0 0	Q
0 1	0
1 0	1
1 1	No

JK	$Q(\text{next}) = J \cdot Q' + K' \cdot Q$
0 0	Q
0 1	0
1 0	1
1 1	Q'

D	$Q(\text{next}) = D$
0	0
1	1

T	$Q(\text{next}) = T \cdot Q' + T' \cdot Q$
0	Q'
1	Q

Registru - grupare de n biți stabile; nr. max. de val. - 2ⁿ.

Load	State-next	Shift	State	State-next
0	data in	0	Q ₃ Q ₂ Q ₁ Q ₀	Q ₃ Q ₂ Q ₁ Q ₀
1	data in	1	Q ₃ Q ₂ Q ₁ Q ₀	data Q ₃ Q ₂ Q ₁

(PIPO) (SIPO)

Numărător - balanșă o succesiune de stări impuse de proiectant

Serie & atare	Non volatil / ROM
Acces random	EPROM
SRAM	EEPROM
DRAM	FLASH

Wire - leagă componente

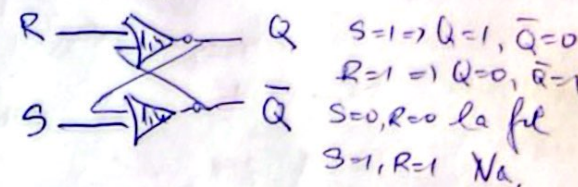
assign

reg - păstrează o val.

always

sincron porridge clk
asincron - or porridge out

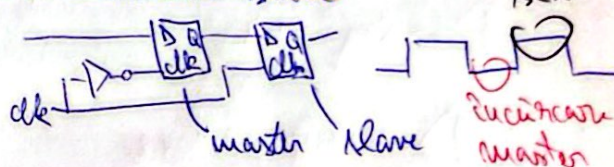
S-R Latch



FF as latch

bascularea pe FF se face pe front (nu pe paliu ca la latch)

FF - master slave



MEALY - starea urm. și ieșirea la un moment dat depind de starea prezente și intrarea prezente

MOORE

ieșirea depinde doar de starea circuitului
starea urm. depinde doar de intrarea prezente

1 kbit = 1024 biți
nr. linii adresă

$\log_2(\text{nr. de loc. de mem.})$

nr. biți adresă

Cap = 2^m - m

biți fiecare adresă

nr. total loc = 2ⁿ

SRAM - mem. starea câtă vreme e alimentat

DRAM - are nev. de alimentare + refresh
are densitate mare

= atribuire ind.

(=) atribuire la finalul ciclului de ceas