

CURS 0x02

Cantitatea de informație

$$I(x_i) = \log_2 \left| \frac{1}{p_i} \right|$$

↳ exemplu $I(\text{rege}) = \log_2 \left| \frac{1}{p_{\text{rege}}} \right| = \log_2 \left(\frac{1}{\frac{4}{52}} \right) \leftarrow 4 \text{ regi din } 52 \text{ de cărți}$

$$= \log_2 \left(\frac{52}{4} \right) = 3.7 \text{ biți de informație}$$

$$P(\text{eveniment } A) = \frac{\text{nr. de situații în care se întâmplă } A}{\text{număr total situații}}$$

Eveniment A și eveniment B = 1 ca ambele să se întâmple | $P(\text{eveniment } A) \cdot P(\text{eveniment } B)$

• N evenimente la fel de probabile, M sunt posibile din cele N

N = 52 (cărți)

M = 13 (am ales că am extras o inimă roșie) dar nu știm ce carte e

$$I(\text{carte}) = \log_2 \frac{1}{13 \cdot \frac{1}{52}} = \log_2 \left| \frac{52}{13} \right|$$

↳ even. A = o carte roșie

Dacă M = 1 → max de inf știm exact ce carte e

Dacă N = M → nu avem nicio inf $\log_2 1 = 0$, incertitudinea este 0.

Dacă M > N → incertitudinea este mai mare

• aruncăm un ban (cap / pajură)

N = nr. cazuri posibile = 2

M = știm că vom vedea și cap și pajură = 1

$$I = \log_2 \frac{1}{\frac{1}{2}} = 1 \text{ bit inf.}$$

• o monedă cu cap pe ambele părți

N = 1

M = 1

$$I = \log_2 \frac{1}{1} = 0 \text{ biți inf.}$$

• aruncăm 2 zaruri

N = 36 pos.

M = vedem doar 1

$$I = \log_2 \frac{1}{\frac{1}{36}} = \log_2 36 = 5.17 \text{ biți inf.}$$

ENTROPIA = limita de compresie posibilă

$$\sum_{i=1}^N p_i \cdot \log_2 \frac{1}{p_i}$$

$X = \{A, B, C, D\}$

$$\frac{1}{3} \quad \frac{1}{2} \quad \frac{1}{12}$$

$$1 - \frac{1}{4} + \frac{1}{6} + \frac{1}{12} = 1 - \frac{11}{12} = \frac{1}{12}$$

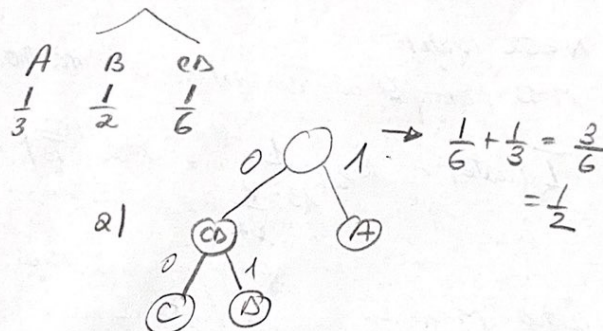
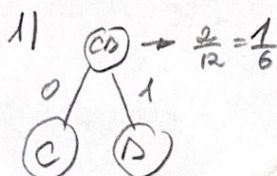
$$\text{entropia} = \frac{1}{3} \cdot \log_2 \frac{1}{3} + \frac{1}{2} \cdot \log_2 \frac{1}{2} + \frac{1}{12} \cdot \log_2 \frac{1}{12} = 1.626 \text{ biti}$$

teoretic X poate fi pus pe 2 biti 13 eveni : 00, 01, 10, 11 dar putem mai bine 1.626 biti
! codarea trebuie să fie unică

ALG. LUI HUFFMAN

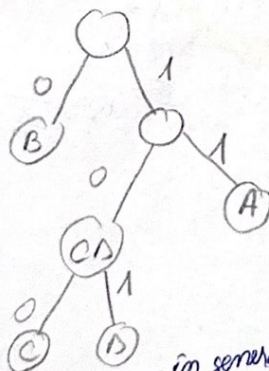
(pentru a obține codarea eficientă și unică)
evenimentele mai dese primesc o codare mai scurtă

$$\frac{A}{\frac{1}{3}}, \frac{B}{\frac{1}{2}}, \frac{C}{\frac{1}{12}}, \frac{D}{\frac{1}{12}}$$



$$\frac{A \text{ CD}}{\frac{1}{2}}, \frac{B}{\frac{1}{2}}$$

3)



$A = 11$

$B = 0$

$C = 100$

$D = 101$

$ABBC = 1100100$

7 biti

eficiența codării : 2.85%
m. de biti $\rightarrow 2 \cdot \frac{1}{3} + 1 \cdot \frac{1}{2} + 3 \cdot \frac{1}{12} + 3 \cdot \frac{1}{12} = 1.667$
 $00100111010011 = BB C A D B B A$

- în general 2E+1, detectează E erori

DISTANȚA HAMMING

- pt. a putea detecta erori

$0 \rightarrow 000$

$1 \rightarrow 111$

NOTIŢE CURS 0X03

- de ce folosim semnale digitale în loc de analogice

- 1) din cauza zgomotului,
- 2) într-un sist. analogic zgomotul se acumulează
- 3) -"- digital avem colecțiile de zgomot

- de ce folosim sistemul binar? și fi mai avantajos sistemul hex

- 1) hexul ar fi de 4 și mai avantajos
- 2) în loc de 2 stări ar trebui să avem 16 adică 16 nivele de voltagi + măsuri de zgomot

A	B	C	X	Y
0	0	0	0	1
0	0	1	0	0
0	1	0	0	0
0	1	1	0	1
1	0	0	0	0
1	0	1	0	1
1	1	0	0	1
1	1	1	0	0

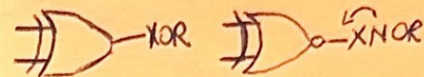
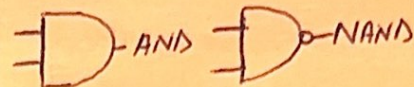
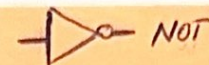
$$X = \bar{A}\bar{B}C + \bar{A}B\bar{C} + A\bar{B}\bar{C} + ABC$$

$$Y = \bar{A}\bar{B}\bar{C} + \bar{A}BC + A\bar{B}C + AB\bar{C}$$

$$X = \bar{A}(\bar{B}C + B\bar{C}) + A(\bar{B}\bar{C} + BC)$$

$$X = \bar{A}(B \oplus C) + A(\overline{B \oplus C})$$

$$X = A \oplus B \oplus C$$



A	B	A ⊕ B
0	0	0
0	1	1
1	0	1
1	1	0

$A \oplus B = \bar{A}B + A\bar{B}$
 Sau $\overline{A \oplus B} = \bar{A}\bar{B} + AB$

NOTIŢE CURS 0X04

A	B	C	X
0	0	0	0
1	0	0	1
0	1	0	0
1	1	0	1
0	0	1	0
1	0	1	0
0	1	1	1
1	1	1	1

$$X = A\bar{B}\bar{C} + A\bar{B}C + A\bar{B}C + ABC$$

$$X = A\bar{C}(\bar{B} + B) + A(\bar{B} + B)C$$

$$= A\bar{C} + AC$$

circuit de adunare

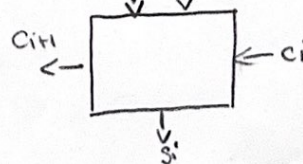
avem a și b pe N biți

$$S = a + b$$

S poate fi pe maxim N+1 biți

număr intrări 2N
număr ieșiri NH

cât de mare va fi? $N+1$ sau 2^N-1

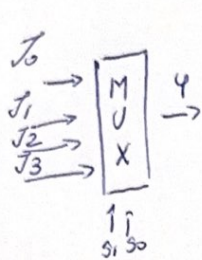


$$S_i = a_i \oplus b_i \oplus c_i$$

$$\begin{aligned}
 0 \oplus 0 &= 0 \oplus 1 = 1 \\
 1 \oplus 1 &= 0 \oplus 0 = 0 \\
 1 \oplus 0 &= 1 \oplus 0 = 1 \\
 0 \oplus 1 &= 1 \oplus 1 = 0
 \end{aligned}$$

locul = 1

multiplexare



→ folositor pt. if și case implementare hardware pt. shift

C. COMB

- 1) nu putem itera
- 2) nu permit "log. int." sau "mem. internă"
- 3) nu există o stare internă a circuitului
- 4) prea simple
- 5) nu toate lucrurile pot fi imp. cu log. comb.

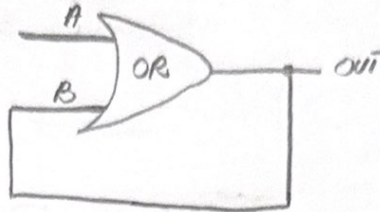
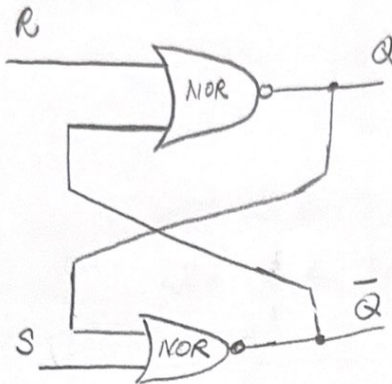
C. SEQU.

- 1) avem stare internă (mem. int.)
- 2) avem var. de timp (CLK)
- 3) intrările / ieșirile nu sunt fixe
- 4) nr. var. de psi : ex
un circ. care adună 2
nr. de dim. oarecare

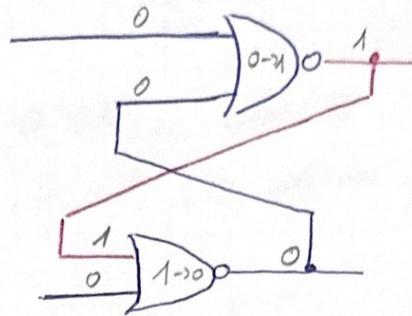
SR LATCH

↳ set
reset

↳ becomes fixed in a particular state
nu se poate reseta (daca puni deconectarea sursei de voltagi)

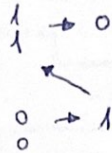
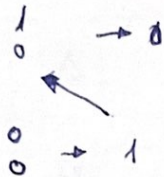


A	B	out
0	0	0
0	1	1
1	0	1
1	1	1



A	B	NOR
0	0	1
0	1	0
1	0	0
1	1	0

↳ pt. 1 și 0 sau 0 → 0 → 0 latch



= latch

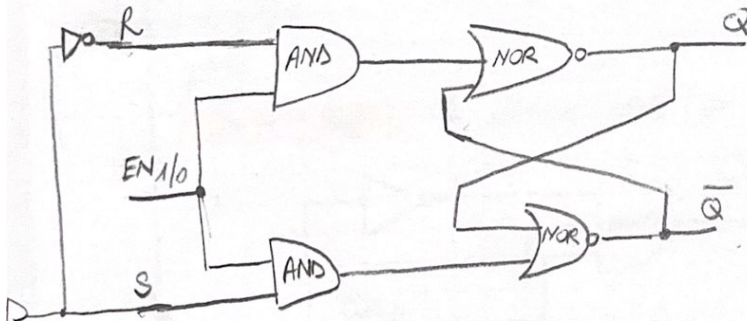
! Stocarează un sg. bit de inf.

Sau:

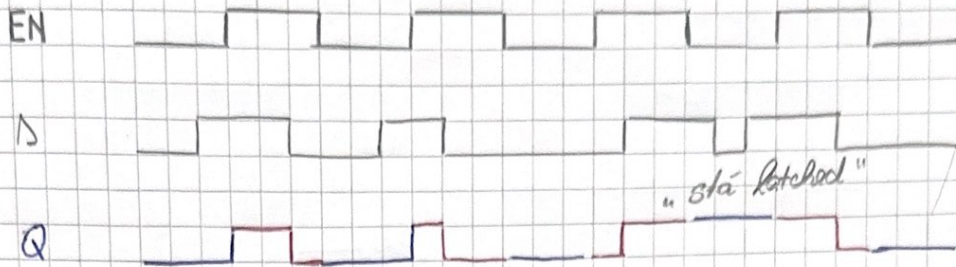


D LATCH

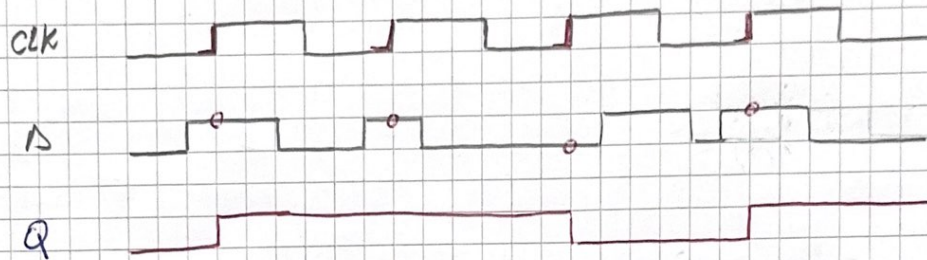
"SR LATCH with ENABLE"



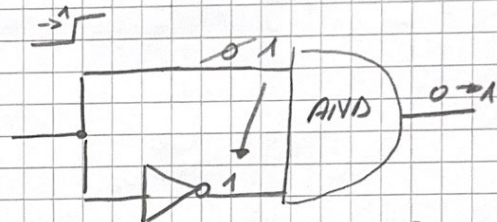
dacă D e "high" 11: 5V-10V ⇒ Q alt. cu \bar{Q}
dacă D e 0 ⇒ nu se modifică și apare latch-ul



when the EN is high, Q will be what the D is
 when the EN is low, Q will stay whatever it is

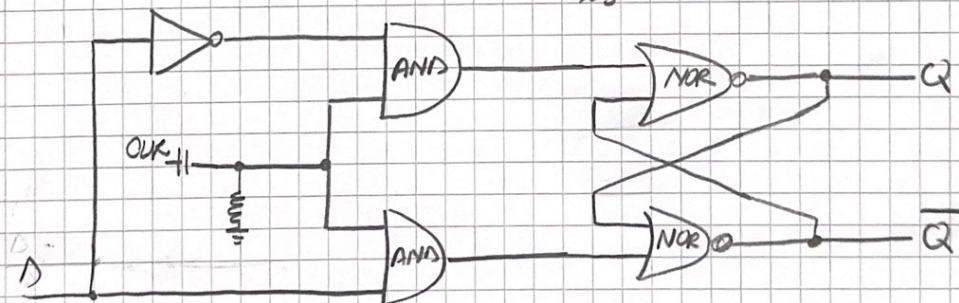


pt. a detecta schimbările de clk

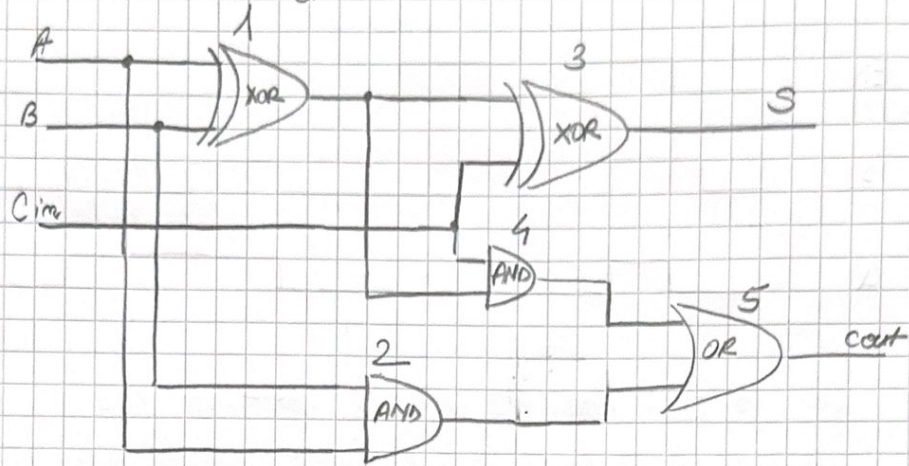


↳ 1 va rămâne pt. câtă va rămasec. până când poarta NOT schimbă 1-ul în 0

D FLIP-FLOP → storing data in a digital logic circuit



CIRCUIT DE ADUNARE



$C: 1010$
 0111
 0101
 1000

$C_{in}=0$
 $C_{in}=1$

I

1: $A: 1$
 $B: 1$
 $\text{XOR} \rightarrow 0$

 2: $A: 1$
 $B: 1$
 $\text{AND} \rightarrow 1$

3: 0
 0
 $\text{XOR} \rightarrow 0$

 4: 0
 0
 $\text{AND} \rightarrow 0$

5: 0
 1
 $\text{OR} \rightarrow 1$
 bit carry

1: $A: 1$
 $B: 0$
 $\text{XOR} \rightarrow 1$

 2: $A: 1$
 $B: 0$
 $\text{AND} \rightarrow 0$

3: 1
 1
 $\text{XOR} \rightarrow 0$

 4: 1
 0
 $\text{AND} \rightarrow 0$

5: 0
 0
 $\text{OR} \rightarrow 0$
 Cout = 0