Белорусский государственный университет
информатики и радиоэлектроники

4-ый курс специальности ПОИТ

ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ ЦИФРОВОГО ПРОЕКТИРОВАНИЯ

2020

Программное Обеспечение Цифрового Проектирования

Осенний семестр 2020 г.

Лектор: Иванюк Александр Александрович, проф. каф. информатики

Консультации:

Понедельник 15:00-15:50 ауд. 402-5

e-mail: ivaniuk@bsuir.by
sit.bachelors.bsuir@gmail.com

Расписание лекций:

Понедельник	15:50 – 17:10	ауд. 108-4	(*)
Понедельник	17:25 – 18:45	ауд. 108-4	(2)

Лабораторные работы: ассистенты каф. ПОИТ
Видничук В.П., Шамына А.Ю.

Промежуточные контрольные работы
Зачет

Программное Обеспечение Цифрового Проектирования

Магистратура: Проектирование Цифровых Систем на Устройствах Программируемой Логики

Аспирантура: 05.13.05

9. Методы и алгоритмы синтеза элементов и устройств для неклонируемой идентификации средств вычислительной техники и систем управления, а также для генерирования невоспроизводимых случайных последовательностей.

10. Теория, методы и алгоритмы проектирования элементов и устройств вычислительной техники и систем управления, обеспечивающих их уникальность, аутентичность, структурную целостность и защиту от несанкционированного использования.

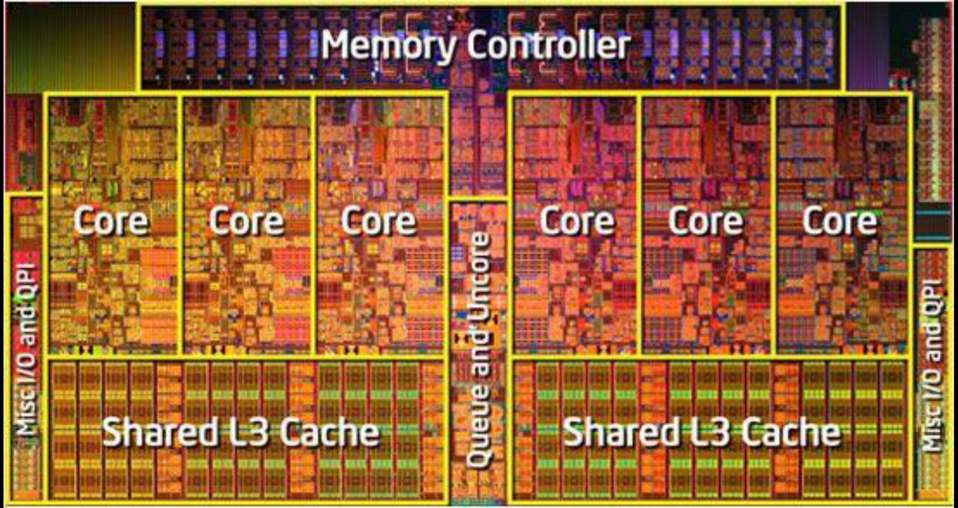
05.13.15:

13. Методы проектирования, в том числе контролепригодного, вычислительных машин, комплексов и компьютерных сетей, обеспечивающих их самотестирование, идентификацию и аутентификацию с целью защиты от несанкционированного использования, структурных изменений и реализации лицензирования, в том числе удаленного.

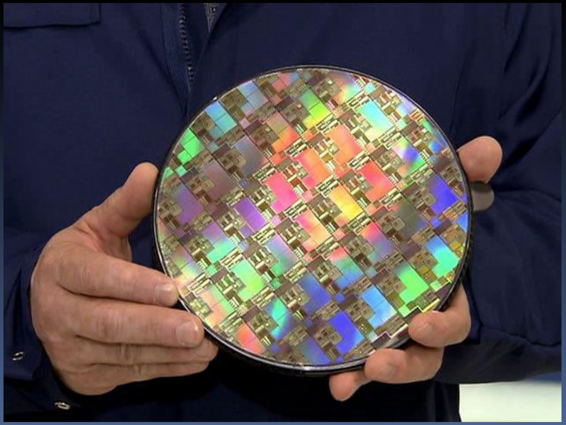
Digital Design:

IC – Integrated Circuit

ASIC – Application Specific IC



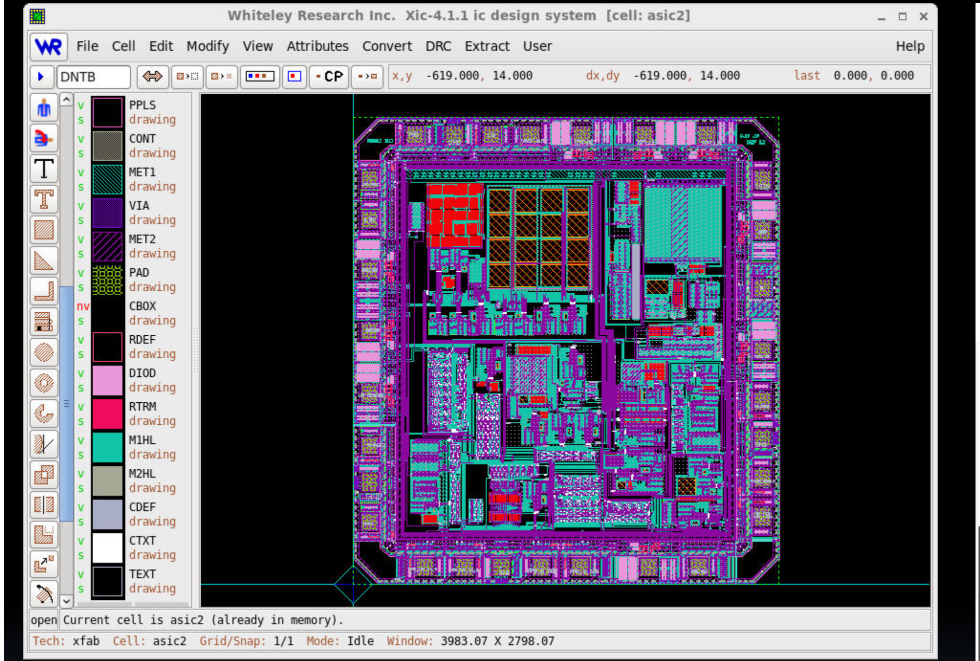
IC Die



Semiconductor Plate

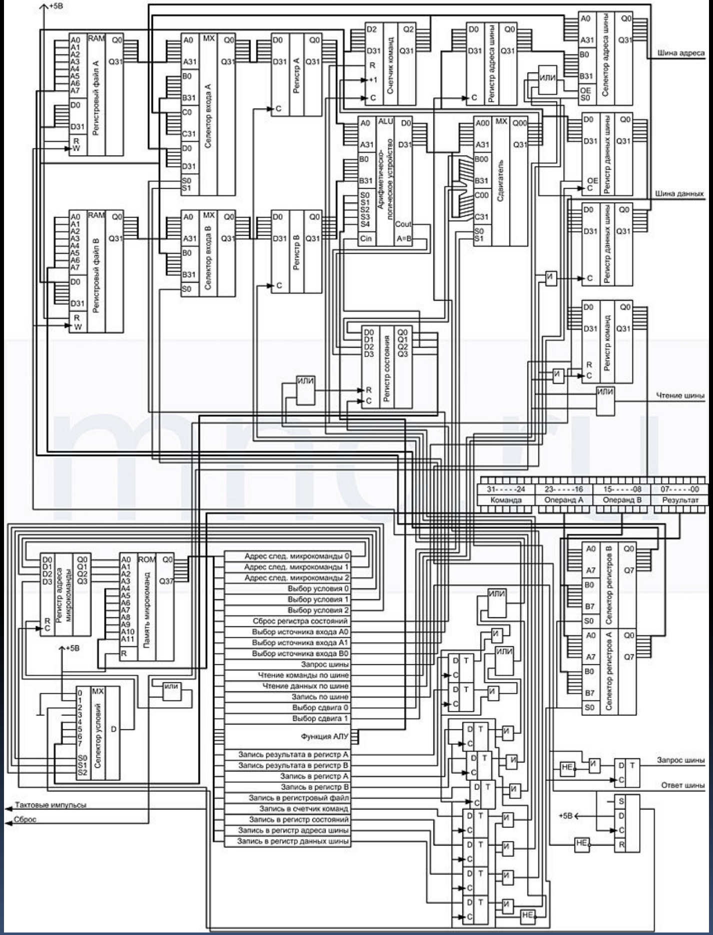
Программное Обеспечение Цифрового Проектирования

Digital Design:



```
x,y  -619.000, 14.000     dx,dy  -619.000, 14.000     last  0.000, 0.000
open Current cell is asic2 (already in memory).
Tech: xfab  Cell: asic2  Grid/Snap: 1/1  Mode: Idle  Window: 3983.07 X 2798.07
```

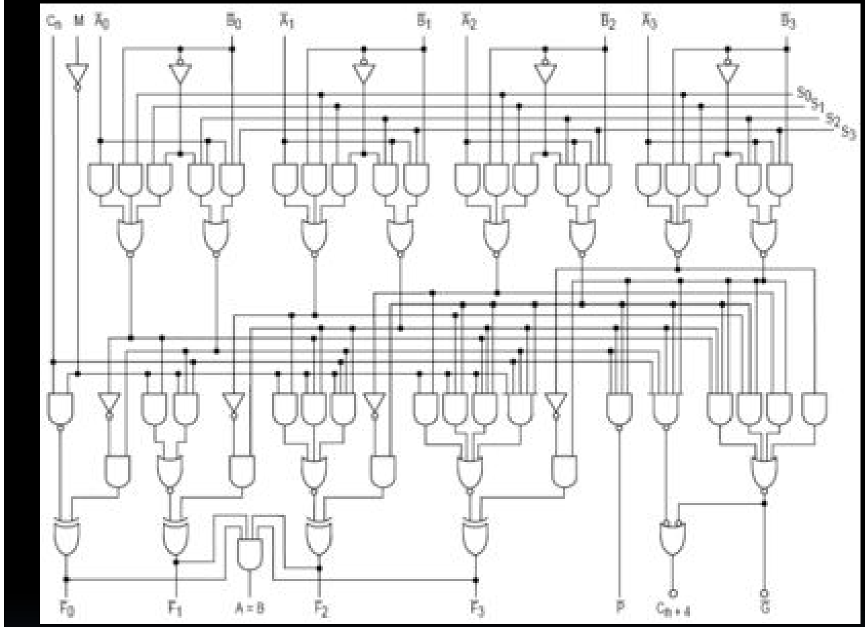
Physical Layout of IC



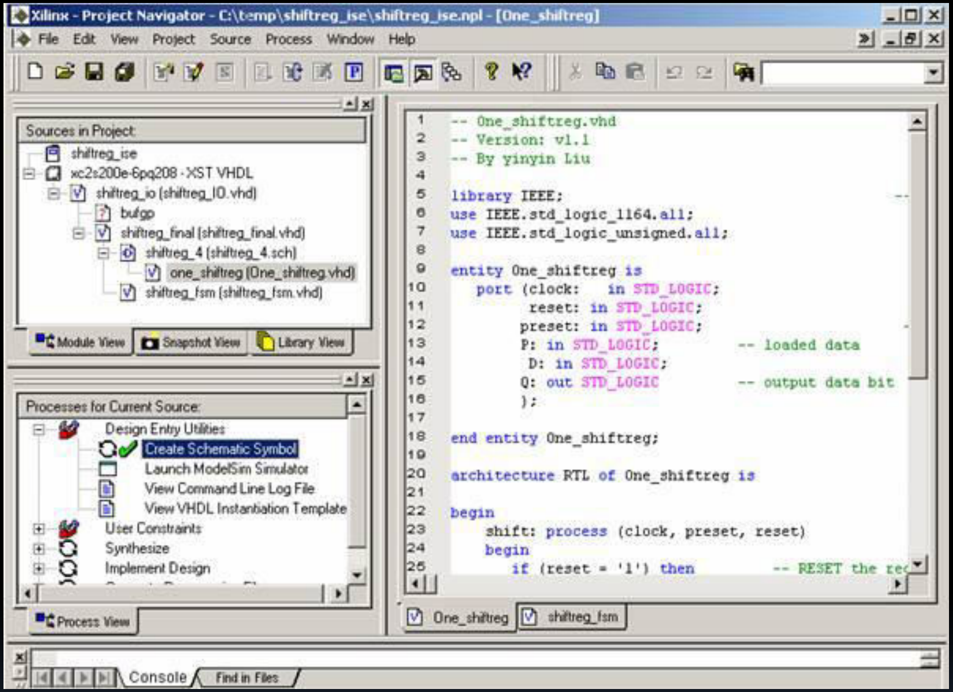
Digital Scheme

Программное Обеспечение Цифрового Проектирования

Digital Design:



Logic Scheme



Xilinx - Project Navigator - C:\temp\shiftreg_ise\shiftreg_ise.npl - [One_shiftreg]

Sources in Project:

- shiftreg_ise
- xc2s200e-6pq208 - XST VHDL
  - shiftreg_io (shiftreg_IO.vhd)
    - bufgp
    - shiftreg_final (shiftreg_final.vhd)
      - shiftreg_4 (shiftreg_4.sch)
        - one_shiftreg (One_shiftreg.vhd)
      - shiftreg_fsm (shiftreg_fsm.vhd)

Processes for Current Source:

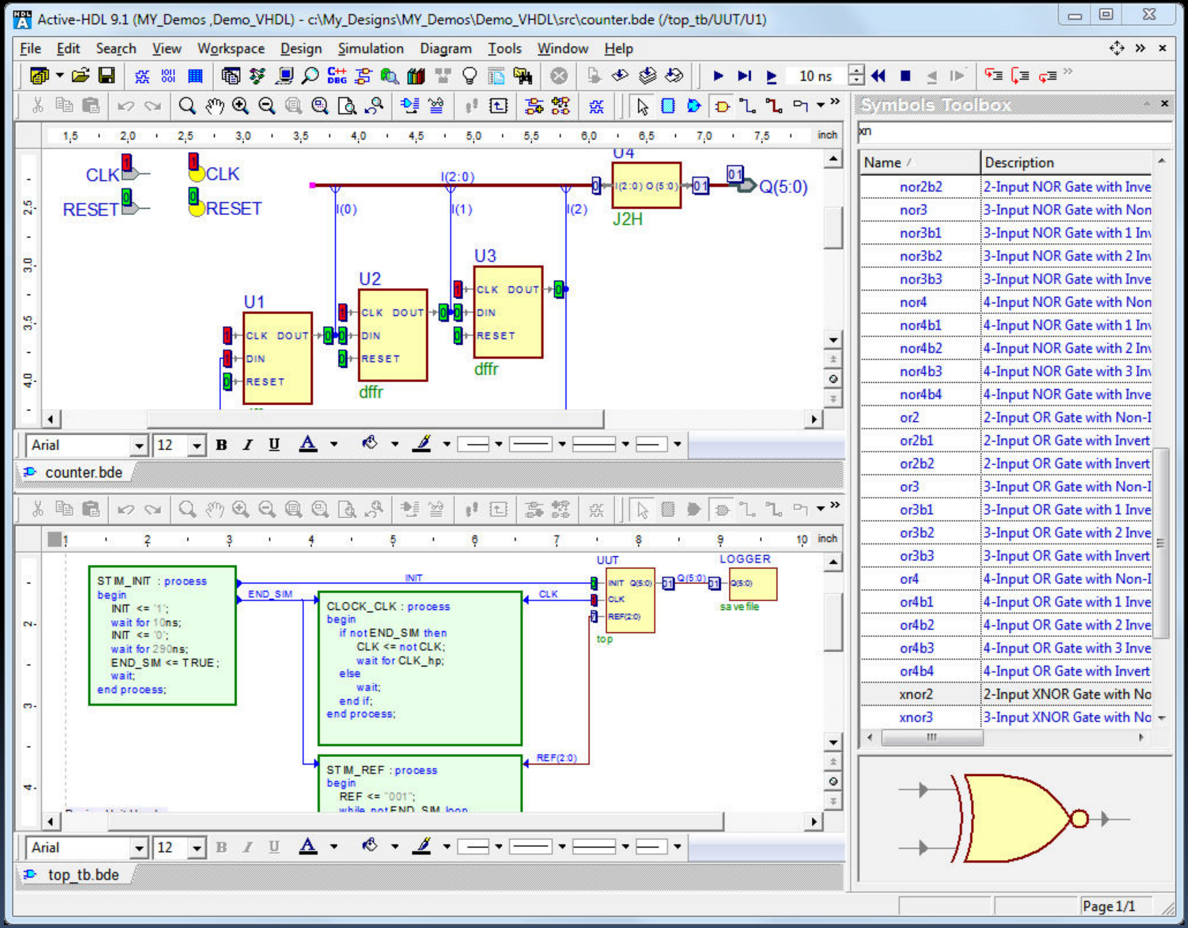
- Design Entry Utilities
  - Create Schematic Symbol
  - Launch ModelSim Simulator
  - View Command Line Log File
  - View VHDL Instantiation Template
- User Constraints
- Synthesize
- Implement Design

```
-- One_shiftreg.vhd
-- Version: v1.1
-- By yinyin Liu

library IEEE;                                                      --
use IEEE.std_logic_1164.all;
use IEEE.std_logic_unsigned.all;

entity One_shiftreg is
   port (clock:   in STD_LOGIC;
          reset: in STD_LOGIC;
         preset: in STD_LOGIC;
         P: in STD_LOGIC;           -- loaded data
          D: in STD_LOGIC;
         Q: out STD_LOGIC           -- output data bit
         );

end entity One_shiftreg;

architecture RTL of One_shiftreg is

begin
    shift: process (clock, preset, reset)
    begin
       if (reset = '1') then          -- RESET the rec
```

HDL-description

Программное Обеспечение Цифрового Проектирования

Digital Design:



```
STIM_INIT  : process
begin
    INIT  <= '1';
    wait for 10ns;
    INIT  <= '0';
    wait for 290ns;
    END_SIM <= TRUE;
    wait;
end process;
```

```
CLOCK_CLK : process
begin
    if not END_SIM then
        CLK <= not CLK;
        wait for CLK_hp;
    else
        wait;
    end if;
end process;
```

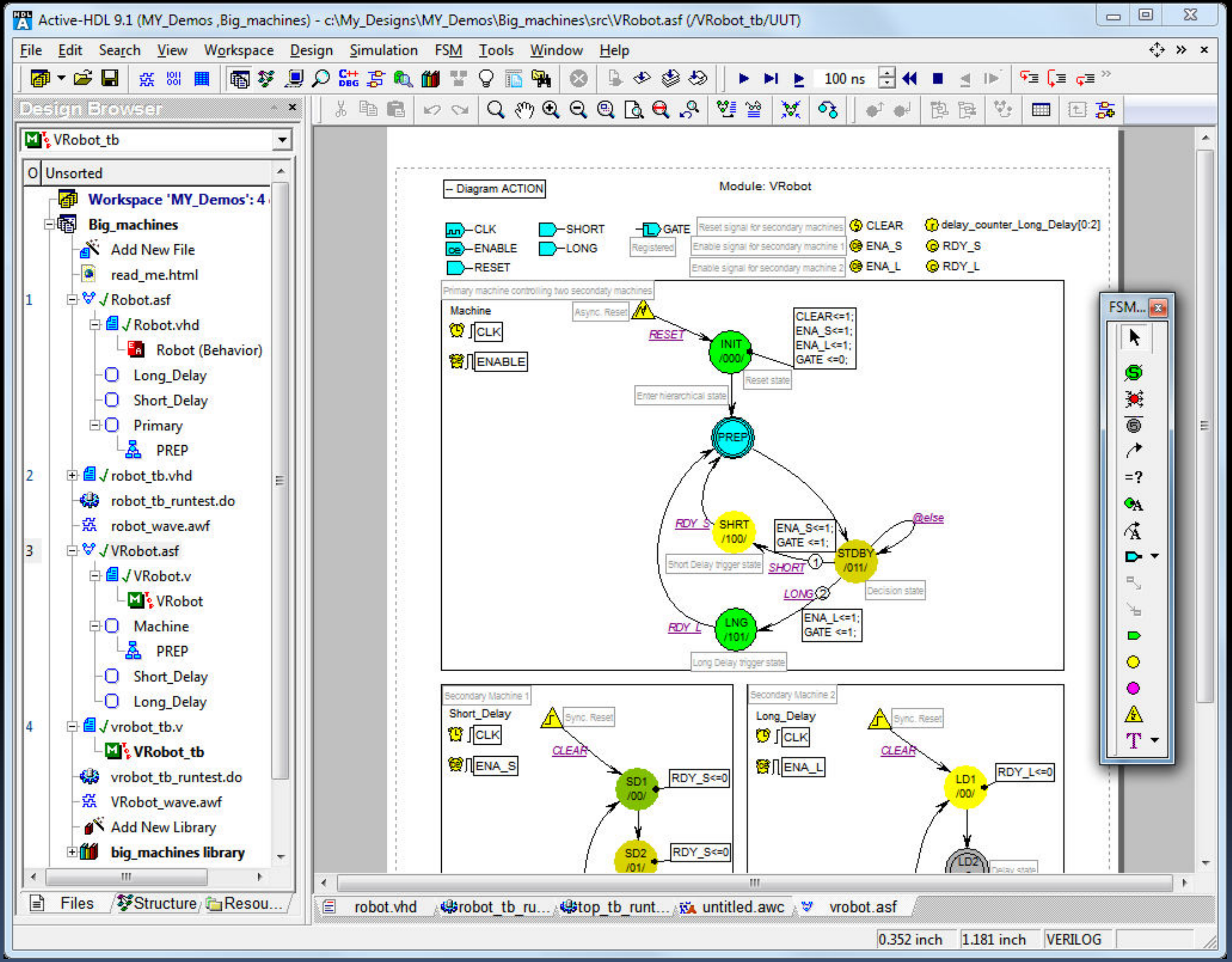
```
STIM_REF : process
begin
    REF <= "001";
    while not END_SIM loop
```

xn

Name	Description
nor2b2	2-Input NOR Gate with Inve
nor3	3-Input NOR Gate with Non
nor3b1	3-Input NOR Gate with 1 Inv
nor3b2	3-Input NOR Gate with 2 Inv
nor3b3	3-Input NOR Gate with Inve
nor4	4-Input NOR Gate with Non
nor4b1	4-Input NOR Gate with 1 Inv
nor4b2	4-Input NOR Gate with 2 Inv
nor4b3	4-Input NOR Gate with 3 Inv
nor4b4	4-Input NOR Gate with Inve
or2	2-Input OR Gate with Non-I
or2b1	2-Input OR Gate with Invert
or2b2	2-Input OR Gate with Invert
or3	3-Input OR Gate with Non-I
or3b1	3-Input OR Gate with 1 Inve
or3b2	3-Input OR Gate with 2 Inve
or3b3	3-Input OR Gate with Invert
or4	4-Input OR Gate with Non-I
or4b1	4-Input OR Gate with 1 Inve
or4b2	4-Input OR Gate with 2 Inve
or4b3	4-Input OR Gate with 3 Inve
or4b4	4-Input OR Gate with Invert
xnor2	2-Input XNOR Gate with No
xnor3	3-Input XNOR Gate with No

Block Scheme Editor

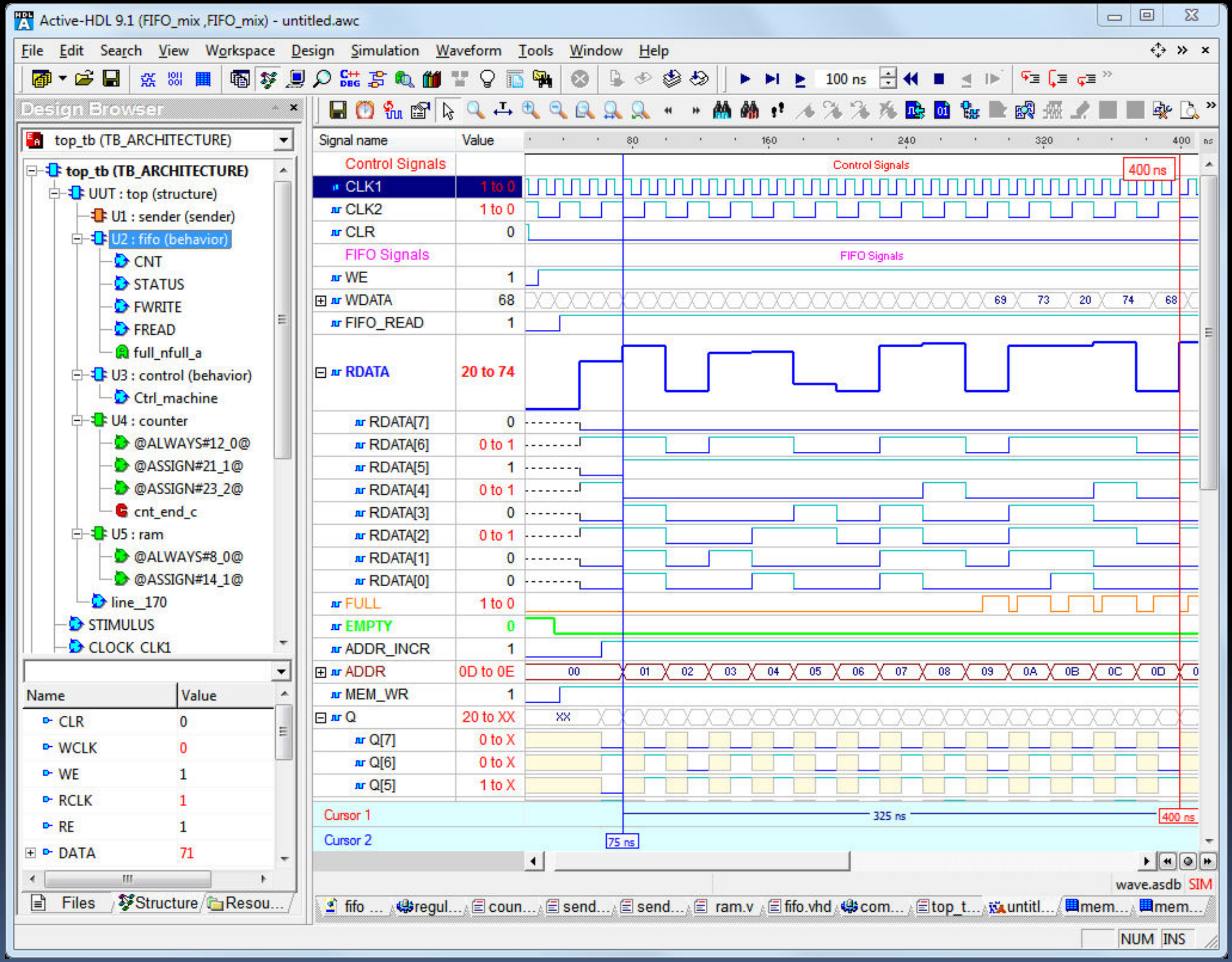
Digital Design:



State Diagram (FSM) Editor

Программное Обеспечение Цифрового Проектирования

Digital Design:



Signal name	Value
Control Signals	
CLK1	1 to 0
CLK2	1 to 0
CLR	0
FIFO Signals	
WE	1
WDATA	68
FIFO_READ	1
RDATA	20 to 74
RDATA[7]	0
RDATA[6]	0 to 1
RDATA[5]	1
RDATA[4]	0 to 1
RDATA[3]	0
RDATA[2]	0 to 1
RDATA[1]	0
RDATA[0]	0
FULL	1 to 0
EMPTY	0
ADDR_INCR	1
ADDR	0D to 0E
MEM_WR	1
Q	20 to XX
Q[7]	0 to X
Q[6]	0 to X
Q[5]	1 to X
Cursor 1	
Cursor 2	

Name	Value
CLR	0
WCLK	0
WE	1
RCLK	1
RE	1
DATA	71

Waveform Editor & Debugger

Digital Design:

Programmable Logic Devices (PLD)

Field Programmable Gate Arrays (FPGA)

Программное Обеспечение Цифрового Проектирования

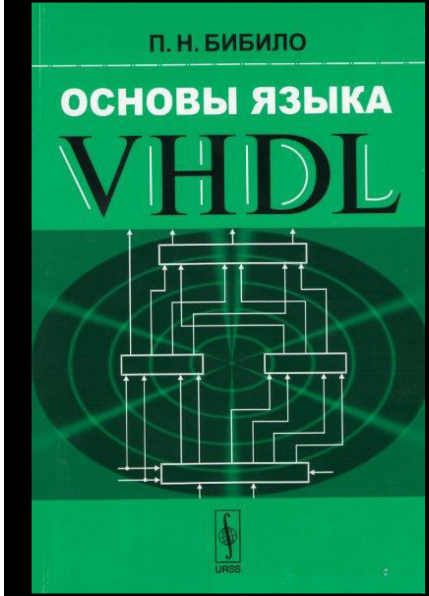
Литература:

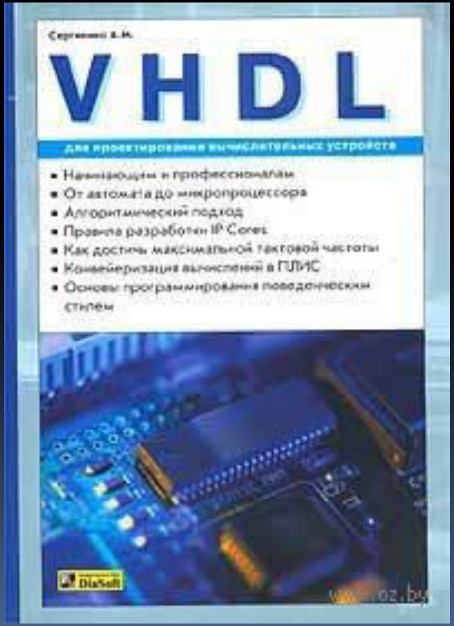
ПРОЕКТИРОВАНИЕ ЦИФРОВЫХ УСТРОЙСТВ
Дж.Уэйкерли
I
РОЕКТИРОВАНИЕ
ЫХ УСТРОЙСТВ
Дж.Уэйкерли
II

Е. П. Угрюмов
Цифровая схемотехника
3-е издание
bhv
Е. П. Угрюмов
Master-serial mode
DRAM
ЦИФРОВАЯ СХЕМОТЕХНИКА
2–е ИЗДАНИЕ
ИС всех уровней интеграции: от простейших логических элементов до перспективных БИС/СБИС с программируемыми структурами
Принципы построения цифровых узлов и их практическая реализация
Методы проектирования цифровых устройств
CFG_IN
CPLD
УЧЕБНОЕ ПОСОБИЕ
bhv

Программное Обеспечение Цифрового Проектирования

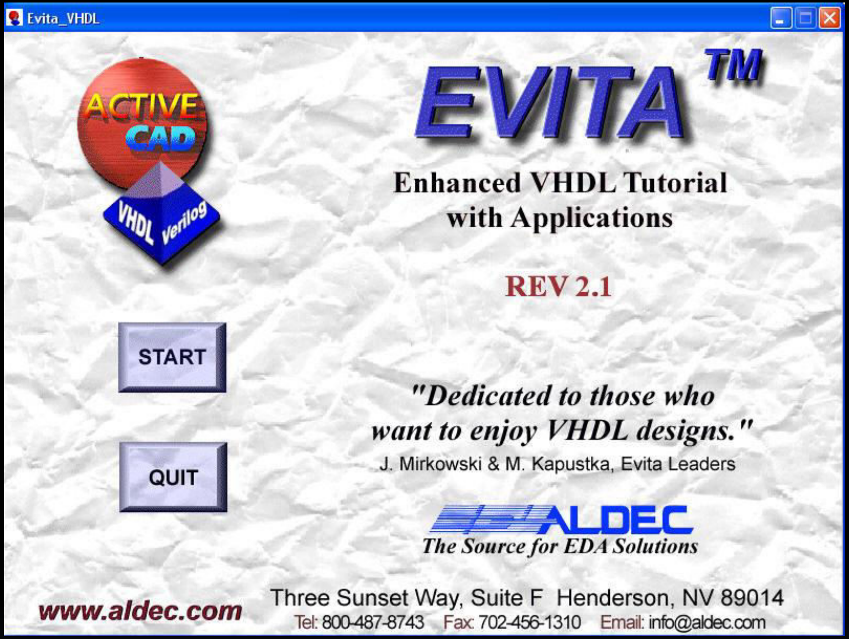
Литература:




Сергиенко А.М.
VHDL
для проектирования вычислительных устройств
Начинающим и профессионалам
От автомата до микропроцессора
Алгоритмический подход
Правила разработки IP Cores
Как достичь максимальной тактовой частоты
Конвейеризация вычислений в ПЛИС
Основы программирования поведенческим стилем
DiaSoft
www.oz.by

Программное Обеспечение Цифрового Проектирования

Литература:



Evita_VHDL

EVITA TM

Enhanced VHDL Tutorial
with Applications

REV 2.1

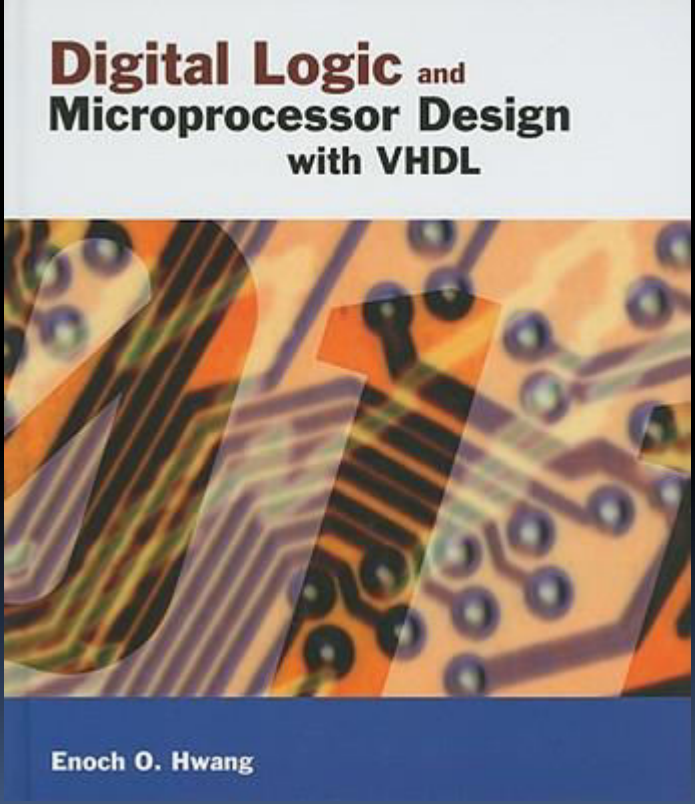
"Dedicated to those who want to enjoy VHDL designs."

J. Mirkowski & M. Kapustka, Evita Leaders

ALDEC
The Source for EDA Solutions

www.aldec.com

Three Sunset Way, Suite F Henderson, NV 89014
Tel: 800-487-8743 Fax: 702-456-1310 Email: info@aldec.com

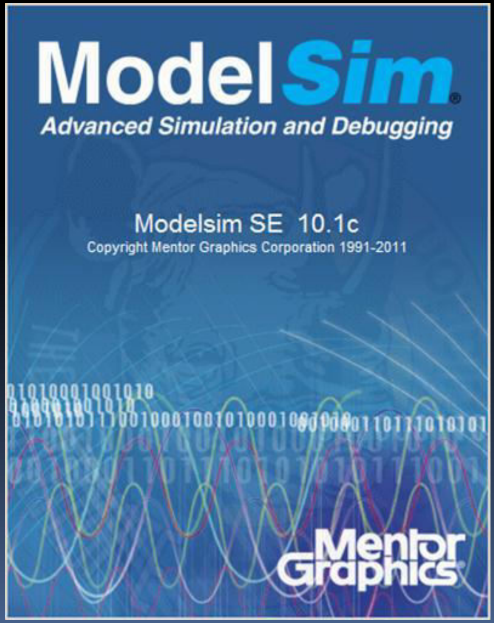


Digital Logic and Microprocessor Design with VHDL

Enoch O. Hwang

Программное Обеспечение Цифрового Проектирования

Программное обеспечение:


ModelSim®
Advanced Simulation and Debugging
Modelsim SE 10.1c
Copyright Mentor Graphics Corporation 1991-2011
Mentor Graphics

Аппаратное обеспечение:

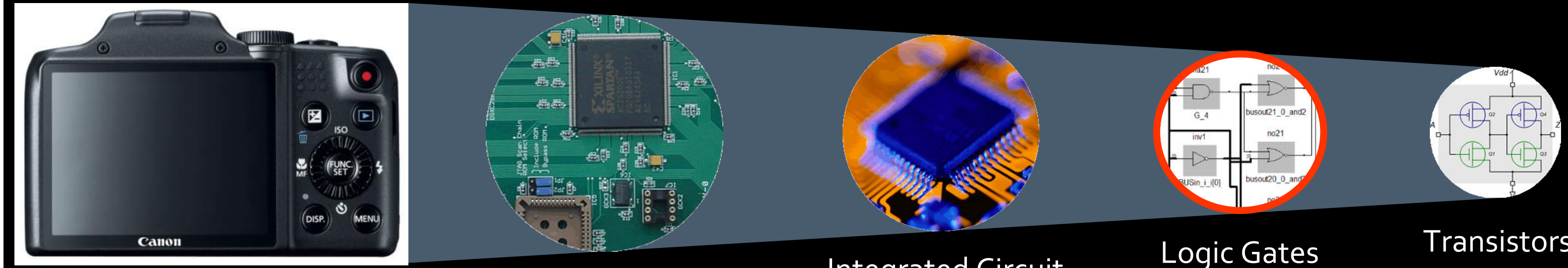
Digilent NEXYS 4

FPGA-Based Development Board based on Xilinx Artix-7

Design of Digital (Systems) Devices = Digital Design

Key words: hardware (HW), digital system, digital device, circuit, integrated circuit (IC), logic gate, computer aided design (CAD), hardware description language (HDL), register transfer level (RTL), finite state machine (FSM), microprogrammed FSM (MP-FSM), flip-flop (FF), latch, programmable logic device (PLD), field programmable gate array (FPGA), application specific IC (ASIC), configurable logic block (CLB), look-up table (LUT), test bench (TB), data path, synthesis, optimization, simulation, verification, timing diagram, waveform, state diagram, combinational logic, sequential logic, (re)configurable logic, design for test/trust (DFT), physical cryptography, IP cores, etc.

Digital {device, system, circuit}



Digital Camera

Digital System

Printed Circuit Board (PCB)

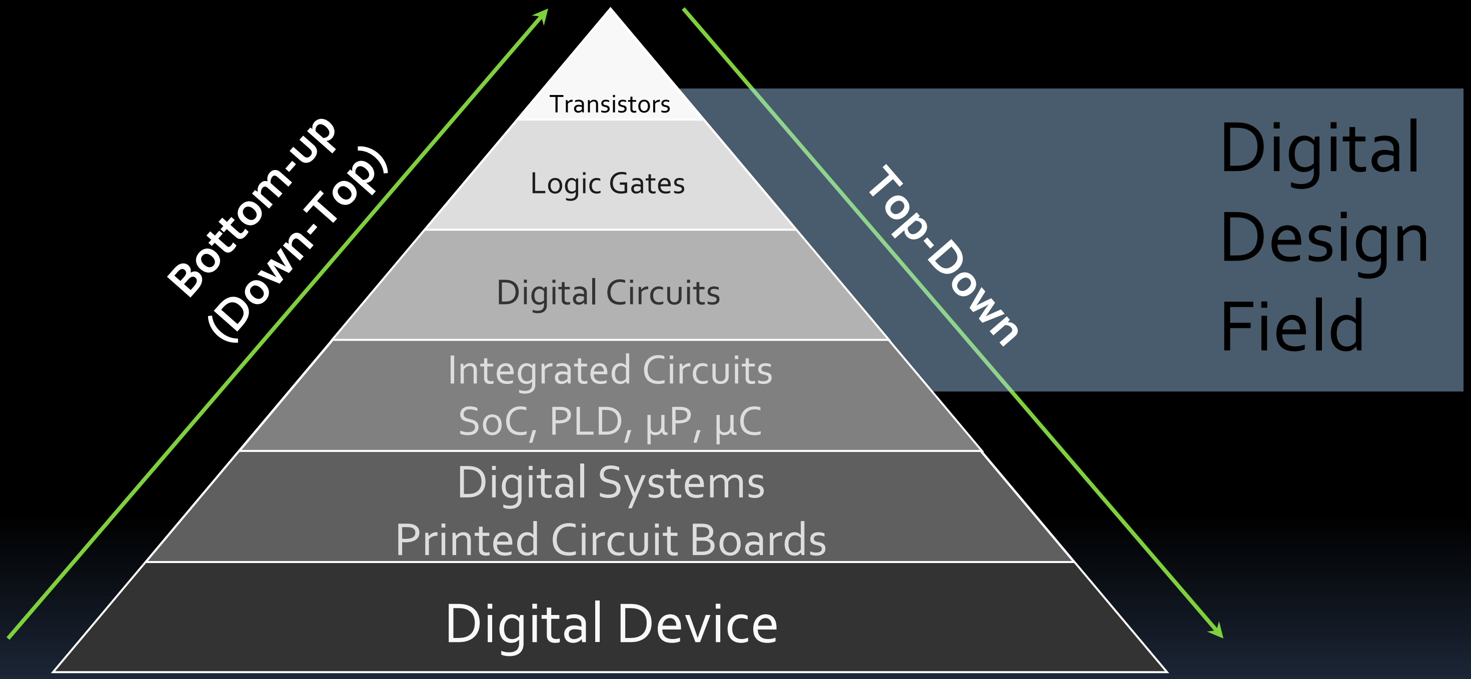
Integrated Circuit

System on a Chip (SoC)

Logic Gates

Transistors

Digital Design Methodologies

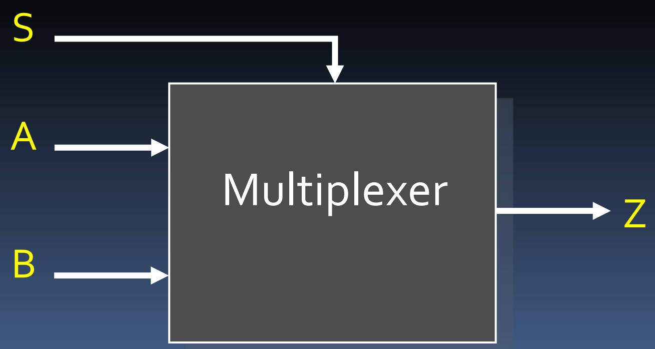


Digital Design *is a development of technical documentation, which allows to produce specified Digital Circuits in specified conditions.*

Digital Design

Hardware Description Problem

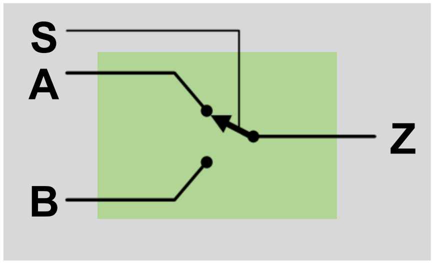
Problem definition: Please, describe digital device called "multiplexer". Multiplexer has got two binary inputs A and B, one control input S and one output Z. In the case which binary value input S assumes – '0' or '1', multiplexer transmits binary value from input A or B to the output Z.



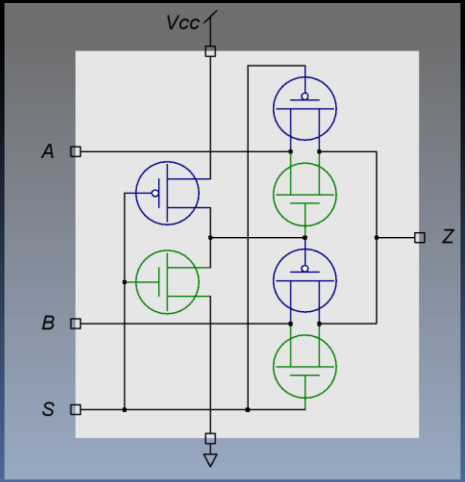
Digital Design

Hardware Description Problem

Description #1: It is possible to describe multiplexer as Electrical Switch.



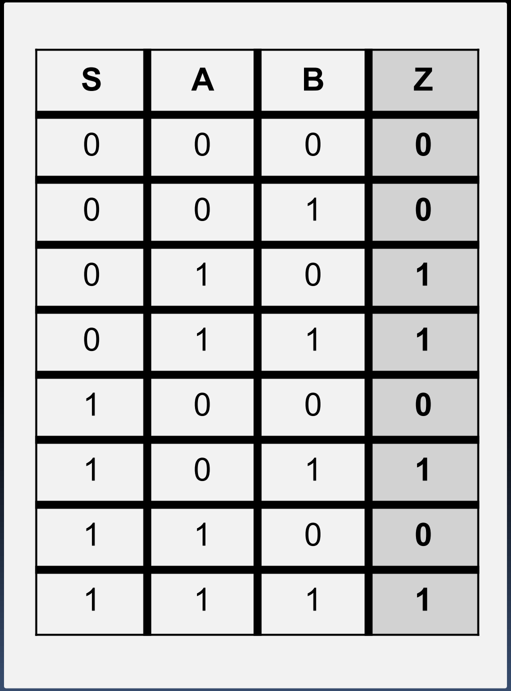
Description #2: Multiplexer can be described by Logical Switches. For example, in CMOS-technology transistors are the such switches.



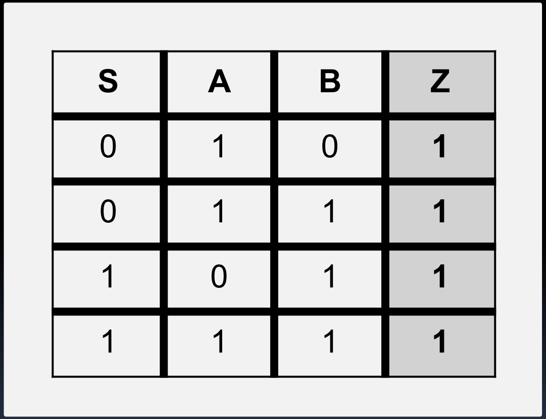
Digital Design

Hardware Description Problem

Description #3: We can describe the multiplexer by Table of Truth (Look-Up Table).



S	A	B	Z
0	0	0	**0**
0	0	1	**0**
0	1	0	**1**
0	1	1	**1**
1	0	0	**0**
1	0	1	**1**
1	1	0	**0**
1	1	1	**1**

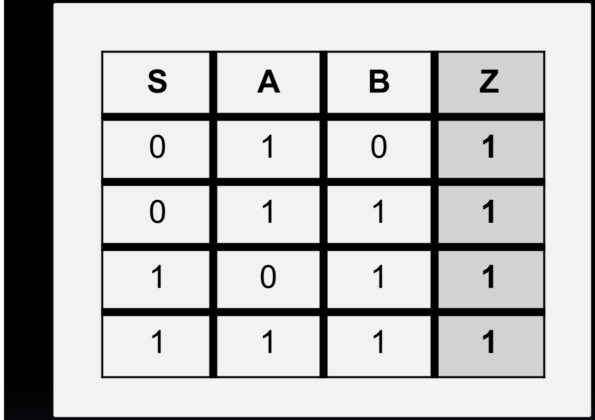


S	A	B	Z
0	1	0	**1**
0	1	1	**1**
1	0	1	**1**
1	1	1	**1**

Digital Design

Hardware Description Problem

Description #4: We can describe the multiplexer by Logical Function.



S	A	B	Z
0	1	0	**1**
0	1	1	**1**
1	0	1	**1**
1	1	1	**1**

Minterms for Z=1 are products

$m_2 = S'\cdot A\cdot B'$
$m_3 = S'\cdot A\cdot B$
$m_5 = S\cdot A'\cdot B$
$m_7 = S\cdot A\cdot B$

The SoP (Sum of Products)

$Z = m_2 + m_3 + m_5 + m_7$

$$Z = S'\cdot A\cdot B' + S'\cdot A\cdot B + S\cdot A'\cdot B + S\cdot A\cdot B$$

Operations: ’ + · are NOT, OR, AND logical operations

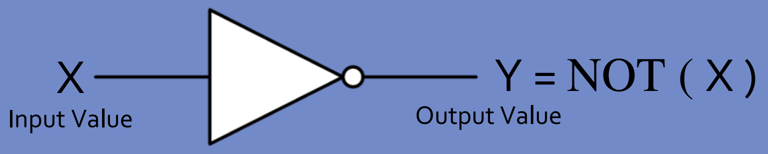
Digital Design

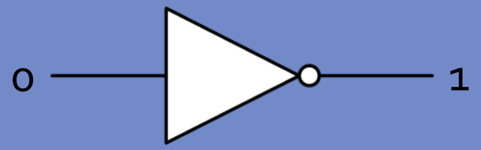
Hardware Description Problem

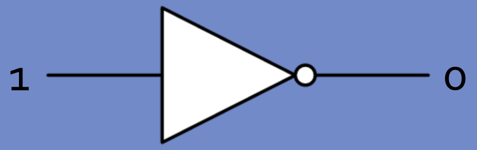
Basic Logical Functions: **NOT, OR, AND**

X	Y=NOT(X)
0	1
1	0

Become the logic function NOT has got only one argument logic gate NOT also has got only one input and one output. Graphical symbol of logic gate NOT is shown below.







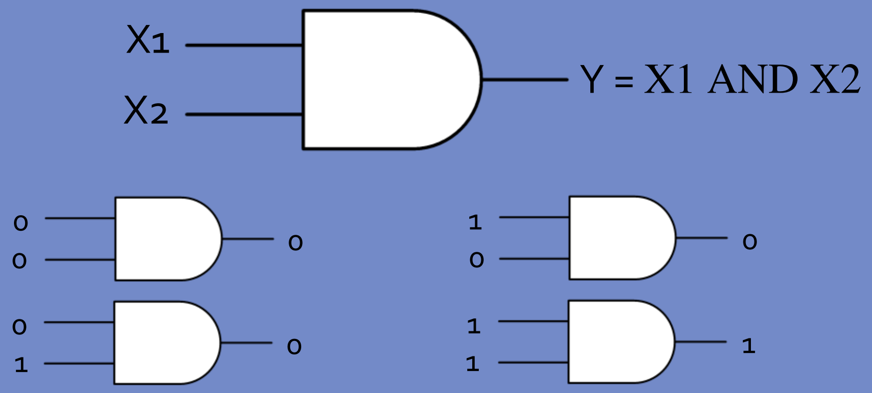
Digital Design

Hardware Description Problem

Basic Logical Functions: **NOT, OR, AND**

X1	X2	Y=X1 AND X2
0	0	0
0	1	0
1	0	0
1	1	1

Graphical Symbol



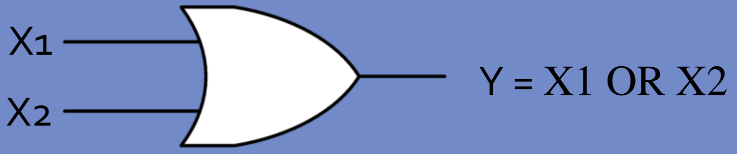
Digital Design

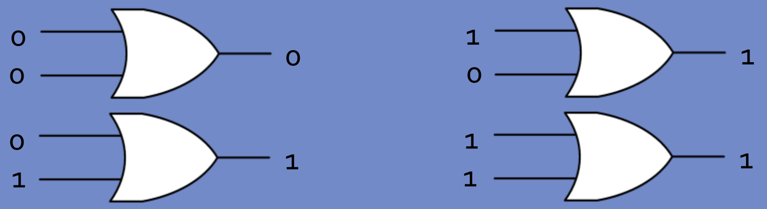
Hardware Description Problem

Basic Logical Functions: **NOT, OR, AND**

X1	X2	Y=X1 OR X2
0	0	**0**
0	1	**1**
1	0	**1**
1	1	**1**

Graphical Symbol





Digital Design

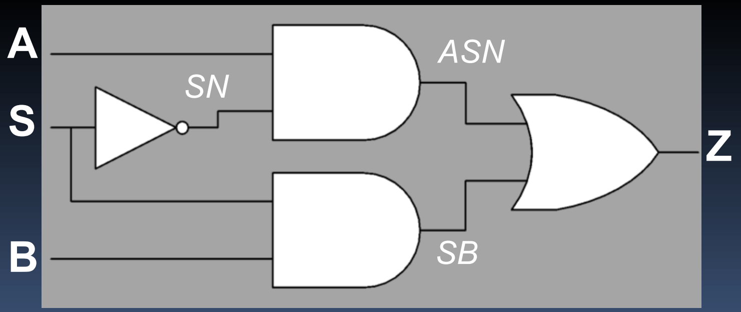
Hardware Description Problem

Description #5 : It is possible to optimize the Table of Truth notation using AND-OR NOT terms. For the multiplexer example we will have next expression:

$$Z = S'\cdot A\cdot B' + S'\cdot A\cdot B + S\cdot A'\cdot B + S\cdot A\cdot B$$

$$Z=S'\cdot A + S\cdot B$$

Description #6: Previous expression can be realized as logic circuit using NOT, 2-inputs AND and OR gates.



Digital Design

Hardware Description Problem

Description #7: It is possible to describe the multiplexer using high-level languages as **VHDL**. Here is an example of multiplexer **behavioral VHDL** description:

```
Library IEEE;
Use IEEE.std_logic_1164.all;
Entity MUX is
            port (
            A,B,S : in std_logic;
            Z       : out std_logic );
End MUX;
Architecture behavioral of MUX is
Begin
            Z <= A when S='0' else B;
End behavioral;
```

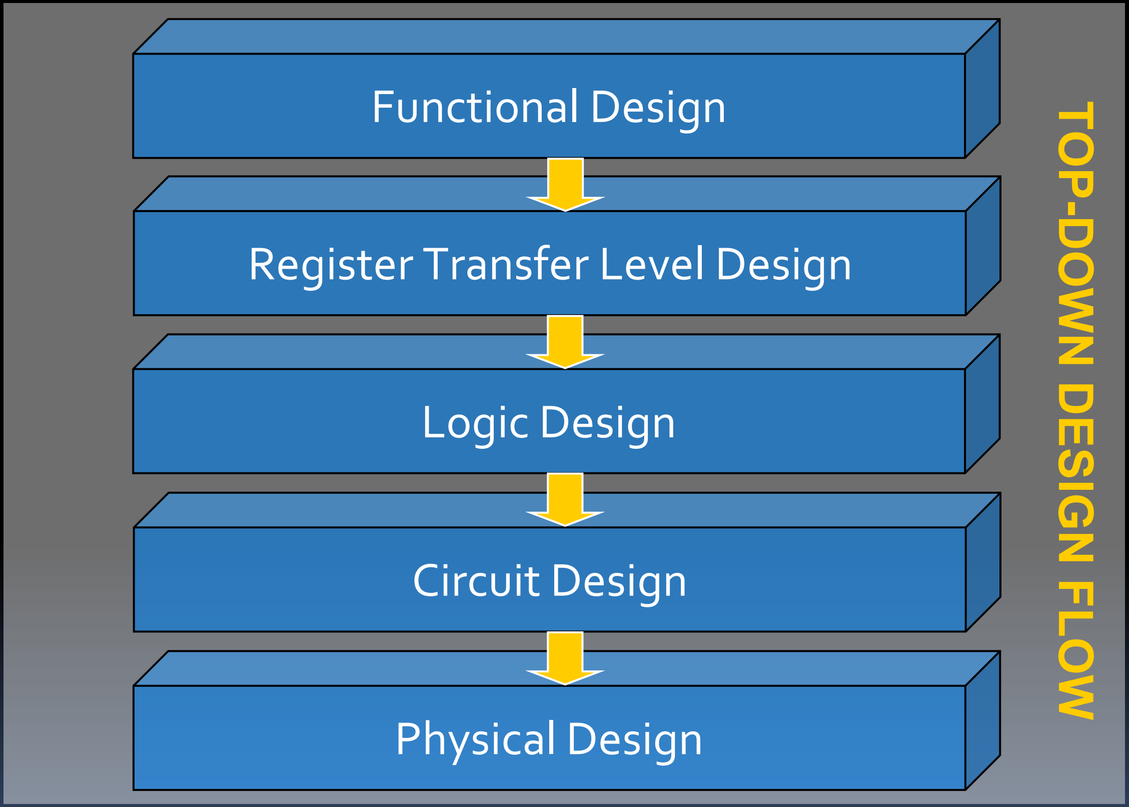
Digital Design

Hardware Description Problem

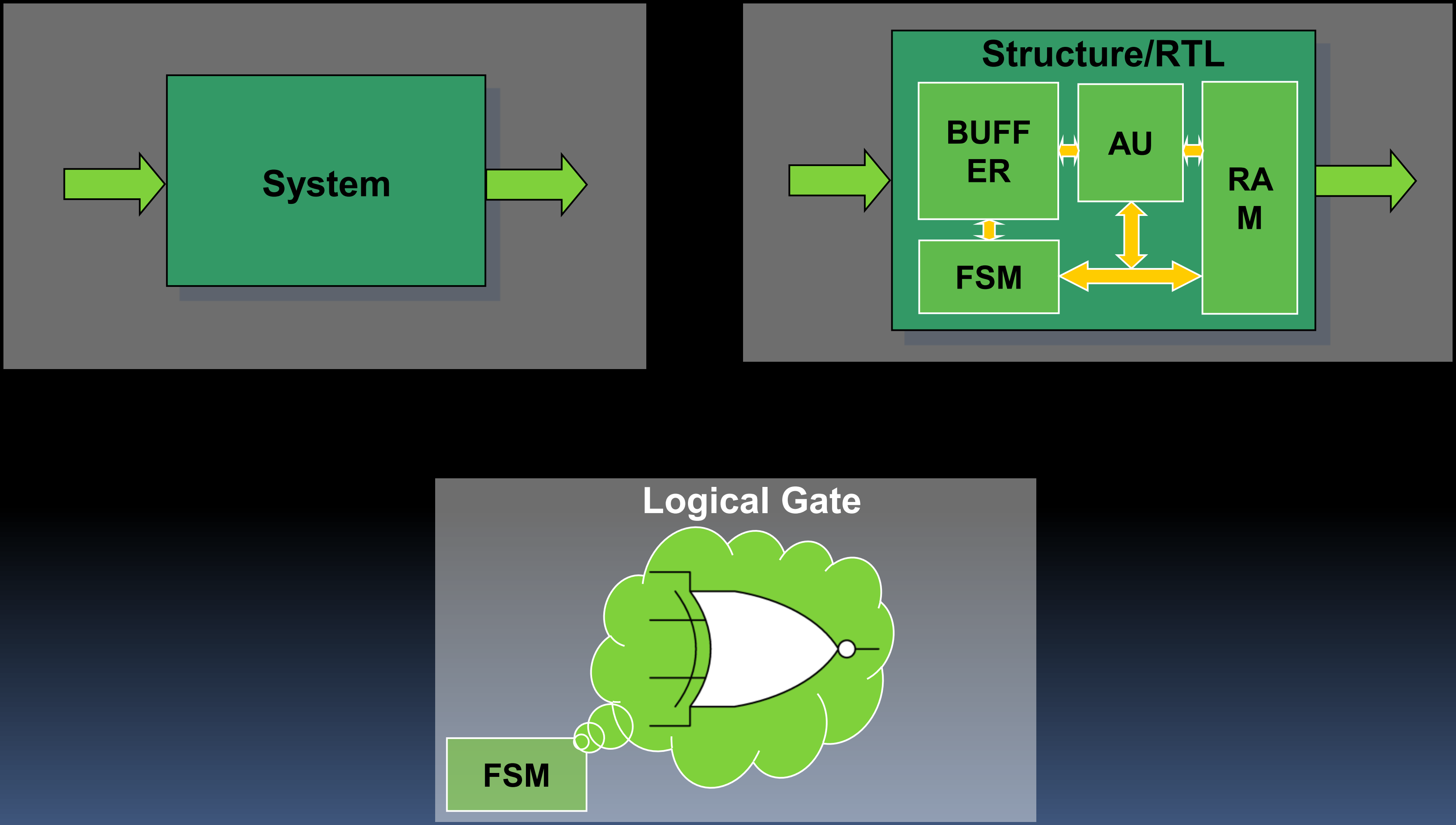
Description #8: Also the multiplexer can be described by using **structural VHDL**:

```
Library IEEE;
Use IEEE.std_logic_1164.all;
Entity MUX is
            port (
            A,B,S : in std_logic;
            Z     : out std_logic );
End MUX;
Architecture structural of MUX is
Begin
            U1: INV   port map (S, SN);
            U2: AND2 port map (A, SN, ASN);
            U3: AND2 port map (S, B, SB);
            U4: OR2   port map (ASN, SB, Z);
End structural;
```

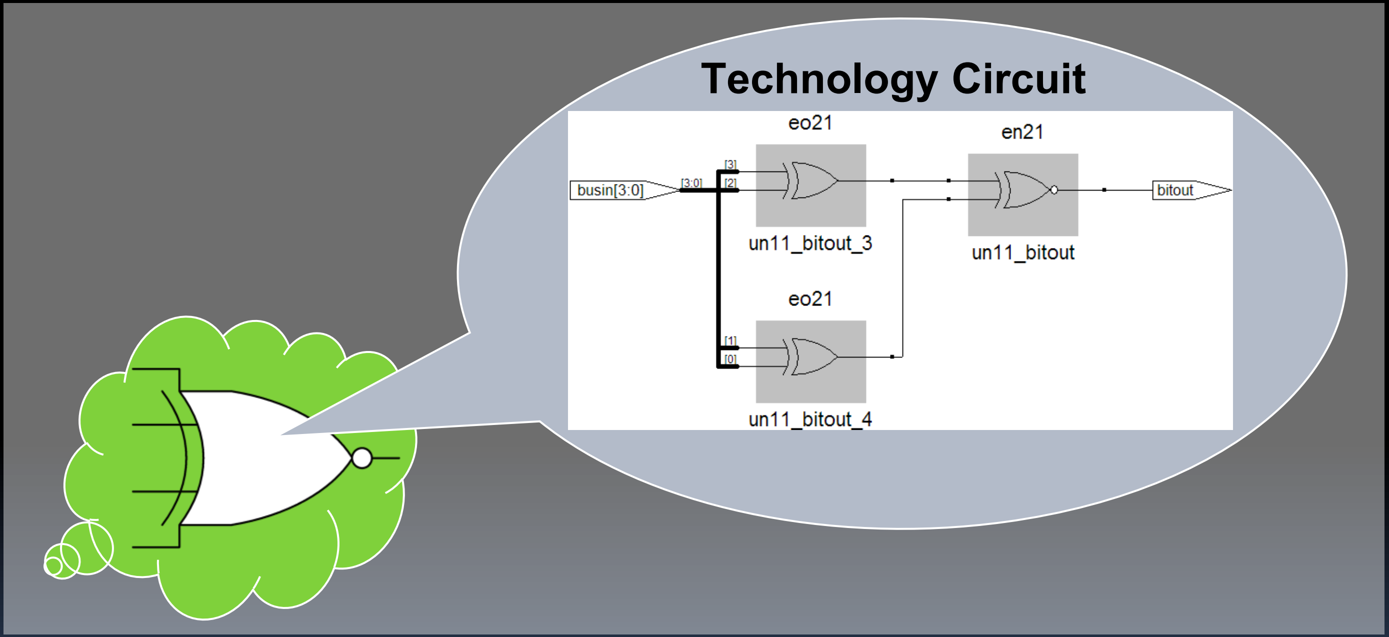
Digital Design Flow



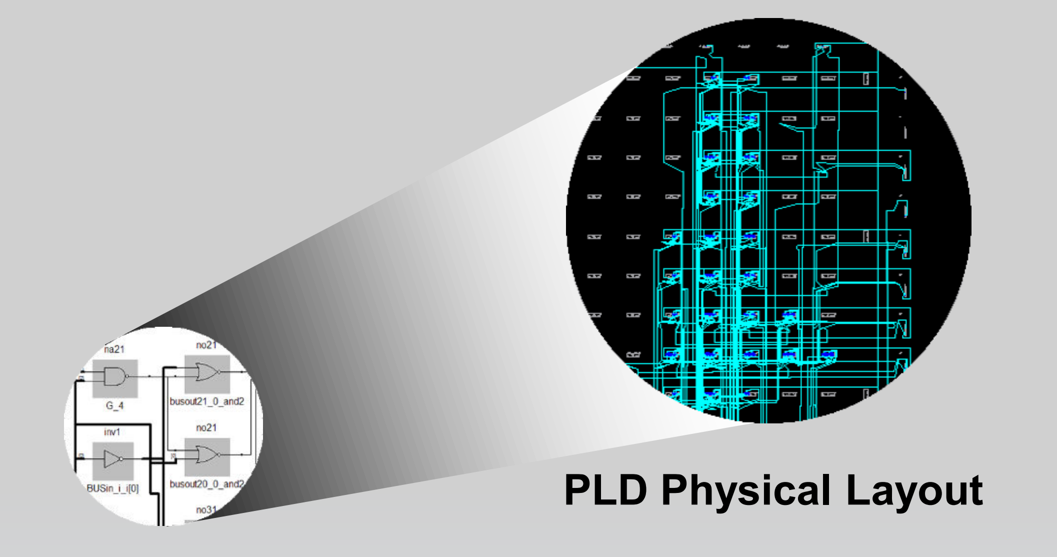
Digital Design Abstractions



Digital Design Abstractions

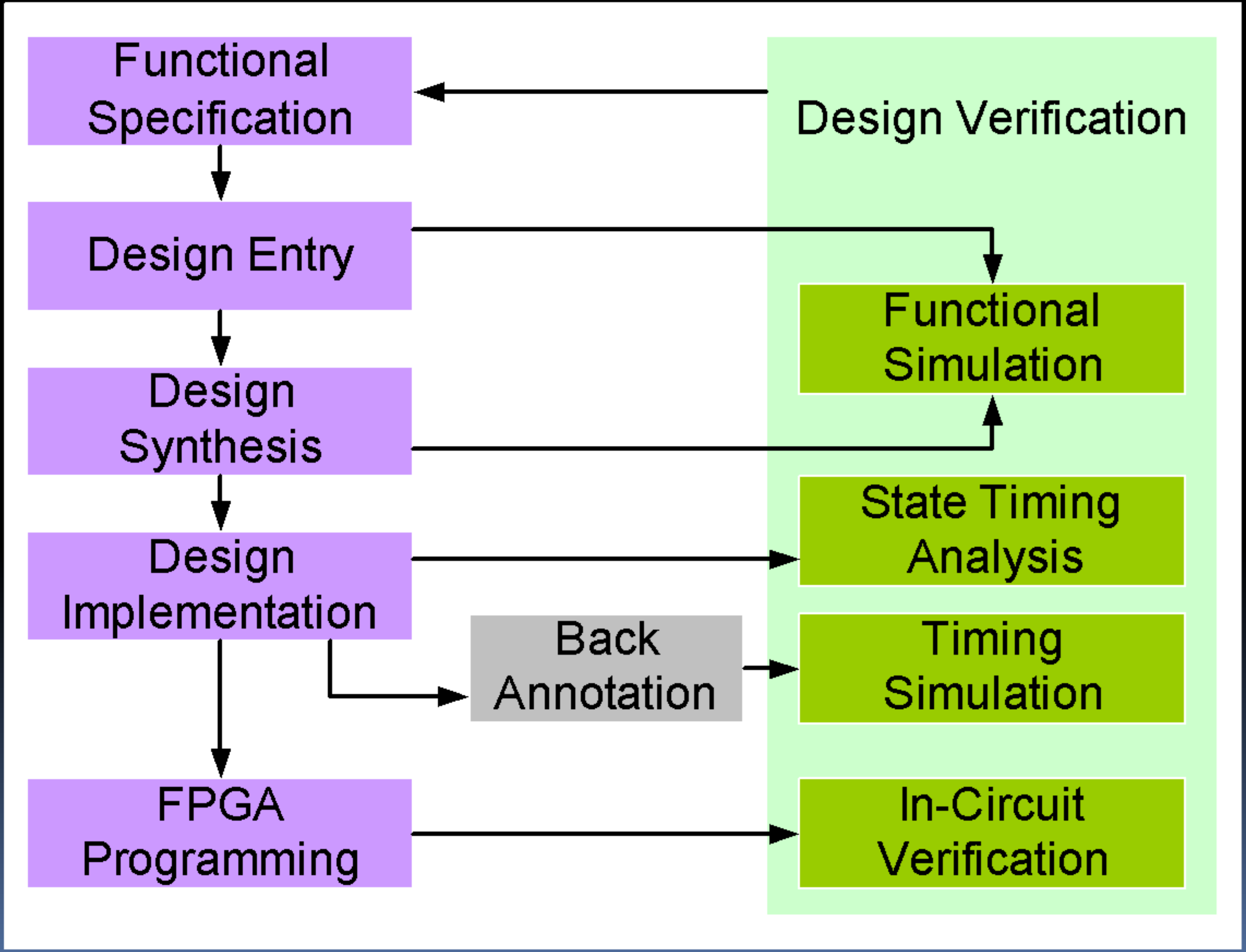


Digital Design Abstractions



PLD Physical Layout

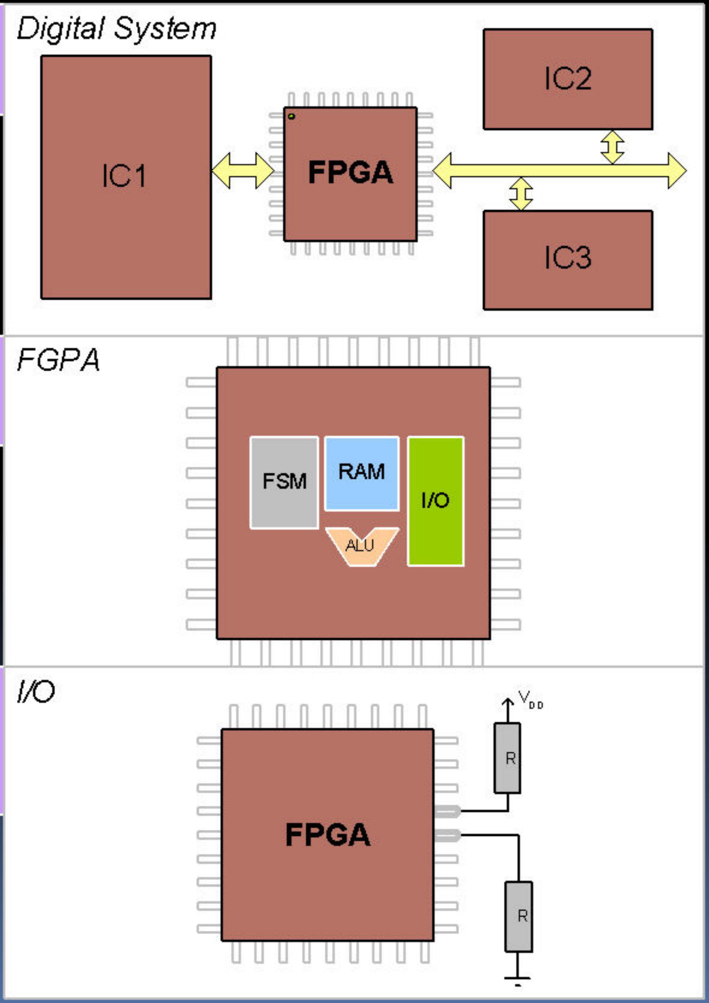
Digital Design Flow (FPGA-based)



Digital Design Flow (FPGA-based)

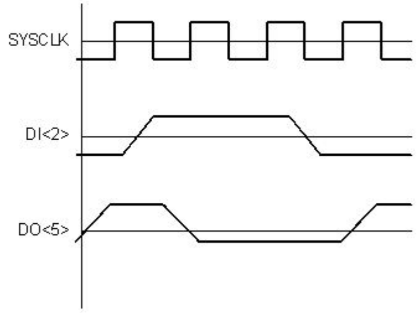
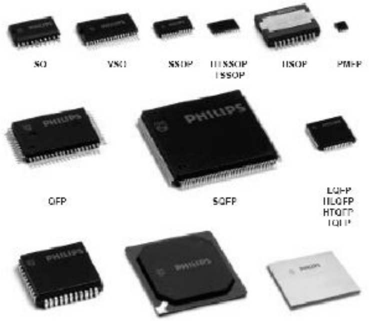
Functional Specification

1	**External Block Diagram** Shows how the FPGA fits into the system.
2	**Internal Block Diagram** Shows all major internal functional sections.
3	**Description of the I/O pins** -Output drive capability -Input threshold level (CMOS, TTL, ECL)



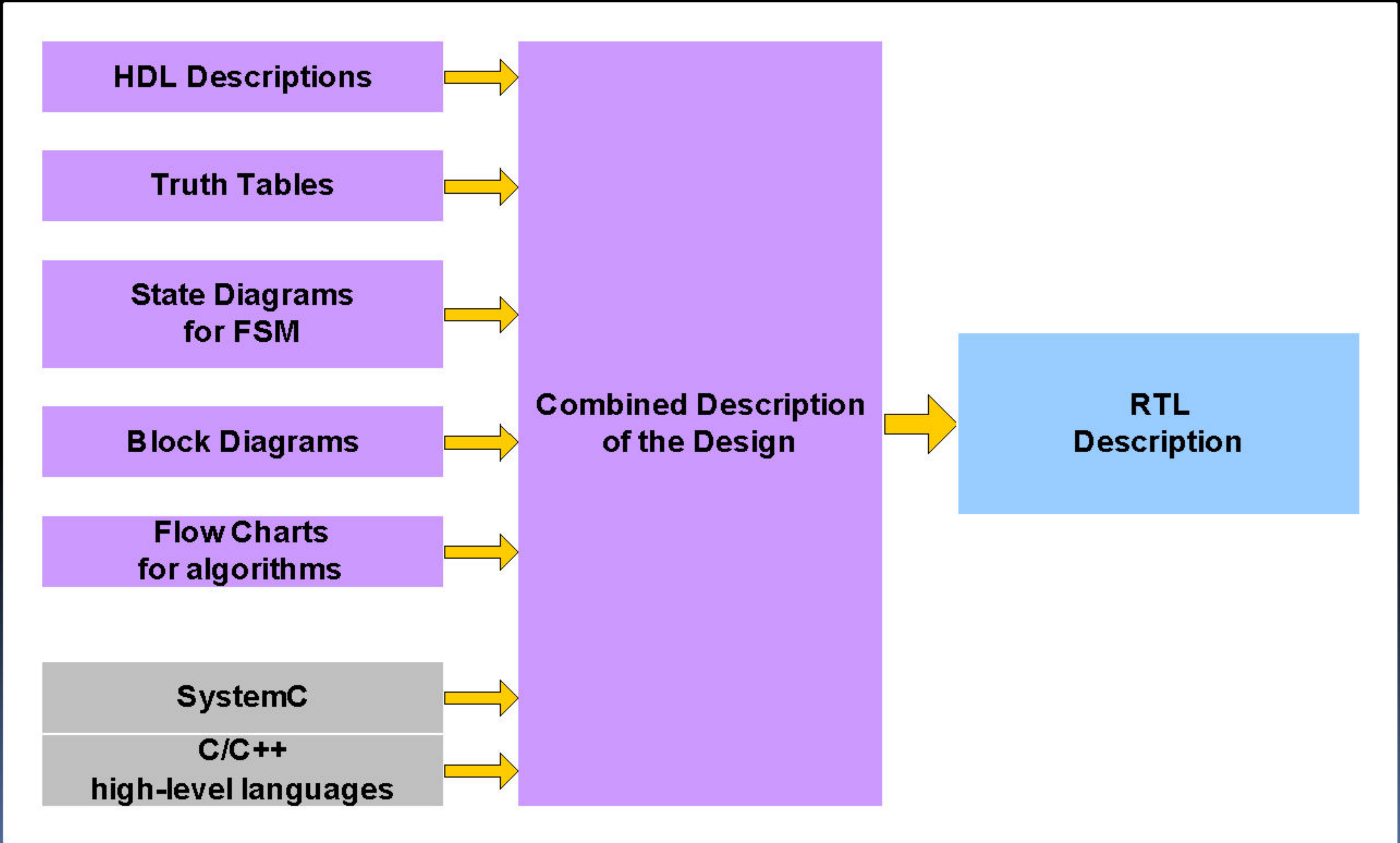
Digital Design Flow (FPGA-based)

Functional Specification

4	**Timing Estimates** -Setup and Hold times for inputs pins. -Propagation times for the output pins. -Clock cycle time.	*Timing* System clock frequency 100 MHz Setup times for inputs: DI<0-7> 10 ns. Propagation times for outputs: DO<0-15> 40 ns. 
5	**Package Type** Printed Circuit Board design depends on the FPGA package type: BGA, TQFP, TSOP etc.	*Package*  SO VSO SSOP HSSOP TSSOP HSOP PMFP QFP SQFP LQFP HLQFP HTQFP TQFP
6	**Power Consumption Target**	*Power Estimation Methodologies* *Estimation of switching activity* *Low Power Design*
7	**Test Methodologies**	*Build-In Self Testing* *Boundary-Scan Testing* *Self-Diagnostics and Self-Repair Techniques* *Design for Test*

Digital Design Flow (FPGA-based)

Design Entry



HDL Descriptions

Truth Tables

State Diagrams for FSM

Block Diagrams

Flow Charts for algorithms

SystemC

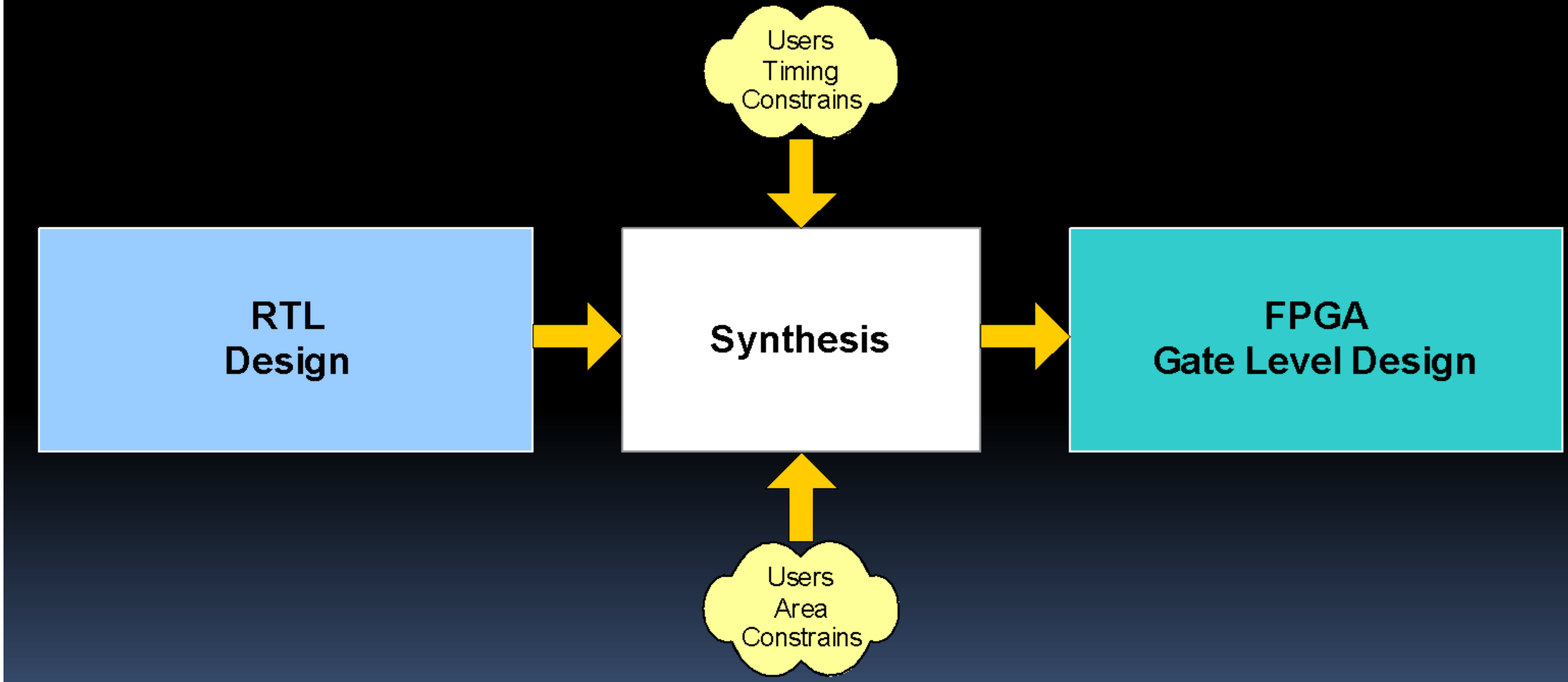
C/C++ high-level languages

Combined Description of the Design

RTL Description

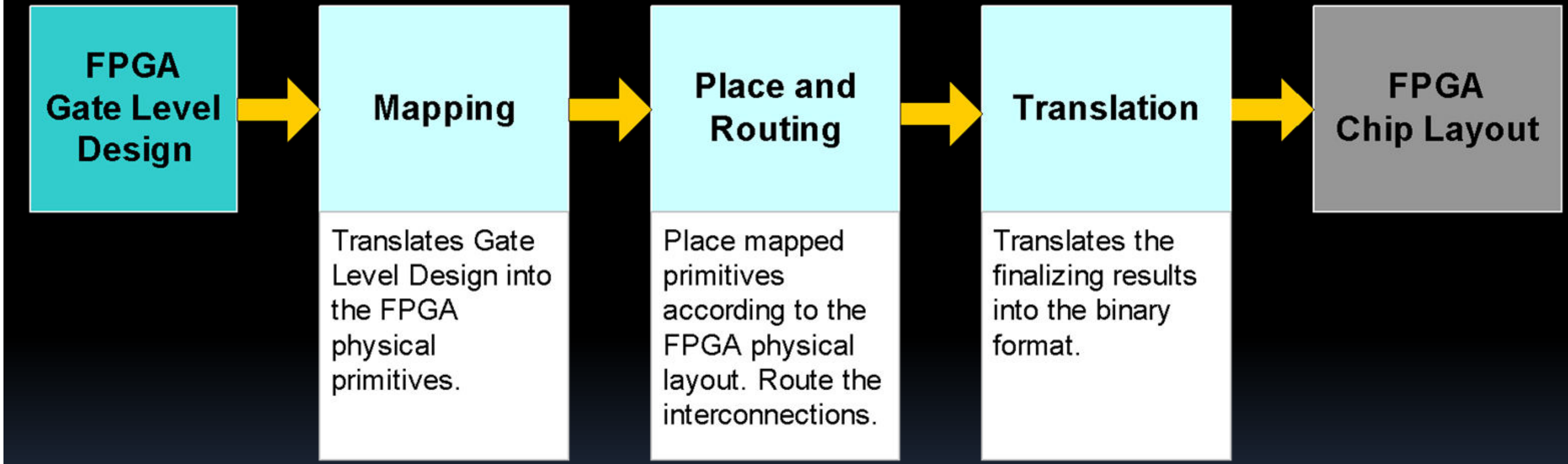
Digital Design Flow (FPGA-based)

Design Synthesis



Digital Design Flow (FPGA-based)

Design Implementation



Translates Gate Level Design into the FPGA physical primitives.

Place mapped primitives according to the FPGA physical layout. Route the interconnections.

Translates the finalizing results into the binary format.