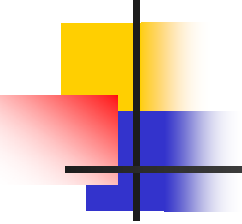




Circuite Integrate Digitale

Conf. Monica Dascălu

- 
-
- Curs
 - Seminar
 - Laborator – notă separată
 - Notare:
 - 40% seminar
 - 20% teme // + TEMA SUPLIMENTARA
 - 40% examen



Bibliografie

- Note de curs
- Cursul profesorului G. Stefan:
<http://arh.pub.ro/gstefan/0-BOOK.pdf>

Bibliografie suplimentară

- Barry Wilkinson – Electronica Digitală – bazele proiectării, Ed. Teora, 2002
- John Wakerly – Circuite digitale. Principiile și practicile folosite în proiectare, Ed. Teora, 2002
- M. Dascalu, D. Stefan, D.Davidescu, Z. Hascsi, B. Mitu. Probleme de circuite integrate digitale Ed. Printech, Bucuresti, 1998.



Cursul 1

- Analog vs. Digital
- Definirea formală a unui sistem digital
- Exemple
- Introducerea limbajului Verilog HDL
- Aritmetica binară



Analog vs. Digital

- Analog

- Orice formă de undă

- Digital

- Numai semnale dreptunghiulare
- În orice punct al circuitului, toate semnalele au o semnificație



Analog vs. Digital

- Transmisia datelor
- Detectia/corectia erorilor
- Memorare
- Procesare

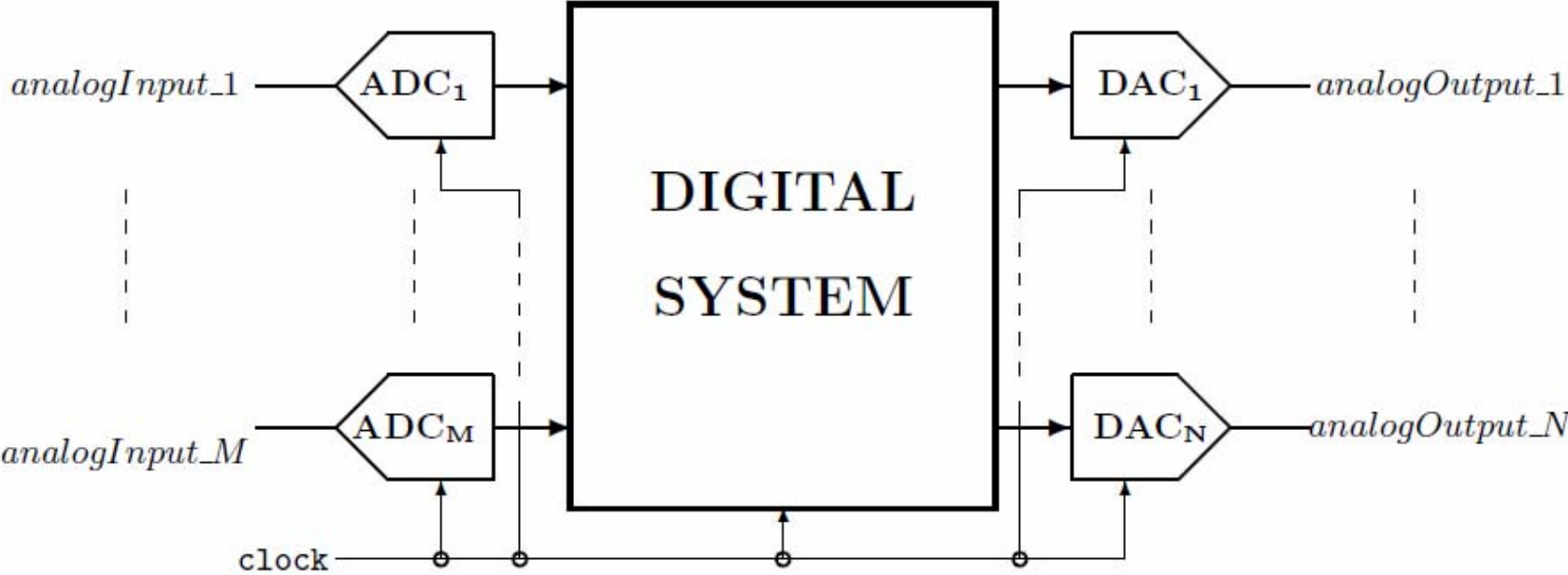


Ce înseamnă "0" și "1"?

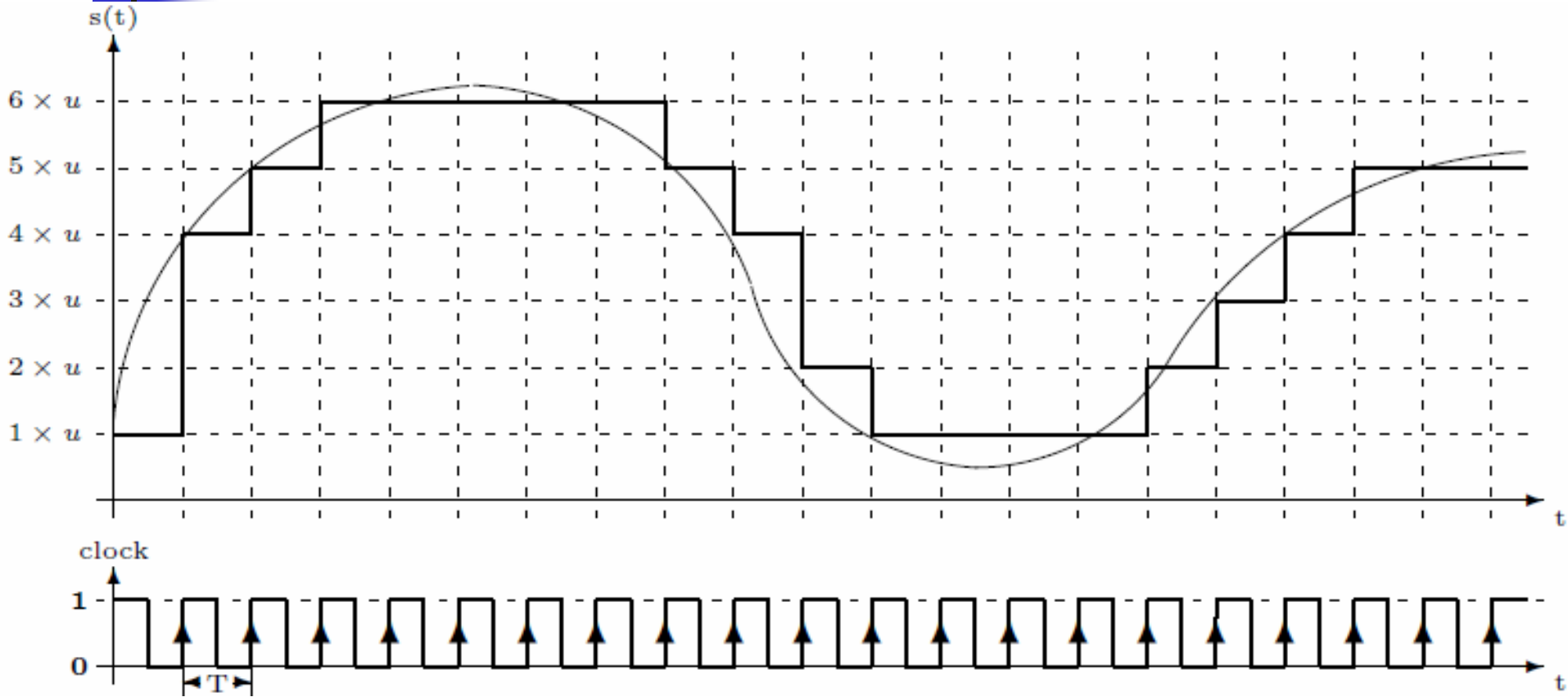
Trei tipuri de semnificații:

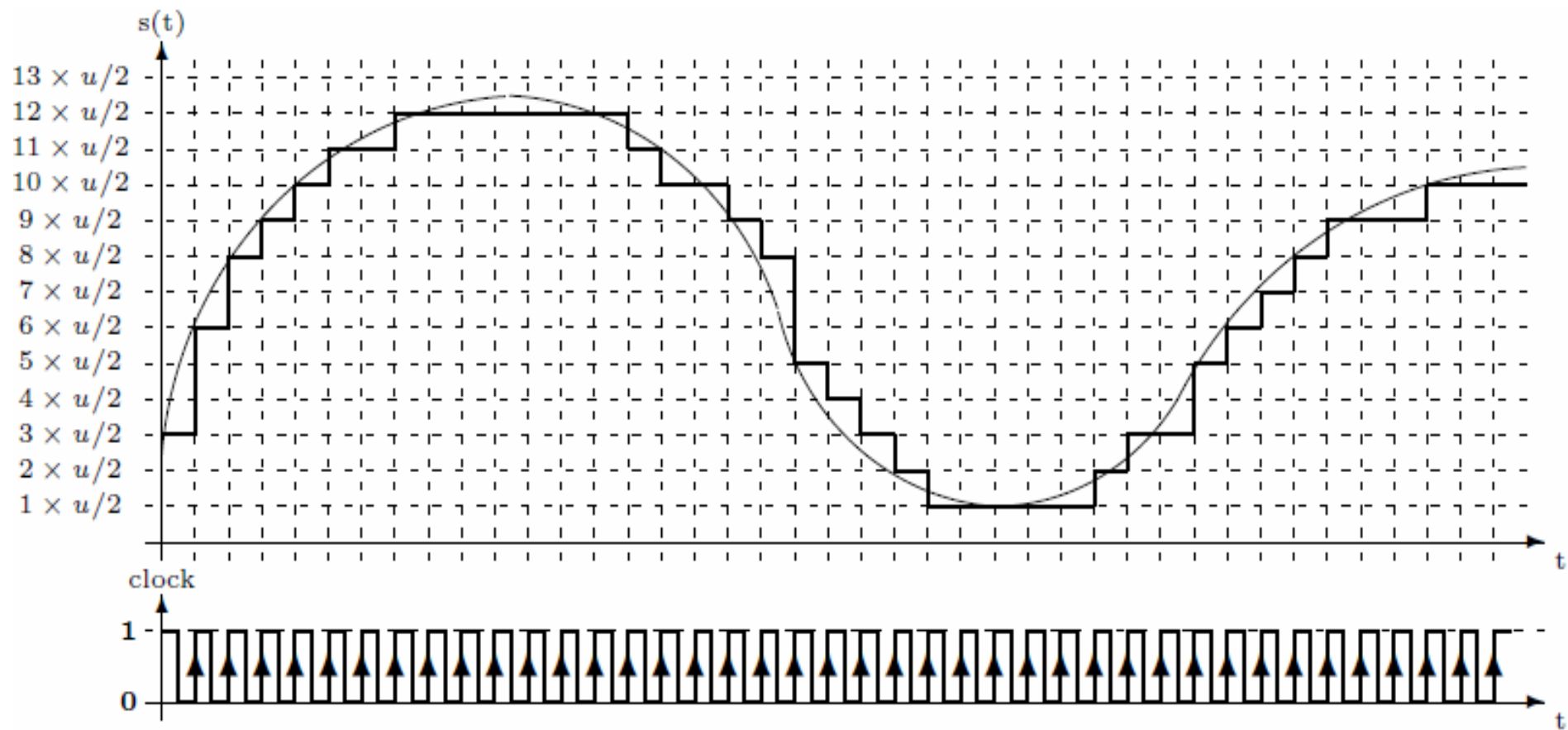
- Aritmetic
 - numere
 - aritmetică binară
- **Logic**
 - valori logice (adevărat/fals)
 - algebră logică
- Simbolic
 - diferite simboluri
 - teoria limbilor formale

Un sistem digital



Conversia analog - digital







Temă

- Desenați formele de undă pentru semnalele digitale corespunzătoare conversiei analog-digital din figurile anterioare



Circuitele digitale

- Versiunea cea mai simplă a sistemelor digitale
- Toate semnalele sunt digitale

- Sunt caracterizate de:
 - Conexiuni
 - Funcția realizată



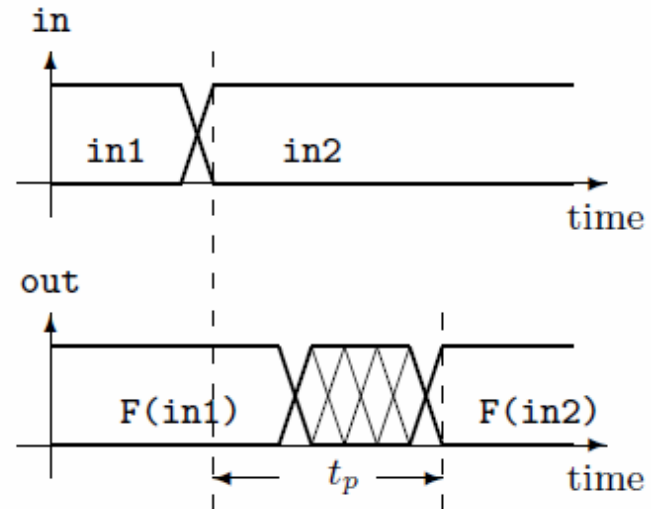
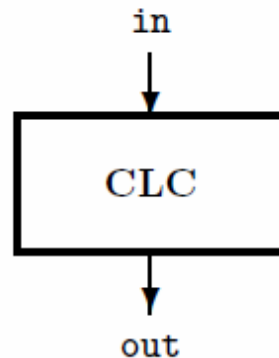
Exemple simple

- Sumatorul
- Numărătorul

- Circuite combinaționale (logice)
- Circuite secvențiale
 - Semnalul de ceas

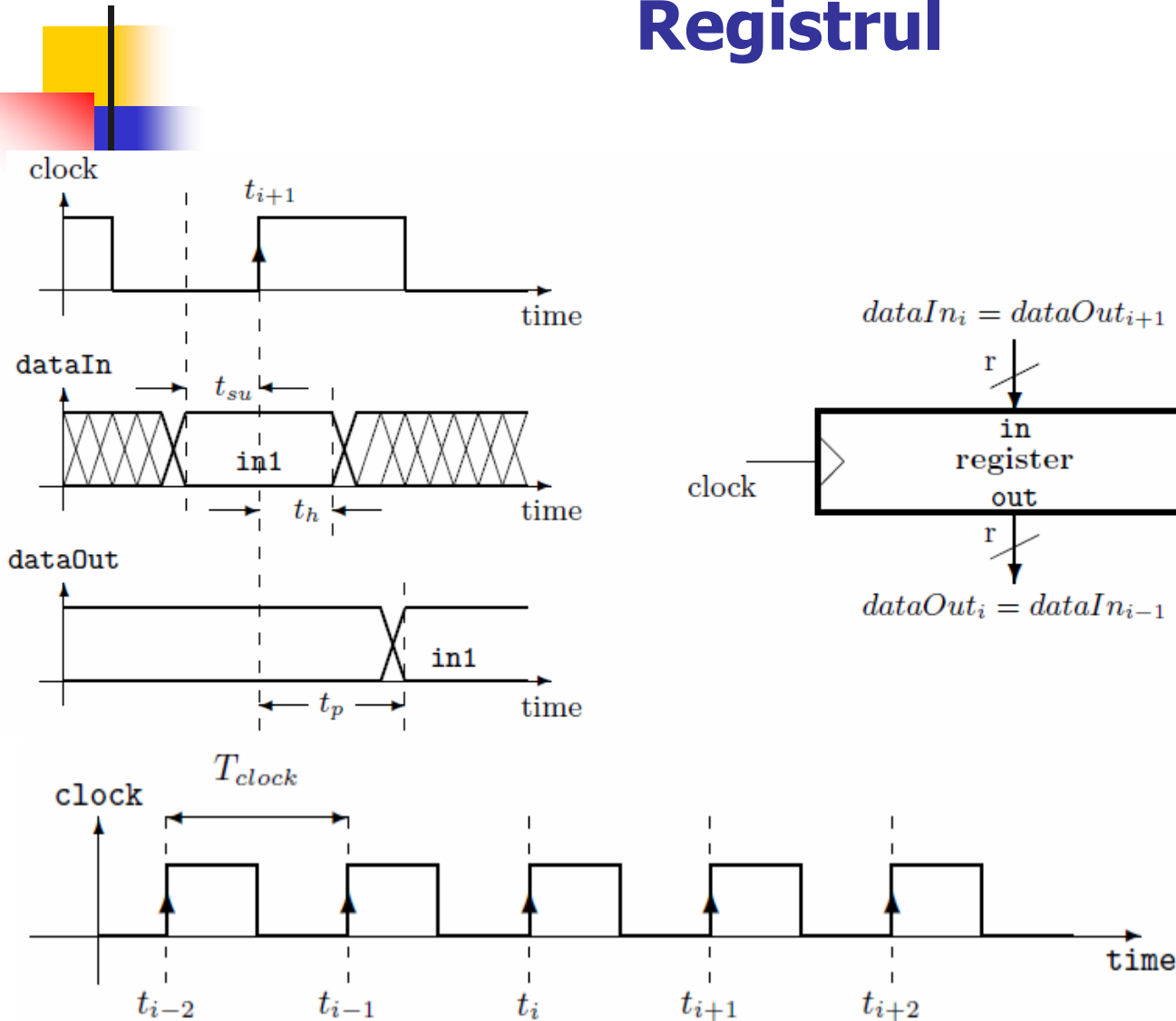
Circuit combinational

X_{n-1}	X_{n-2}	...	X_1	X_0	Y_{m-1}	Y_{m-2}	...		
0	0	...	0	0	1	0	1	0	...
0	0	...	0	1	0	1	0	0	...
0	0	...	1	0	0	1	0	1	...
		
1	1	...	0	1	0	1	0	1	...
1	1	...	1	0	0	1	1	0	...
1	1	...	1	1	1	0	1	0	...



La intrare: configurații binare pe n biți (le luăm în considerare pe toate)
 ieșirea se poate modifica de oricâte ori se modifică o intrare
 Pe ieșire apar doar o parte din configurațiile binare pe m biți
 t_p : timp de propagare
 intervalul marcat – tranziția ieșirii după comutarea intrării

Registrul



t_{su} : timp de set-up
 t_{hold} : timp de hold
 t_p : timp de propagare
 (de la frontul ceasului la ieșire)

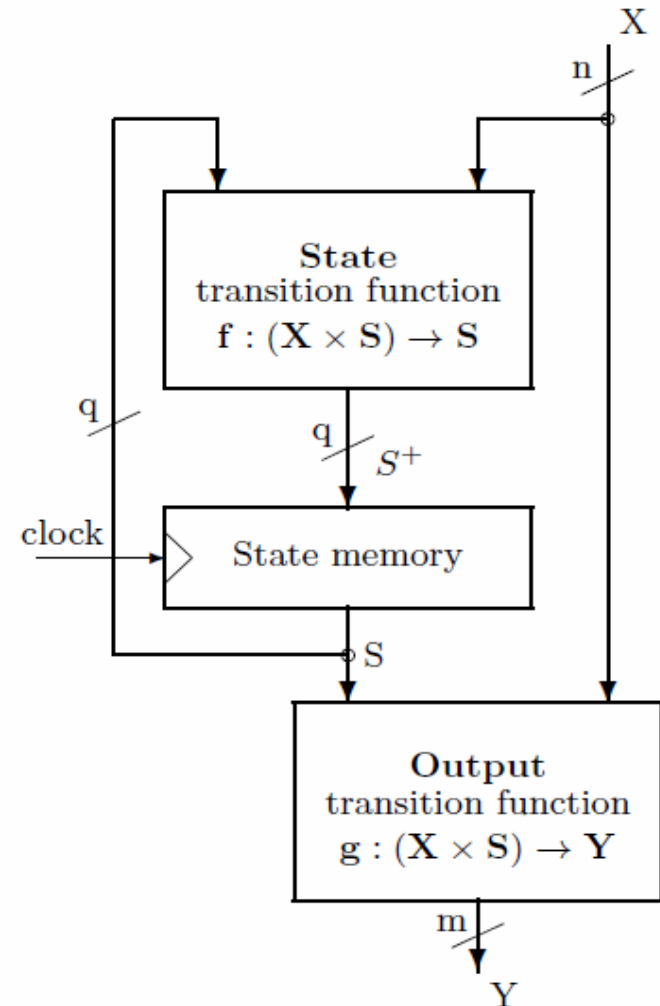
T_{clock} : perioada
 semnalului de ceas

$\dots t_{i-1} t_i t_{i+1} \dots$: timpul
 este discret

Intrările se pot
 modifica doar în
 intervalele indicate

Definirea formală a unui sistem digital

- porturi:
 - Intrări
 - Ieșiri
- Starea internă (mulțimea stărilor posibile)
- Funcțiile care determină:
 - Tranziția stărilor
 - Calculul ieșirilor





Noțiuni preliminare



Limbajul Verilog

- Limbajul de descriere hardware Verilog HDL este folosit pentru descrierea circuitelor digitale, pentru
 - descrierea circuitelor
 - verificare si testare
 - simulare
- se poate face descrierea circuitelor de la nivel de sistem până la nivel de tranzistor
- de obicei lucrăm cu descrieri RTL (*register transfer logic*)



Software

- Xilinx ISE
 - pentru platforme experimentale cu circuite Xilinx
 - implementarea circuitelor proiectate în FPGA
 - optimizarea proiectului pe FPGA
- ModelSim
 - simulator de circuite digitale
 - elaborarea proiectelor
 - compilare, verificare, testare, debugging
 - simulare



Noțiuni de bază

- conceptul de *modul*
- corespondența cu circuite reale
- NU se face optimizarea la nivel de algoritm
- conceptul de *testbench*



Module

- modulele nu se pot intersecta, include, apela unele pe altele
- principala modalitate de a apela un modul se numește **instanțiere**
- semnalele pot avea nume identice cu porturile și nume identice în diferite module



Semnale și variabile

- diferența porturi / semnale
- diferența semnale / variabile
- vectori

- două clase principale
 - *net*
 - *reg*



Porturile unui circuit

- trei tipuri
 - in
 - out
 - inout

- două modalități de definire
 - in lista de porturi
 - in cadrul definițiilor



Instanțierea modulelor

- utilizarea modulelor în circuit, prin conectarea unor semnale din circuit la porturile modulului, sau prin interconectarea porturilor unor module
- trebuie urmărită consistența / compatibilitatea interconexiunilor
- erori și avertismente (warninguri)



Descrierea circuitelor

- structurală
- comportamentală – *behavioural*

- cele două tipuri de descriere a circuitelor sunt axate pe
 - schemă
 - algoritm



Secvențial vs. combinațional

- În principiu nu trebuie să amestecăm descrierile secvențiale și combinaționale, pentru ca programul să poată sintetiza corect circuitul
- de ex, dacă folosim o instrucțiune *if then...* fără *else...* ne așteptăm la sinteza unui latch

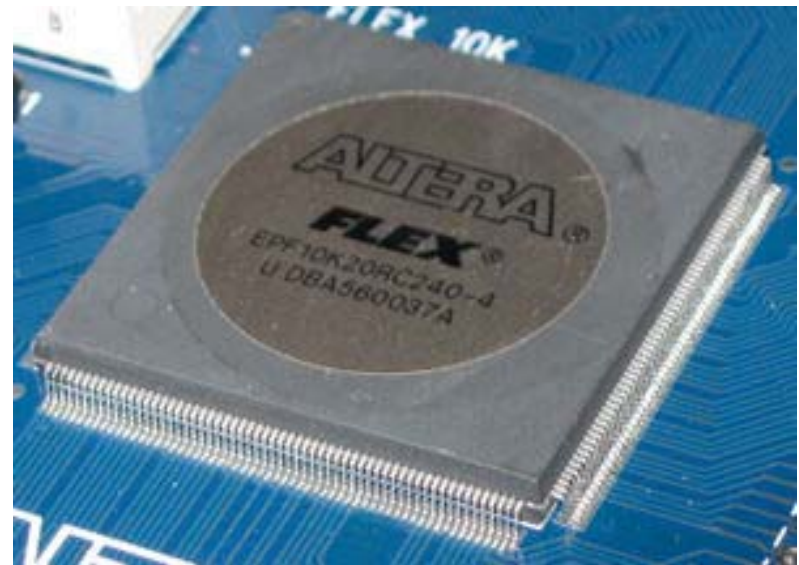


FPGA

Ce este un FPGA?

FPGA

- Field Programmable Gate Array
 - o clasă specială de PLD-uri (dispozitive logice programabile)
 - conțin o rețea de blocuri logice și de interconexiuni programabile





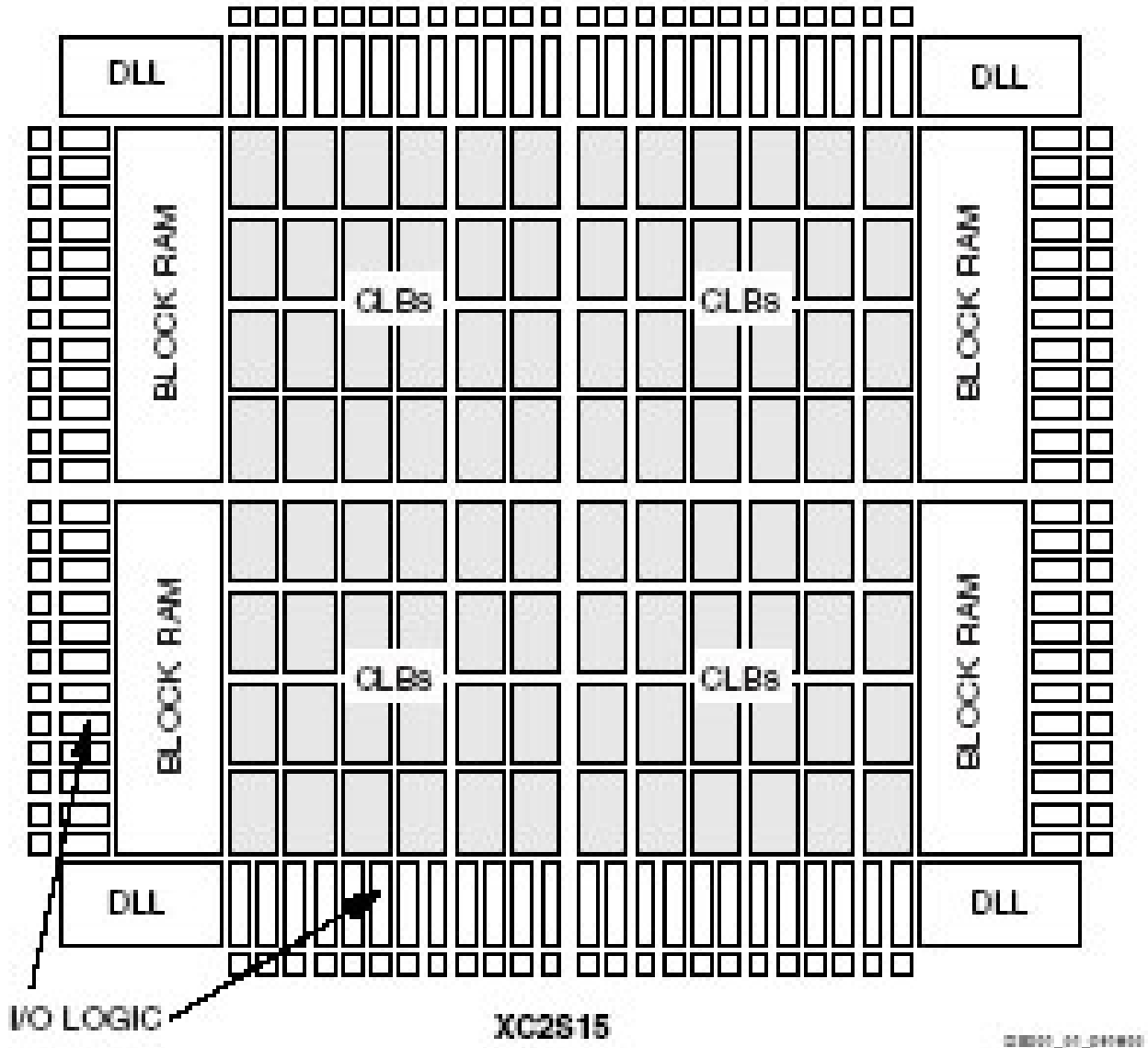
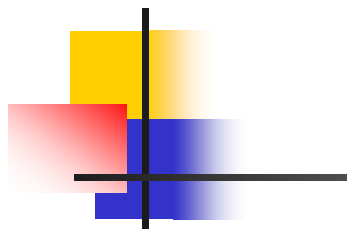
Structura unui FPGA

1. matrice de unități funcționale programabile (de obicei blocuri logice configurabile) care implementează logica sequentială și combinațională
2. structură de interconectare programabilă care stabilește rutarea semnalelor
3. o memorie pentru configurare, care programează funcționalitatea dispozitivului
4. resursele I/O



Specific FPGA

1. performanța pentru orice aplicație depinde de rutare
2. funcționalitatea este implementată prin tabele de adevăr
3. în general sunt volatile



Basic Spartan-II Family FPGA Block Diagram



Programarea unui FPGA

- un program (secvență binară complexă) încărcat într-o memorie CMOS statică.
- conținutul acestui SRAM este aplicat apoi liniilor de control ale porților de transmisie și celorlalte dispozitive
- se programează
 1. funcționalitatea blocurilor logice
 2. se configurează diverși parametri și caracteristici
 3. se stabilește conectarea între diferitele blocuri funcționale
 4. sunt configurate porturile bidirecționale I/O



De ce avem nevoie de FPGA-uri?

- alternativă pentru ASIC (implementează pe un singur chip toată logica suplimentară aferentă unui sistem digital)
- posibilitatea reprogramării
- implementare relativ facilă (nu industrial)
- aplicație particulară în dezvoltarea prototipurilor ASIC