FIZILEAN ALEXANDRU
LH 615468

$X+X'=1$, $X\cdot X'=0$ } complementul

$X+X=X$, $X\cdot X=X$ } idempotența

$y\cdot x+x=x$, $(y+x)\cdot x=x$ } absorbția

$x+1=x$, $x\cdot 0=0$ } absorbția inversă

$x(\bar{x}+y)=xy$, $x+\bar{x}\cdot y=x+y$

$(x+y)'=x'\cdot y'$, $(x\cdot y)'=x'+y'$ } De Morgan

$B\bar{C}+\bar{B}C=B\oplus C$

$B\cdot C+\bar{B}\bar{C}=\overline{B\oplus C}$

Porți	Simbol	T
inversor	x — D — x'	1
AND	x, y — D — $x\cdot y$	2,4
NAND	x, y — D — $(x\cdot y)'$	1,4
OR	x, y — ⊃ — $x+y$	2,4
NOR	x, y — ⊃ — $(x+y)'$	1,4
XOR	x, y — ⊃ — $x\oplus y$	4,2
XNOR	x, y — ⊃ — $x\odot y$	3,2

	B_2	B_8	B_{16}
0	0000	0	0
1	0001	1	1
2	0010	2	2
3	0011	3	3
4	0100	4	4
5	0101	5	5
6	0110	6	6
7	0111	7	7
8	1000	10	8
9	1001	11	9
10	1010	12	A
11	1010	13	B
12	1100	14	C
13	1101	15	D
14	1110	16	E
15	1111	17	F

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	y	⊕	⊙
0	0	0	1
0	1	1	0
1	0	1	0
1	1	0	1

$f=\sum(3,5,6,7) \Rightarrow m_i=1$ - mintermi

$f=\pi(1,2,4,10) \Rightarrow M_i=0$ - maxtermi

n variabile $\Rightarrow 2^{2^n}$ fct.

implicant prim - la o grupare pe diagramă nu mai poți grupa un minterm

— u — esențial - dacă un implicant prim conține un minterm care nu apare în altul $\Rightarrow$ esențial

ROM - utilizate doar pt. citirea informației; nevolatilă (nu pierde info.)

PROM - ca ROM dar poate fi programată de utilizator

COMBINAȚIONAL

SM { C_1: S>0 → biți; S<0 → biți } { C_2: S>0 biți; S<0 C_{1+1} } BCΔ

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

Virgulă flotantă

$nr. = \pm m \cdot b^e$

normalizăm

$e + 127$

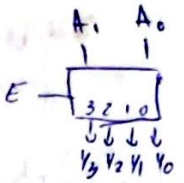
IEEE | S | e | mantisa

0 | 1 8 9 | 31

1b 8b 23b

DECODIFICATOR → n intrări cu 2^n ieșiri, activează doar o ieșire

2-la-4

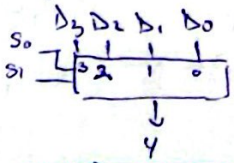


exemplu: $E=1$ $A_1=1$ $A_0=0 \Rightarrow Y_2=1$

$E=1$ $A_1=1$ $A_0=1 \Rightarrow Y_3=1$

MULTIPLEXOR → conectează ieșirea la una din cele n intrări

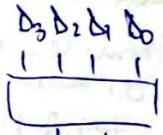
4-la-1



exemplu: $S_0=0$, $S_1=1 \Rightarrow 2_{10}$ ($sel=10$) $\Rightarrow Y=D_2$

COMUTICATOR → invers decodificator → 2^n intrări, n ieșiri

4-la-2



exemplu. $D_0=1 \Rightarrow Y_1=Y_0=0$

$D_3=1 \Rightarrow Y_1=Y_0=1$

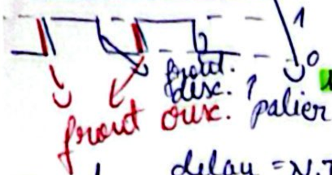
$D_2=1 \Rightarrow Y_1=1$, $Y_0=0$.

FIZĂDEAN ALEXANDRU
LM05468

$2^0 = 1$, $2^1 = 2$, $2^2 = 4$ | $2^3 = 8$, $2^4 = 16$, $2^5 = 32$ | $2^6 = 64$, $2^7 = 128$, $2^8 = 256$ | $2^9 = 512$, $2^{10} = 1024$, $2^{11} = 2048$ | $2^{12} = 4096$

SECVENȚIALE → **asincron**: modifică starea și valoarea de ieșire în funcție de modificările semnalelor de la intrare, oricând se modifică acestea

Semnal de tact

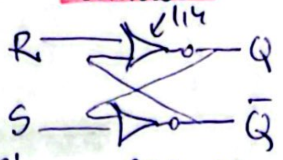


sincron: modifică val. în funcț. de semnalele de intrare la momente bine definite de timp dictate de clk.

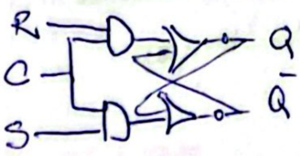
$T = \frac{1}{f}$ (T - perioadă, f - frec.) delay $= N \cdot T = \frac{N}{f}$ (N - nr. cicluri)

Latchuri

S-R Latch:



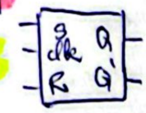
Gated S-R Latch



se pot da val. doar când C=1.

Tipuri FF

SR

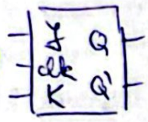


$Q(next) = S + R' \cdot Q$

S	R	Q(next)
0	0	Q
0	1	0
1	0	1
1	1	NA

FF vs. Latchuri → bascularea pe FF se face pe front (nu pe palier ca pe latch).

JK



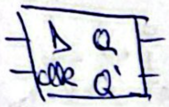
$Q(next) = J \cdot Q' + K' \cdot Q$

J	K	Q(next)
0	0	Q
0	1	0
1	0	1
1	1	Q'

MEALY → starea urm. și ieșirea la un moment dat depind de starea prezentă și intrarea prezentă.

MOORE → ieșirea depinde doar de starea circuitului
↳ starea urm. depinde doar de intrarea prezentă.

D

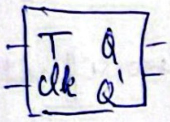


$Q(next) = D$

D	Q(next)
0	0
1	1

Scriere & Citire		Non volatile	ROM
Acces random	non random	EPROM	măști programate
SRAM	FIFO	EEPROM	
DRAM	LIFO	FLASH	

T



$Q(next) = T Q' + T' Q$

T	Q(next)
0	Q
1	Q'

1Kbit = 1024biți

nr. linii adresă = $\log_2$(nr. loc. de mem.).

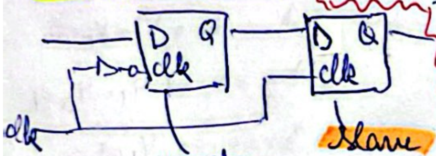
nr. total loc 2^n

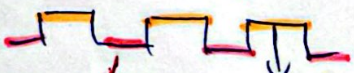
cap $= 2^n \cdot m$ → (n - nr. biți adresă; m - biți fiecare adresă)

SRAM – mem. starea câtă vreme e alimentat

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

Master - Slave





= - atribuie imediat

<= - atribuie la finalul ciclului de tact

wire – leagă componente
└ assign

reg – păstrează o val.
└ always

REGISTRU - grupare de n bistabile; nr. max. de val. 2^n.

PIPO L=1 D=1010 => Q=1010 instant

SIPO Q=0000 D=1 => Q=0001; D=0 => Q=0010; D=1 => Q=0101; D=1 => Q=1011.

PISO L=1 Q=1101; L=0, S=1 => 1,1,0,1. - iese Q_3 la fiecare shift.

SISO input=1 Q=0000 => Q=1000 output=0

NUMĂRĂTOARE → succesiune de stări binare impuse de proiectant.