

# Logică digitală

-Curs 14-  
SINTEZĂ  
- 2024-

# Reprezentarea numerelor în sistemele de calcul

---

- ❑ Sisteme de numerație poziționale (binar, octal, hexazecimal);
  - ❑ Reprezentarea numerelor în virgulă fixă (SM, C1, C2);
  - ❑ Reprezentarea numerelor de virgulă flotantă;
  - ❑ Coduri binare pentru numere zecimale;
-

# Reprezentarea numerelor în virgulă fixă – C2

$$\begin{array}{r}
 + \quad 2 \quad = \quad 0 \quad 0 \quad 1 \quad 0 \quad + \\
 + \quad 3 \quad = \quad 0 \quad 0 \quad 1 \quad 1 \\
 \hline
 \phantom{+} \phantom{2} \phantom{=} \phantom{0} \phantom{0} \phantom{1} \phantom{0} \phantom{+} \phantom{0} \phantom{1} \phantom{0} \phantom{1} \phantom{=} \phantom{+} \phantom{5}
 \end{array}$$

**a.**

$$\begin{array}{r}
 - \quad 2 \quad = \quad 1 \quad 1 \quad 1 \quad 0 \quad + \\
 - \quad 3 \quad = \quad 1 \quad 1 \quad 0 \quad 1 \\
 \hline
 \phantom{-} \phantom{2} \phantom{=} \phantom{1} \phantom{1} \phantom{1} \phantom{0} \phantom{+} \phantom{1} \phantom{1} \phantom{0} \phantom{1} \phantom{=} \phantom{-} \phantom{5}
 \end{array}$$

**b.**

$$\begin{array}{r}
 - \quad 2 \quad = \quad 1 \quad 1 \quad 1 \quad 0 \quad + \\
 + \quad 3 \quad = \quad 0 \quad 0 \quad 1 \quad 1 \\
 \hline
 \phantom{-} \phantom{2} \phantom{=} \phantom{1} \phantom{1} \phantom{1} \phantom{0} \phantom{+} \phantom{0} \phantom{0} \phantom{0} \phantom{1} \phantom{=} \phantom{+} \phantom{1}
 \end{array}$$

**c.**

$$\begin{array}{r}
 + \quad 2 \quad = \quad 0 \quad 0 \quad 1 \quad 0 \quad + \\
 - \quad 3 \quad = \quad 1 \quad 1 \quad 0 \quad 1 \\
 \hline
 \phantom{+} \phantom{2} \phantom{=} \phantom{0} \phantom{0} \phantom{1} \phantom{0} \phantom{+} \phantom{1} \phantom{1} \phantom{1} \phantom{1} \phantom{=} \phantom{-} \phantom{1}
 \end{array}$$

**d.**

- Domeniul valoric pentru numere întregi:  
 $-2^{n-1}$  și  $2^{n-1} - 1$

# Reprezentarea numerelor în virgulă flotantă (mobilă)

---

- reprezentate folosind notația științifică (care nu este pozițională) → un domeniu valoric foarte mare.
- Pentru a reprezenta un număr în virgulă flotantă folosim trei numere conform relației:

$$N = M * B^E$$

$M$  - mantisa. ( $M$  poate fi reprezentată în SM sau C2)

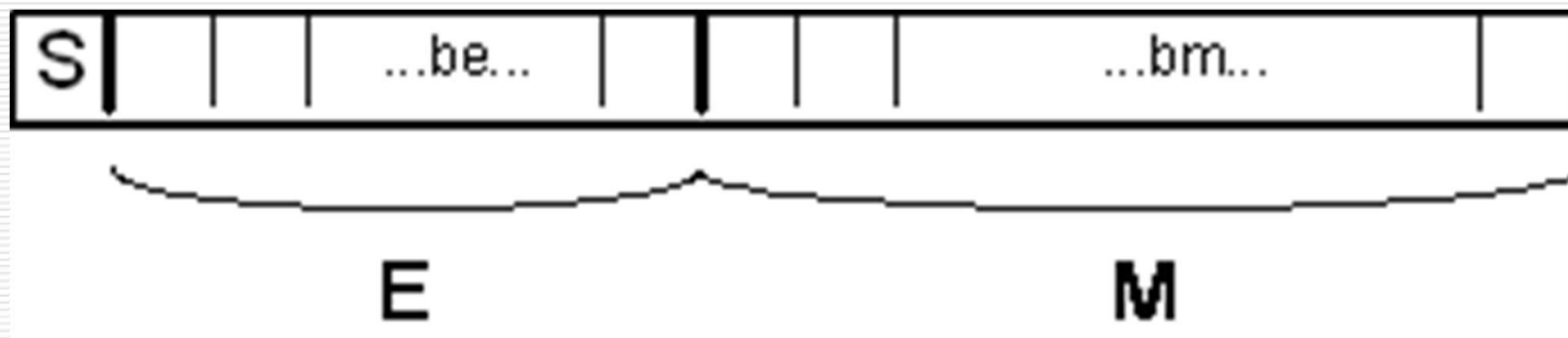
$B$  - baza (de obicei e 2 sau o putere a lui 2)

$E$  - exponent. ( $E$  este reprezentat în SM sau cod exces)

---

# Reprezentarea numerelor în virgulă flotantă

---



*M* - mantisa. (*M* poate fi reprezentată în SM sau C2)

*B* - baza (de obicei e 2 sau o putere a lui 2)

*E* - exponent. (*E* este reprezentat în SM sau cod exces)

---

# Reprezentarea numerelor în virgulă flotantă

---

- ❑ Pentru a minimiza eroarea → exponentul aferent lui 0 să fie cel mai mic posibil.
  - ❑ valoarea min. a oricărui exponent să fie 0.
  - ❑ Toate valorile negative reprezentabile pe N biți sunt deplasate (devin pozitive) prin adunarea unui bias (unui surplus) = valoarea absolută a celui mai mic număr reprezentabil pe N numărul de biți exponent.
  - ❑ Pentru exponent reprezentat în:
    - SM pe 8 biți → valoarea bias-ului este 127
    - C2 pe 8 biți → valoarea bias-ului este 128
-

# Reprezentarea numerelor în virgulă flotantă

---

- Standardul IEEE 754/2008- formate:
    - Half precision
    - Simple precision
    - Double precision
    - Double-extended
-

# Reprezentarea numerelor în virgulă flotantă: IEEE 754

---

## ☐ Caracteristici:

- E și M – format SM
  - Exponentul este reprezentat în exces de:
    - ☐ 127 pentru simplă precizie
    - ☐ 1023 pentru dublă precizie.
  - *Hidden bit.*
    - ☐ Mantisa are un bit de 1 ascuns.
    - ☐ bitul la dreapta virgulei care trebuie să fie 1 (din condiția de normalizare).
    - ☐ (S.**1**M) (unde S este semnul iar M este mantisa)  
→ virgula a fost mutată la dreapta bit-ului de 1  
cel mai semnificativ: S**1**.M
-



# Reprezentarea numerelor în virgulă flotantă: IEEE 754 valori speciale

Nr.	Exponent (E)	Mantisa (M)	Valoare speciala
1.	0	0	$\pm 0$
2.	0	$\neq 0$	Denormalized numbers
3.	255	0	$\pm \infty$
4.	255	$\neq 0$	NaN

- Nr. denormalizate: rezultat care este mai mic decât valoarea minimă reprezentabilă
- Infinit: situația în care rezultatul intermediar este infinit sau avem overflow
- 0/0 sau radical din nr. negativ

# Coduri binare pentru numere zecimale - BCD

---

- ❑ există situații când se dorește afișarea rezultatelor de către interfețele externe ale dispozitivului de calcul într-un format ușor de înțeles (decodificat) de către utilizator – și anume mult întrebuițatul format zecimal;
  - ❑ cel mai la îndemână cod zecimal este BCD (binary-code decimal):
    - reprezentarea unei cifre BCD → înlocuirea cu reprezentarea în binar care îi corespunde → cu un nr. pe 4 biți
-

# Coduri binare pentru numere zecimale - BCD

- conversia unui număr zecimal în BCD prin înlocuirea succesivă a cifrelor zecimale cu tetradele corespunzătoare

$$N_1 = 4\ 5\ 9_{10} = 0100\ 0101\ 1001_{BCD}$$

- operația inversă de transformare a unui număr reprezentat în BCD în omologul zecimal

$$N_2 = 1000\ 0111\ 0000\ 0010_{BCD} = 8\ 7\ 0\ 2_{10}$$

# Coduri binare pentru numere zecimale – Exces de 3

---

- Exces de 3 se obține din codul BCD astfel:
    - la fiecare cifră reprezentată în cod BCD se adună valoarea 3 (0011 în binar).
    - fiecare cifră zecimală se reprezintă cu ajutorul unei combinații de 4 biți (o tetradă de biți)
-

# Algebra booleană și logica digitală

---

- Forma canonică;
  - Forma standard;
  - Aspecte legate de implementarea funcțiilor booleene cu porților logice;
  - Hărți Karnaugh
-

NAND



X	Y	Z
0	0	1
0	1	1
1	0	1
1	1	0

$$Z = \overline{X \cdot Y}$$

AND



X	Y	Z
0	0	0
0	1	0
1	0	0
1	1	1

$$Z = X \cdot Y$$

NOR



X	Y	Z
0	0	1
0	1	0
1	0	0
1	1	0

$$Z = \overline{X + Y}$$

OR



X	Y	Z
0	0	0
0	1	1
1	0	1
1	1	1

$$Z = X + Y$$

# Porți logice (cont.)

**XOR**  
 $(X \oplus Y)$



X	Y	Z
0	0	0
0	1	1
1	0	1
1	1	0

$Z = X \bar{Y} + \bar{X} Y$   
X or Y but not both  
("inequality", "difference")

**XNOR**  
 $\overline{(X \oplus Y)}$



X	Y	Z
0	0	1
0	1	0
1	0	0
1	1	1

$Z = \bar{X} \bar{Y} + X Y$   
X and Y the same  
("equality")

# Porți logice

---

- ❑ Fiecare poartă logică realizează una sau mai multe funcții logice;
  - ❑ Colecția de porți logice folosită în realizarea unui circuit se numește **bibliotecă de porți**, iar porțile din cadrul ei **porți standard**;
  - ❑ Bibliotecile moderne conțin zeci de porți a.î. să scadă costul cu întreținerea și să simplifice realizarea tool-urilor CAD
-



# Minimizarea funcțiilor logice

---

□ se înțelege simplificarea/rescrierea ecuațiilor logice booleene în vederea:

■ Unui cost mai mic și/sau;

■ Performanță mai ridicată;

□ Cheia simplificării este:  $y(x + \bar{x}) = y$

■ distributivitatea -  $x(y+z) = xy+xz$  —

■ Proprietatea complementului  $x + \bar{x} = 1$

---

# Minimizarea funcțiilor logice

- Găsirea a doi termeni (suma sau produs funcție de reprezentarea dorită SOP/POS) pentru care:
  - funcția ia valoare 1
  - numai o variabilă își modifică valoarea

A	B	F
0	0	1
0	1	0
1	0	1
1	1	0

B are aceeași  
valoare → B este  
păstrat

A are valori  
diferite → A este  
eliminat

$$F = \bar{A}\bar{B} + A\bar{B} = (\bar{A} + A)\bar{B} = \bar{B}$$

# Construcție diagrame Karnaugh

- construcția diagramei Karnaugh pentru o funcție logică cu 4 variabile de intrare

cd \ ab		c			
		00	01	11	10
a	00	0000	0001	0011	0010
	01	0100	0101	0111	0110
	11	1100	1101	1111	1110
	10	1000	1001	1011	1010
		d			

# Pași:

---









- ❑ 1. Introducerea mintermilor în diagramă conform tabelului de adevăr.
  - ❑ 2. se formează grupe de mintermi bazate pe reguli de adiacență:
    - O grupare are forma unor dreptunghiuri/pătrate și conține  $2^n$  mintermi!
    - Din totalul de  $m$  variabile booleene a funcției, termenul asociat grupării formate va conține  $m-n$  variabile
  - ❑ 3. Ecuația minimizată va conține toți implicantii primi esențiali, si uneori si implicantii primi neesențiali, astfel încât toate celule marcate cu 1 logic să fie acoperite.
-

# Minimizarea folosind diagrame Karnaugh

---

- Dacă la o astfel de grupare nu mai pot fi adăugați mintermi înseamnă că s-a obținut un **implicant prim**.
  - Dacă un anumit implicant prim conține cel puțin un minterm care nu poate apare în alt implicant prim atunci acesta este un **implicant prim esențial**
-

# PORȚI LOGICE CU MAI MULTE ÎNTRĂRI

Name	Graphic Symbol	Functional Expression	Number of transistors	Delay in <i>ns</i>
3-input AND		$F = xyz$	8	2.8
4-input AND		$F = xyzw$	10	3.2
3-input OR		$F = x + y + z$	8	2.8
4-input OR		$F = x + y + z + w$	10	3.2
3-input NAND		$F = (xyz)'$	6	1.8
4-input NAND		$F = (xyzw)'$	8	2.2
3-input NOR		$F = (x + y + z)'$	6	1.8
4-input NOR		$F = (x + y + z + w)'$	8	2.2

# Circuite logice combinaționale

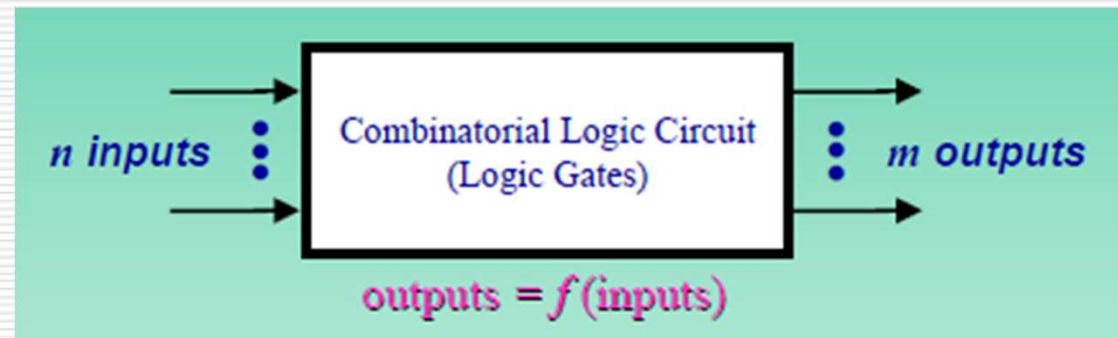
---

- ☐ Circuite de procesare
  - ☐ Circuite de conversie
  - ☐ Circuite de interconectare
  - ☐ Componente universale
-

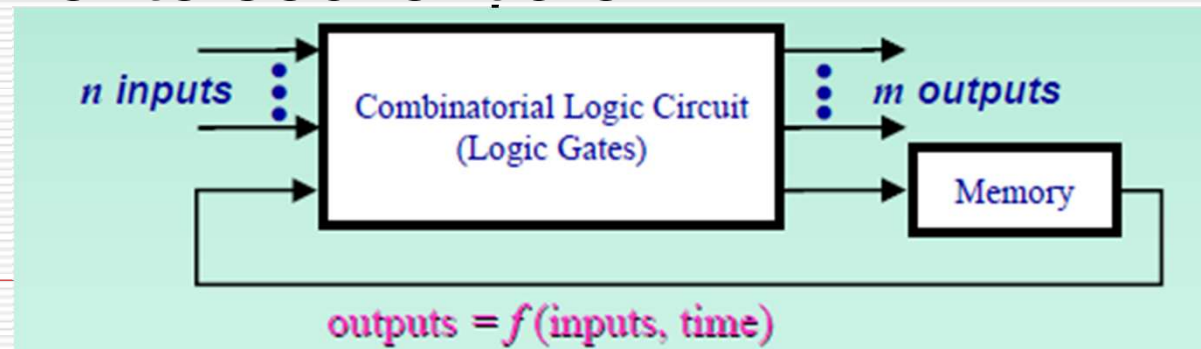
# Clasificare componente digitale

## □ Componente combinaționale

- Ușor de analizat, partiționat, verificat



## □ Componente secvențiale





# Circuite reprezentative

---

- ☐ Sumator
- ☐ Decodificator
- ☐ Codificator (ex. Codificator de prioritate)
- ☐ Multiplexor

Caracterizarea acestora: definiție, ce se întâmplă dacă aplic o serie de intrări, care ar fi ieșirile, simbol.

---

# Circuite combinaționale

---

- ☐ Cunoașterea porților logice elementare;
  - ☐ Desenarea unei scheme logice;
  - ☐ Identificarea ecuației booleene aferente unei scheme logice;
  - ☐ Determinarea ieșirilor unei scheme logice la care i s-au aplicat o serie de intrări;
-

# Decodificator

---

- circuite logice combinaționale ce prezintă un anumit  $n$  intrări și până la  $2^n$  ieșiri, care activează **ieșirea (UNA SINGURĂ)** corespunzătoare valorii combinației vectorului de intrare
  - Pot avea intrări de activare, astfel încât ieșirea selectată nu pot fi activată decât dacă intrările de activare sunt active.
  - Pt.  $n$  intrări și cu  $m$  ieșiri → decodificator  $n$ -la- $m$ .
  - Uzual sunt folosite pt. activarea (EN) componentelor
-

# Multiplexor(Selector)

---

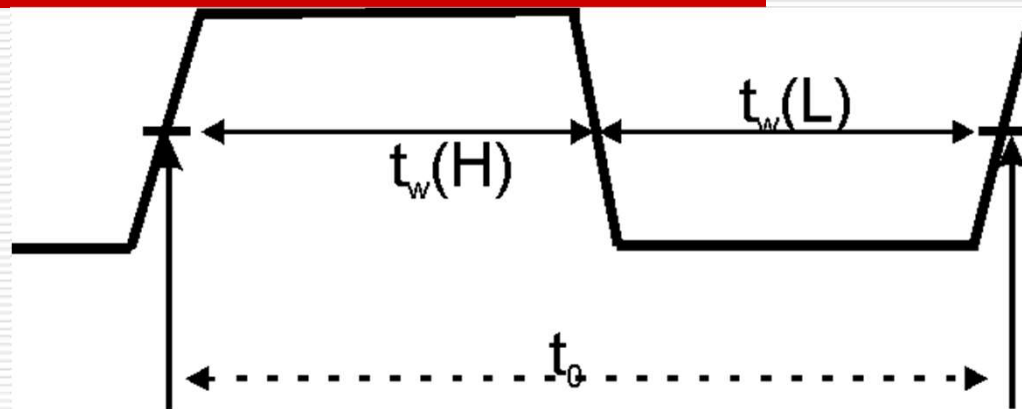
- ❑ Multiplexorul este un circuit logic combinațional ce conectează ieșirea acestuia la una din cele  $n$  intrări.
  - ❑ Selecția uneia din cele  $n$  intrări se face cu ajutorul a  $\log_2 n$  intrări de selecție.
  - ❑ Poate fi privit ca un comutator digital.
  - ❑ Este folosit pt. selecția unei singure surse de date din mai multe.
-

# Circuite secvențiale

---

- **Circuitele secvențiale se clasifică:**
    - **Asincrone**
    - **Sincrone**
  
  - Componentele secvențiale asincrone își modifică starea și valorile de ieșire funcție de modificările semnalelor de la intrare (**oricând!**) se modifică acestea.
  
  - Componentele secvențiale sincrone își modifică valoarea funcție de valoarea semnalelor de intrare la **momente bine definite de timp**, dictate de un semnal (de intrare) care se numește tact (*clock*)
-

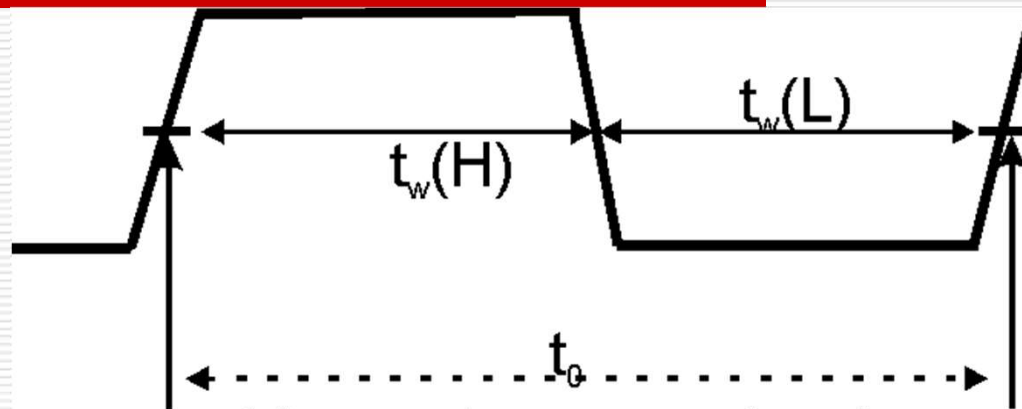
# Semnalul de tact



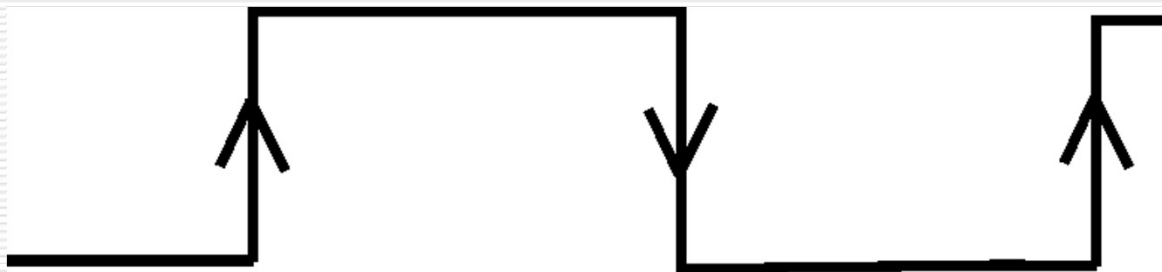
- ❑ **Palierul** unui semnal reprezintă porțiunea unde acesta rămâne constant 0 logic (palier negativ) și 1 logic (palier pozitiv).
- ❑ **Frontul crescător** se referă la porțiunea unde semnalul își modifică valoarea de la 0 logic la 1 logic (mai exact de la 10% din nivelul corespunzător lui 1 logic la 90% din nivelul corespunzător lui 1 logic)
- ❑  $t_0$  - perioada semnalului de tact,
- ❑  $t_w(H)$  și  $t_w(L)$  reprezintă durata unui impuls de 1 respectiv 0 logic

# Semnalul de tact

---



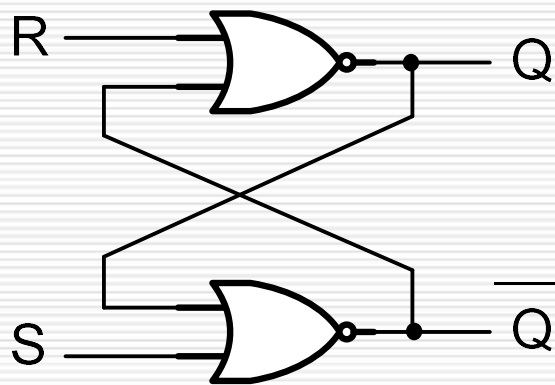
a) Semnal rectangular de tact



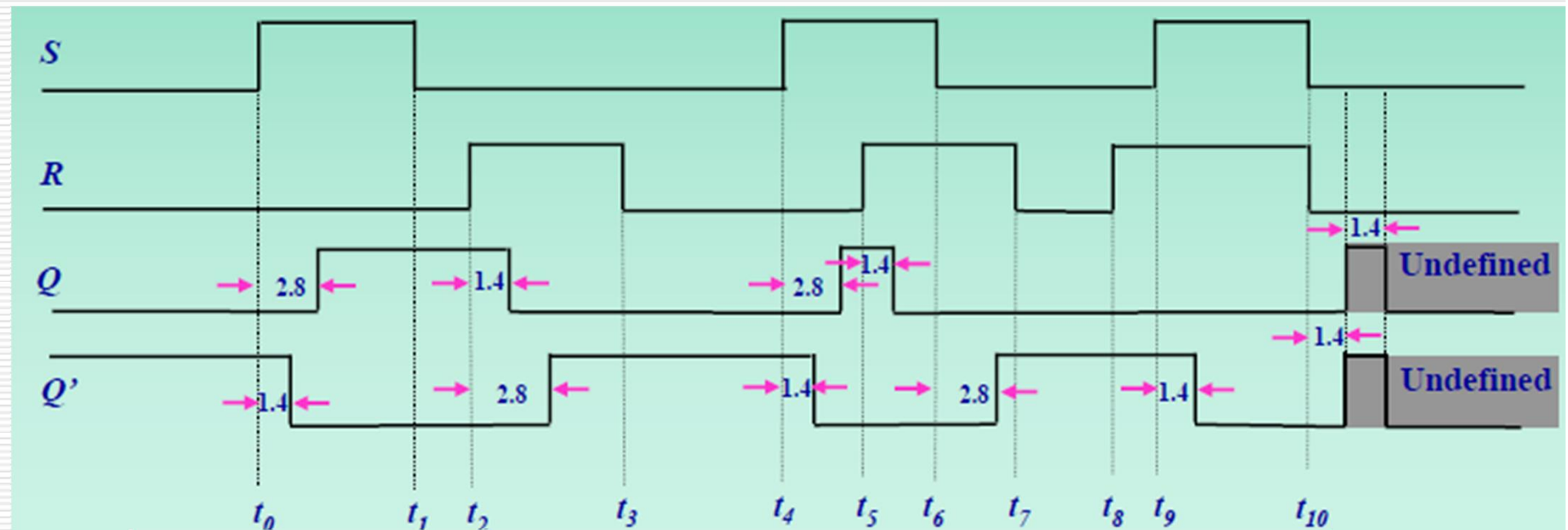
b) Formă idealizată a semnalului

---

# S-R Latch (SAU-NU) - asincron

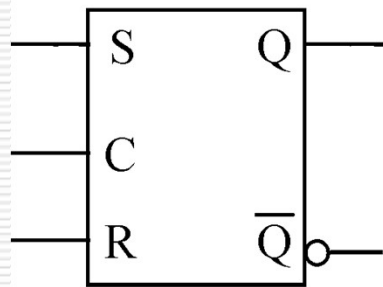


R	S	$Q_{t+1}$
0	0	$Q_t$
0	1	1
1	0	0
1	1	interzis





# Gated SR-latch



S	R	C	Q <sub>next</sub>	$\overline{Q}_{next}$
0	0	1	Q	$\overline{Q}$
0	1	1	0	1
1	0	1	1	0
1	1	1	-	-
*	*	0	Q	$\overline{Q}$

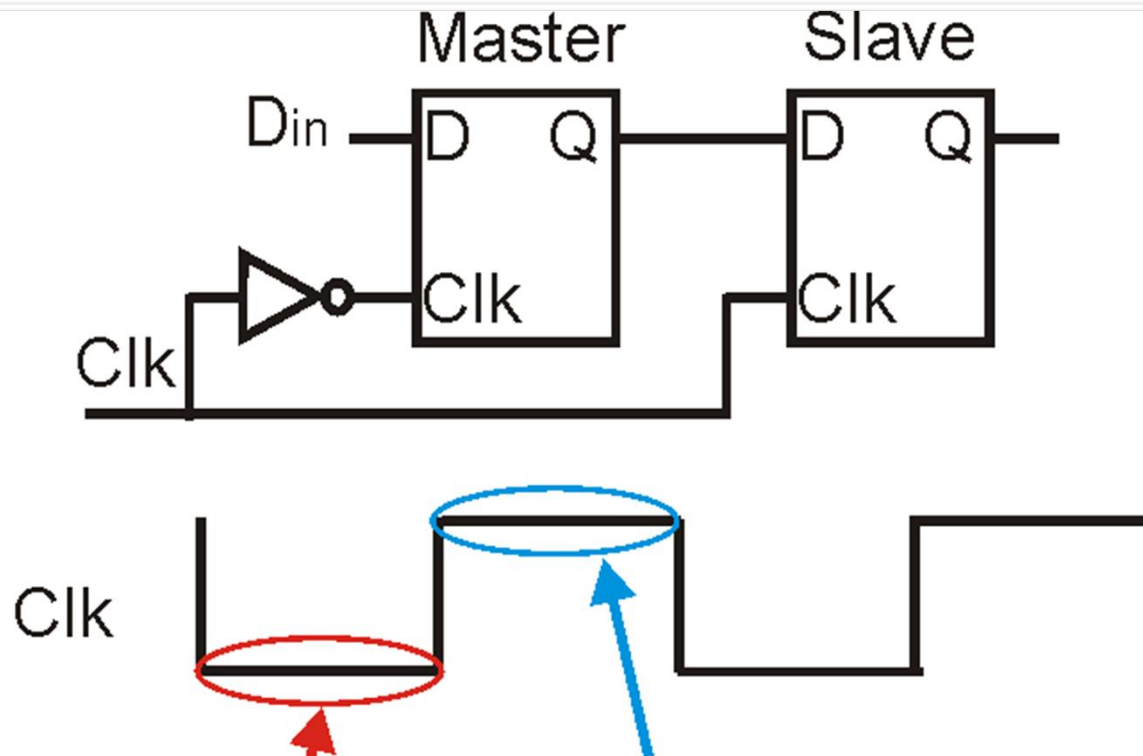
- ❑ Când semnalul C este activ valorile de la intrare sunt propagate prin latch
- ❑ Semnalele de intrare nu trebuie să se modifice în intervalul  $t_{setup}$  și  $t_{hold}$  al frontului descrescător

# Flip-flop-uri

---

- ❑ Se mai numesc și latch-uri sensibile pe frontul semnalului de tact;
  - ❑ Bascularea se face pe frontul semnalului de tact (!nu pe palier – latch-uri)
  - ❑ Două variante de arhitecturi:
    - Configurația master-slave
    - Edge-triggered FF
-

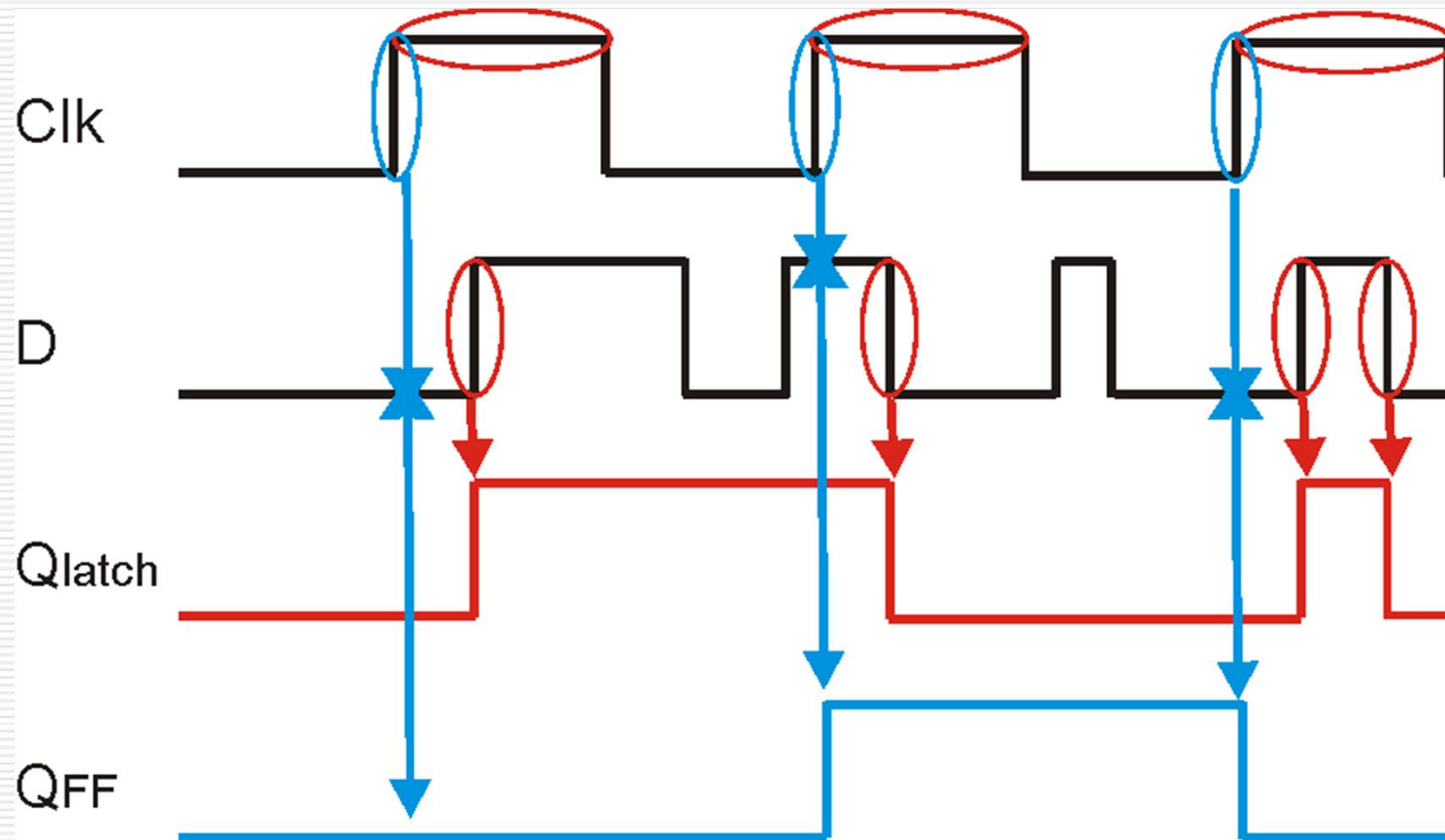
# FF-Master-Slave



Incarcare Master  
Latch cu  $D_{in}$   
Slave Latch  
memoreaza  
data anteriora

Incarcare Slave  
Latch cu  $Q_{master}$   
Master Latch  
memoreaza  
valoarea  $D_{in}$   
dinaintea comutarii  
semnalului Clk

# Latch sincron vs. FF sincron



# Operatia de reset a elementelor secventiale

---

## ☐ Semnalul de reset (set)

- Functionalitate – aducerea bistabilului intr-o stare “initiala” cunoscuta (de obicei starea 0)
- Reset este un semnal global – este aplicat tuturor elementelor de memorie dintr-un sistem digital
- Tipuri de reset
  - ☐ Reset sincron
  - ☐ Reset asincron

# Tipuri de FF-uri

Flip-flop name	Flip-flop symbol	Characteristic table	Characteristic equation	Excitation table																																			
SR		<table> <tr> <th>S</th> <th>R</th> <th>Q(next)</th> </tr> <tr><td>0</td><td>0</td><td>0</td></tr> <tr><td>0</td><td>1</td><td>0</td></tr> <tr><td>1</td><td>0</td><td>1</td></tr> <tr><td>1</td><td>1</td><td>NA</td></tr> </table>	S	R	Q(next)	0	0	0	0	1	0	1	0	1	1	1	NA	$Q(next)=S+R'Q$ $SR=0$	<table> <tr> <th>Q</th> <th>Q(next)</th> <th>S</th> <th>R</th> </tr> <tr><td>0</td><td>0</td><td>0</td><td>X</td></tr> <tr><td>0</td><td>1</td><td>1</td><td>0</td></tr> <tr><td>1</td><td>0</td><td>0</td><td>1</td></tr> <tr><td>1</td><td>1</td><td>X</td><td>0</td></tr> </table>	Q	Q(next)	S	R	0	0	0	X	0	1	1	0	1	0	0	1	1	1	X	0
S	R	Q(next)																																					
0	0	0																																					
0	1	0																																					
1	0	1																																					
1	1	NA																																					
Q	Q(next)	S	R																																				
0	0	0	X																																				
0	1	1	0																																				
1	0	0	1																																				
1	1	X	0																																				
JK		<table> <tr> <th>J</th> <th>K</th> <th>Q(next)</th> </tr> <tr><td>0</td><td>0</td><td>0</td></tr> <tr><td>0</td><td>1</td><td>0</td></tr> <tr><td>1</td><td>0</td><td>1</td></tr> <tr><td>1</td><td>1</td><td>Q'</td></tr> </table>	J	K	Q(next)	0	0	0	0	1	0	1	0	1	1	1	Q'	$Q(next)=JQ'+K'Q$	<table> <tr> <th>Q</th> <th>Q(next)</th> <th>J</th> <th>K</th> </tr> <tr><td>0</td><td>0</td><td>0</td><td>X</td></tr> <tr><td>0</td><td>1</td><td>1</td><td>X</td></tr> <tr><td>1</td><td>0</td><td>X</td><td>1</td></tr> <tr><td>1</td><td>1</td><td>X</td><td>0</td></tr> </table>	Q	Q(next)	J	K	0	0	0	X	0	1	1	X	1	0	X	1	1	1	X	0
J	K	Q(next)																																					
0	0	0																																					
0	1	0																																					
1	0	1																																					
1	1	Q'																																					
Q	Q(next)	J	K																																				
0	0	0	X																																				
0	1	1	X																																				
1	0	X	1																																				
1	1	X	0																																				
D		<table> <tr> <th>D</th> <th>Q(next)</th> </tr> <tr><td>0</td><td>0</td></tr> <tr><td>1</td><td>1</td></tr> </table>	D	Q(next)	0	0	1	1	$Q(next)=D$	<table> <tr> <th>Q</th> <th>Q(next)</th> <th>D</th> </tr> <tr><td>0</td><td>0</td><td>0</td></tr> <tr><td>0</td><td>1</td><td>1</td></tr> <tr><td>1</td><td>0</td><td>0</td></tr> <tr><td>1</td><td>1</td><td>1</td></tr> </table>	Q	Q(next)	D	0	0	0	0	1	1	1	0	0	1	1	1														
D	Q(next)																																						
0	0																																						
1	1																																						
Q	Q(next)	D																																					
0	0	0																																					
0	1	1																																					
1	0	0																																					
1	1	1																																					
T		<table> <tr> <th>T</th> <th>Q(next)</th> </tr> <tr><td>0</td><td>Q</td></tr> <tr><td>1</td><td>Q'</td></tr> </table>	T	Q(next)	0	Q	1	Q'	$Q(next)=TQ'+T'Q$	<table> <tr> <th>Q</th> <th>Q(next)</th> <th>T</th> </tr> <tr><td>0</td><td>0</td><td>0</td></tr> <tr><td>0</td><td>1</td><td>1</td></tr> <tr><td>1</td><td>0</td><td>1</td></tr> <tr><td>1</td><td>1</td><td>0</td></tr> </table>	Q	Q(next)	T	0	0	0	0	1	1	1	0	1	1	1	0														
T	Q(next)																																						
0	Q																																						
1	Q'																																						
Q	Q(next)	T																																					
0	0	0																																					
0	1	1																																					
1	0	1																																					
1	1	0																																					

# Circuite secvențiale reprezentare

---

## □ **Circuitele secvențiale:**

- **MEALY** sunt caracterizate prin faptul că starea următoare și ieșirea la un moment dat depind de starea ***prezentă*** și de ***intrarea prezentă***;
- **MOORE** sunt caracterizate prin faptul că ieșirea depinde **numai** de **starea circuitului**. Starea următoare depinde de intrarea prezentă;

- Modelele matematice ale circuitelor secvențiale se numesc în teoria comutațiilor **automate finite**.
-

# Registre

---

- ❑ Reprezinta o colectie/grupare de  $n$  bistabile
  - ❑ Nr maxim de valori a unui registru pe  $n$  biti –  $2^n$  valori binare
  - ❑ Folosit pentru memorarea unui cuvant de date/unei stari curente a sistemului
-

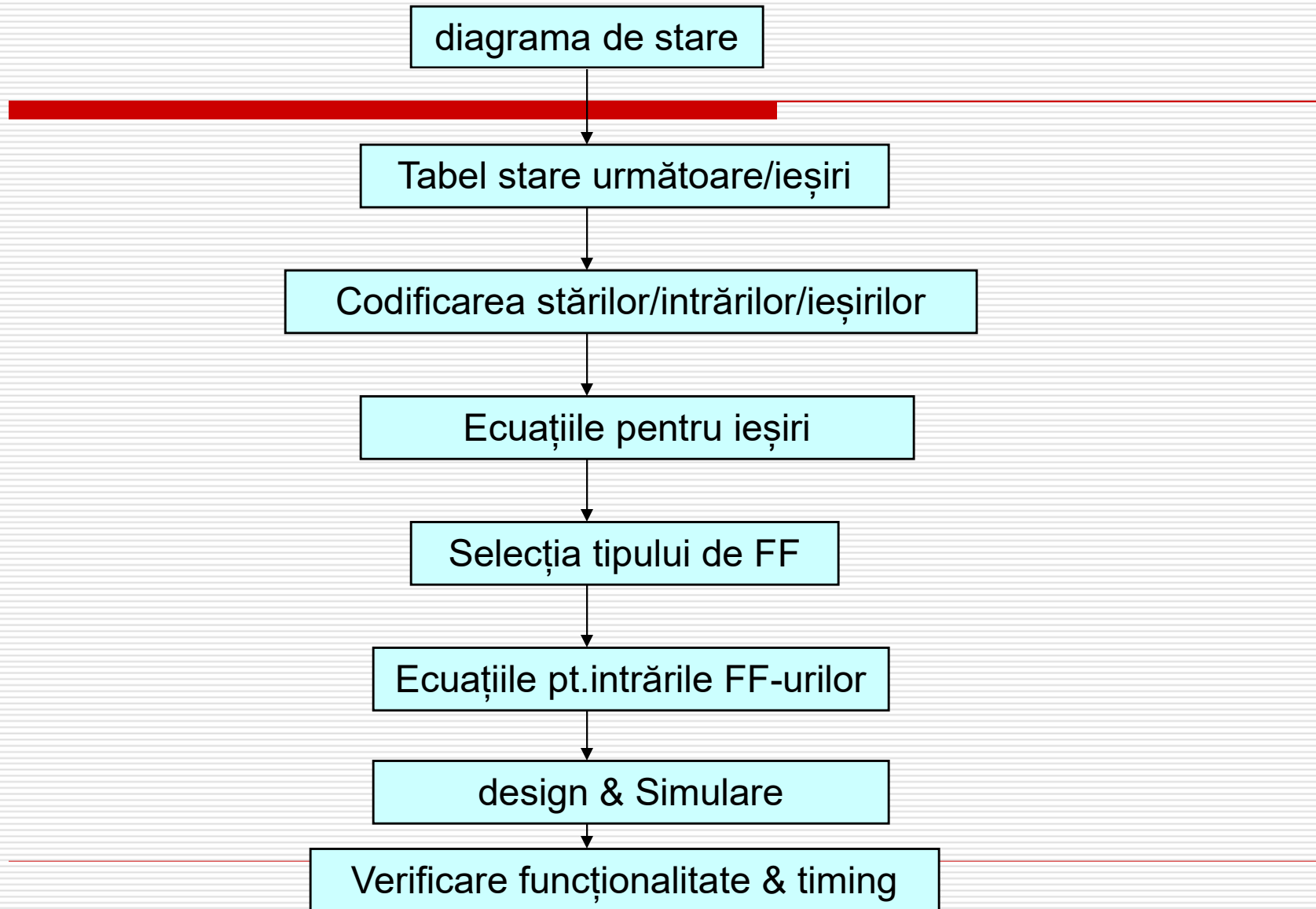


# Numărătoare

---

- ❑ circuite **secvențiale sincrone autonome** (mulțimea intrărilor vidă), care baleiază o secvență de stări impuse de proiectant.
  - ❑ de regulă este inițializat cu starea „0” , după care la fiecare impuls de numărare, comuta într-o nouă stare.
  - ❑ caracterul **asincron** al unui numărător este dat de faptul că impulsul de tact nu comandă simultan toate bistabilele numărătorului.
  - ❑ Funcție de direcția de parcurgere a secvenței de stări:
    - numărător în sens crescător,
    - numărător în sens descrescător,
    - numărător reversibil (ambele sensuri).
-

# Etape de sinteză circuit secvențial



# Circuite secvențiale

---

- ❑ Citirea unei diagrame de stări și realizarea tabelului tranzițiilor;
  - ❑ Citirea unei diagrame de simulare:
    - Identificarea tipului de RST
    - Extragerea caracteristicilor unui circuit simplu (secvența de stări prin care trece circuitul pe baza intrărilor și stării curente din diagrama de timp)
  - ❑ Clasificarea memoriilor, număr biți de adrese vs. capacitate, DRAM vs. SRAM
-

# Verilog HDL

---

- ❑ Diferența dintre *reg* și *wire*;
- ❑ Aspecte legate de modelarea corectă a unui block combinațional;
- ❑ Aspecte legate de modelarea corectă a unui block secvențial;
- ❑ Diferența dintre cei 2 operatori de atribuire  $=$ ,  $=>$
- ❑ Identificarea pe baza codului a tipului de RST; a tipului de bloc modelat; ...

# Laborator

---

## ☐ 3 note:

- Test 1
- Test 2
- Nota3\_Activitate: Nota Activitate (assignment-uri notate, răspunsuri în timpul laboratorului, simulare prezentată în timpul laboratorului)

## ☐ VERIFICAȚI pe CV!

## ☐ Nota LAB: media aritmetica a celor 3 note

---

# Examen

---

- ☐ QUIZ (2 pag pe aceasi foaie: 1 comb & 1 secv)
  - Poate conține întrebări de tip eseu (max 1 p)
  - 1 foaie scrisa de mana cu nume si prenume cu majuscule (1 pag comb., 1 pag secventiale)
- ☐ Problema (2p –open book – hard copy)
- ☐ Date examinare:
  - Prezentarea 1
    - ☐ 05 iunie (ora 14-18) -seria C – A101
    - ☐ 07 iunie (ora 8-12) -seria D – D1
  - Prezentarea 2 - sala D1
    - ☐ 27 iunie, ora 17 - ambele serii
  - Prezentarea 3 - sala D1
    - ☐ (tentative) - 28 August (to be confirmed)

# Întrebări?

---

“Pregătirea și calificarea nu trebuie să se limiteze doar la însușirea de tehnici, proceduri sau cunoștințe.

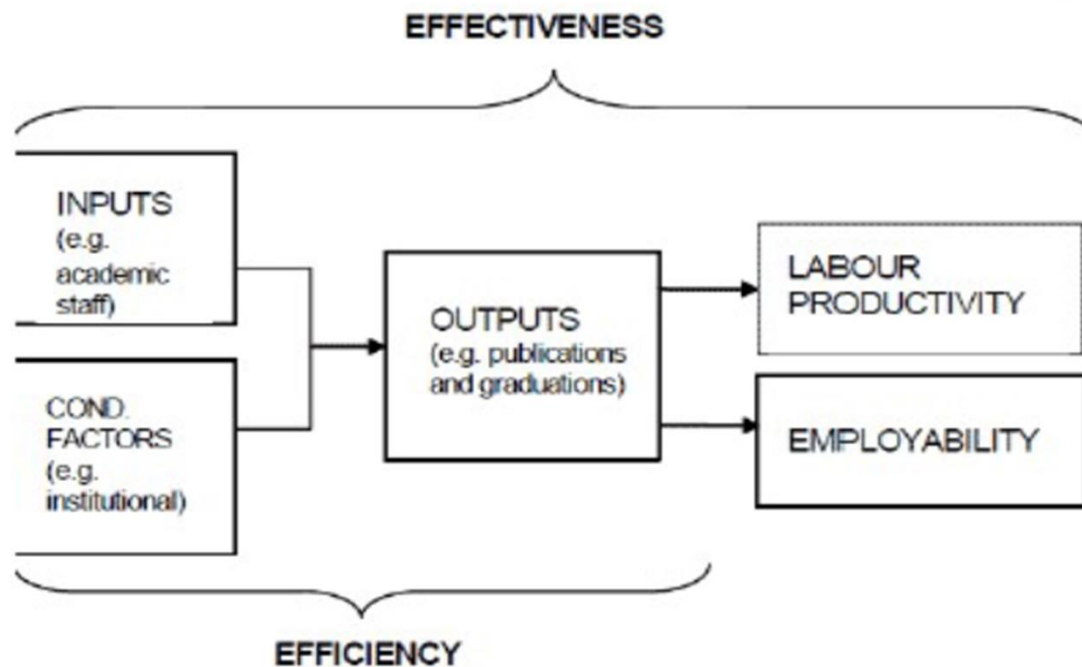
Însușirea de valori,  
formarea de atitudini,  
modelarea personalității,  
stimularea gândirii critice,  
a responsabilității,  
a abilității de a lucra în echipă,  
a gândirii creative,  
a stimulării liberei inițiative

sunt cele mai importante unelte de lucru,  
uneori mult mai importante decât cele asociate cunoștințelor teoretice.”

---

# Întrebări?

*Ilustrarea definiției Comisiei Europene cu privire la eficiență și*

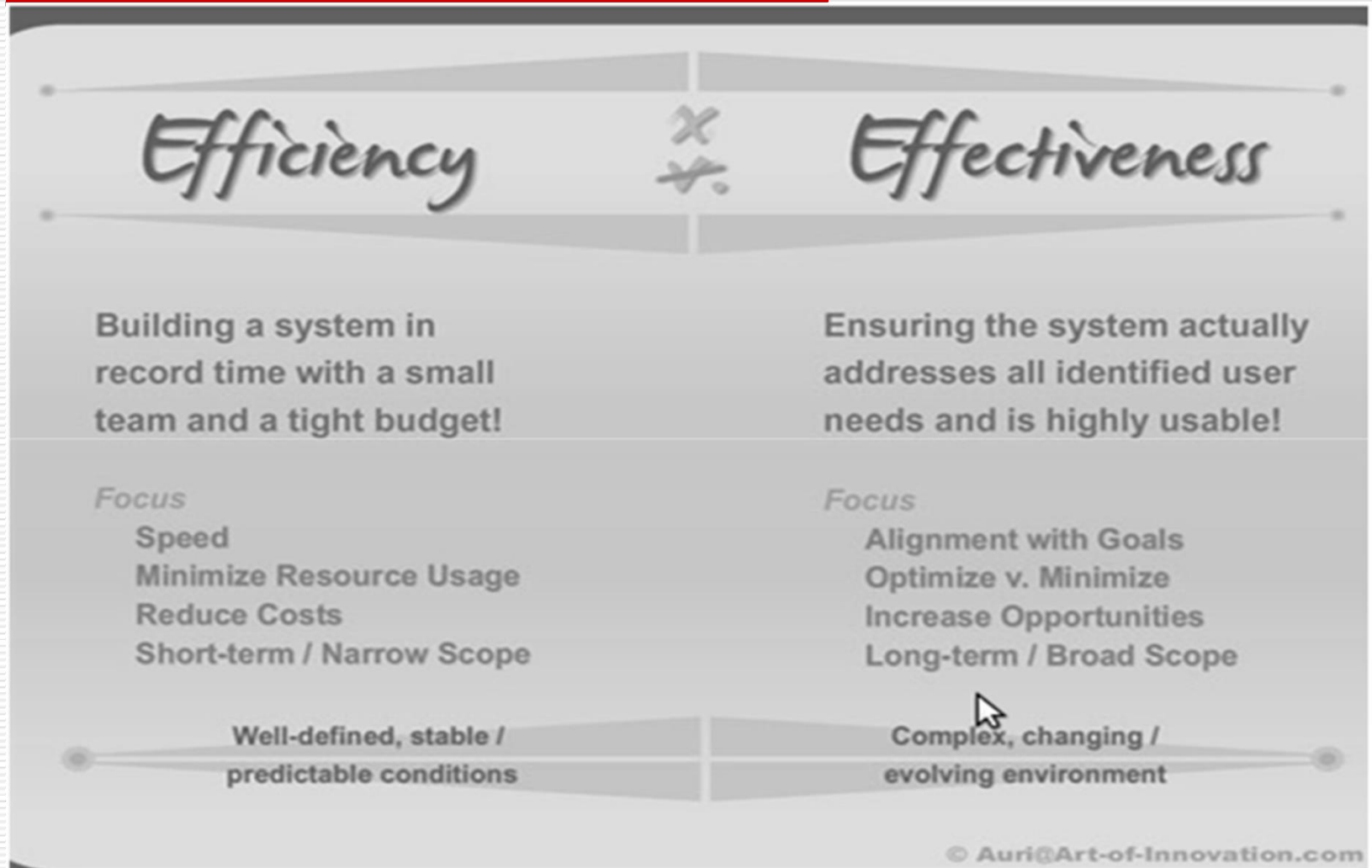


*eficacitate*

Sursa: [www.gov.scot](http://www.gov.scot)



# Întrebări?



Sursa: [www.art-of-innovation.com](http://www.art-of-innovation.com)

---

**Enough Talking Let's Get To It  
!!Brace Yourselves!!**

