



آزمایشگاه اجزای کامپیوتر

مدرس: محمد معین مهدویان شاد
سال تحصیلی ۱۴۰۰-۰۱



جلسه سوم

طراحی و پیاده‌سازی مدارات ترکیبی

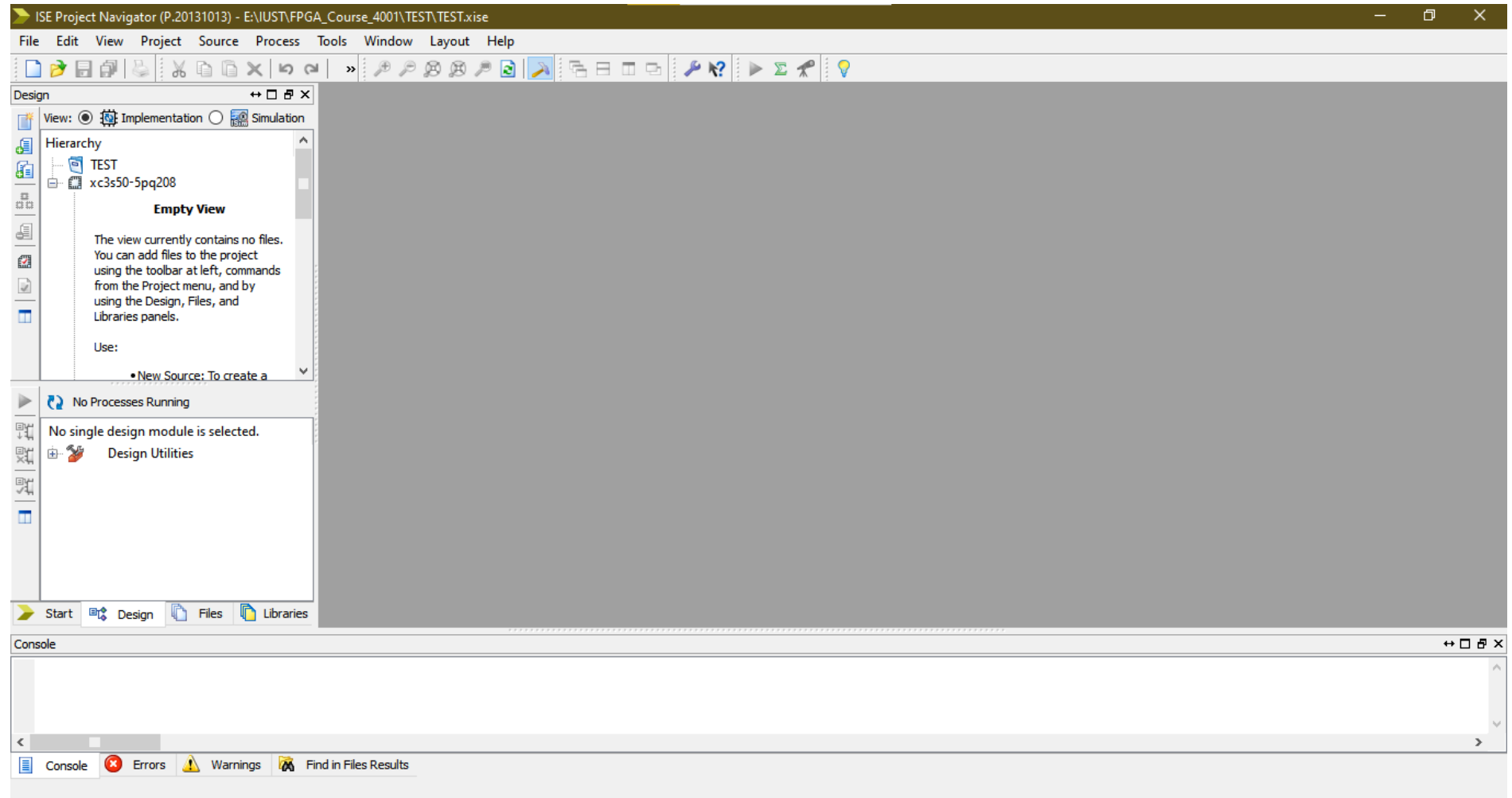


مقدمه

- آشنایی با محیط ISE و تشکیل پروژه جدید
- بررسی ساختار کد VHDL
- نوشتن اولین کد VHDL
- پیاده‌سازی یک مدار ماژولار در کد VHDL
- شبیه‌سازی و بررسی صحت عملکرد مدار



آشنایی با محیط ISE



ساختار کد VHDL




Libraries & Packages

Entity

Architecture

```
1  library IEEE;  
2  use IEEE.STANDARD.ALL;  
3  use IEEE.STD_LOGIC_1164.ALL;  
4  use IEEE.STD_LOGIC_ARITH.ALL;  
5  use IEEE.STD_LOGIC_UNSIGNED.ALL;  
6  use IEEE.STD_LOGIC_SIGNED.ALL;  
7  
8  
9  
10  
11
```



```
variable counter : integer;  
variable temp : BIT;  
begin  
    if (load = '1') then  
        temp := din;  
        dout <= din(7);  
        counter := 0;  
        temp := '0';  
    elsif (clk'event and clk = '1' and temp = '0') then  
        counter := counter + 1;  
        dout <= din(counter);  
        if (counter = 7) then  
            counter := 0;  
            temp := '1';  
        end if;  
    end if;  
end process;  
end Behavioral;
```



ساختار کد VHDL

Libraries & Packages

Entity

Architecture

INTEGER

$-2^{31}-1 \sim 2^{31}$

BOOLEAN

False, True

STD_LOGIC

'U', 'X', '0', '1', 'Z', 'W', 'L', 'H', '-'

STD_LOGIC_Vector

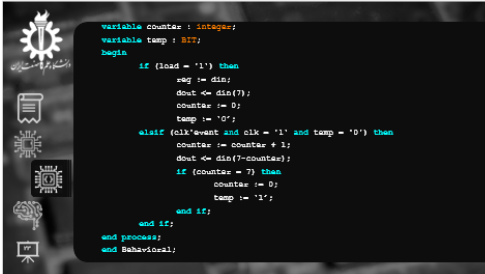
UNSIGNED

SIGNED

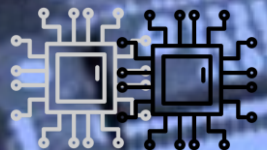
BIT

'0', '1'

BIT_VECTOR



```
variable counter : integer;  
variable temp : BIT;  
begin  
    if (load = '1') then  
        temp := din;  
        dout <= din(7);  
        counter := 0;  
        temp := '0';  
    elsif (clk'event and clk = '1' and temp = '0') then  
        counter := counter + 1;  
        dout <= din(counter);  
        if (counter = 7) then  
            counter := 0;  
            temp := '1';  
        end if;  
    end if;  
end process;  
end Behavioral;
```



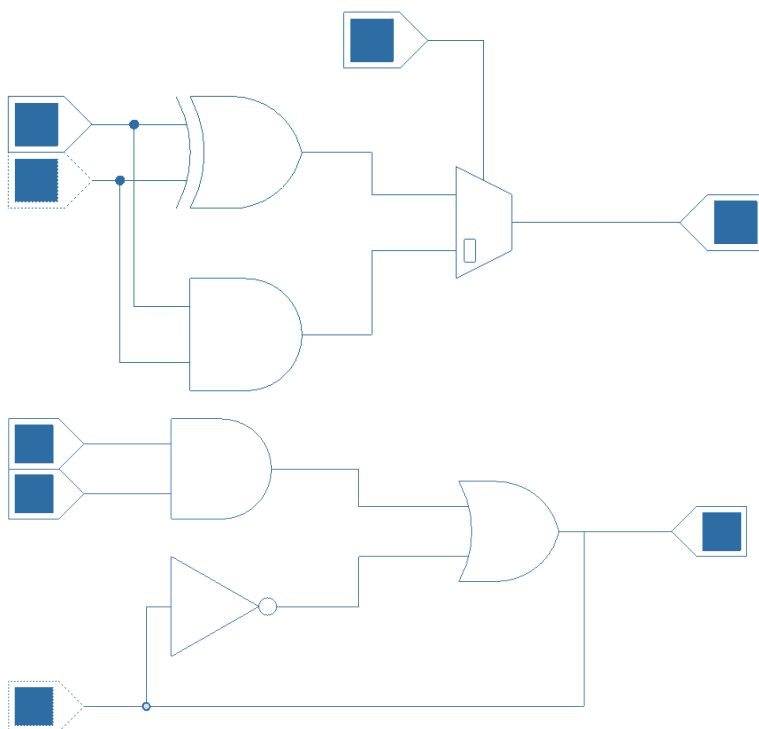
Architecture

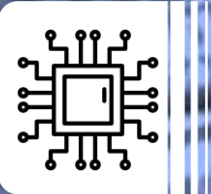
ترکیبی

- خروجی‌های مدار تنها به ورودی‌های همان لحظه وابسته است.

ترتیبی

- خروجی‌های مدار علاوه بر ورودی‌های همان لحظه، به ورودی لحظات قبل نیز وابسته است.





Architecture

```
1 architecture Behavioral of main vhd1 is
```

تعریف سیگنال‌ها، کامپوننت‌ها و ...

```
2  
3  
4  
5 begin
```

بدنه کد ترکیبی

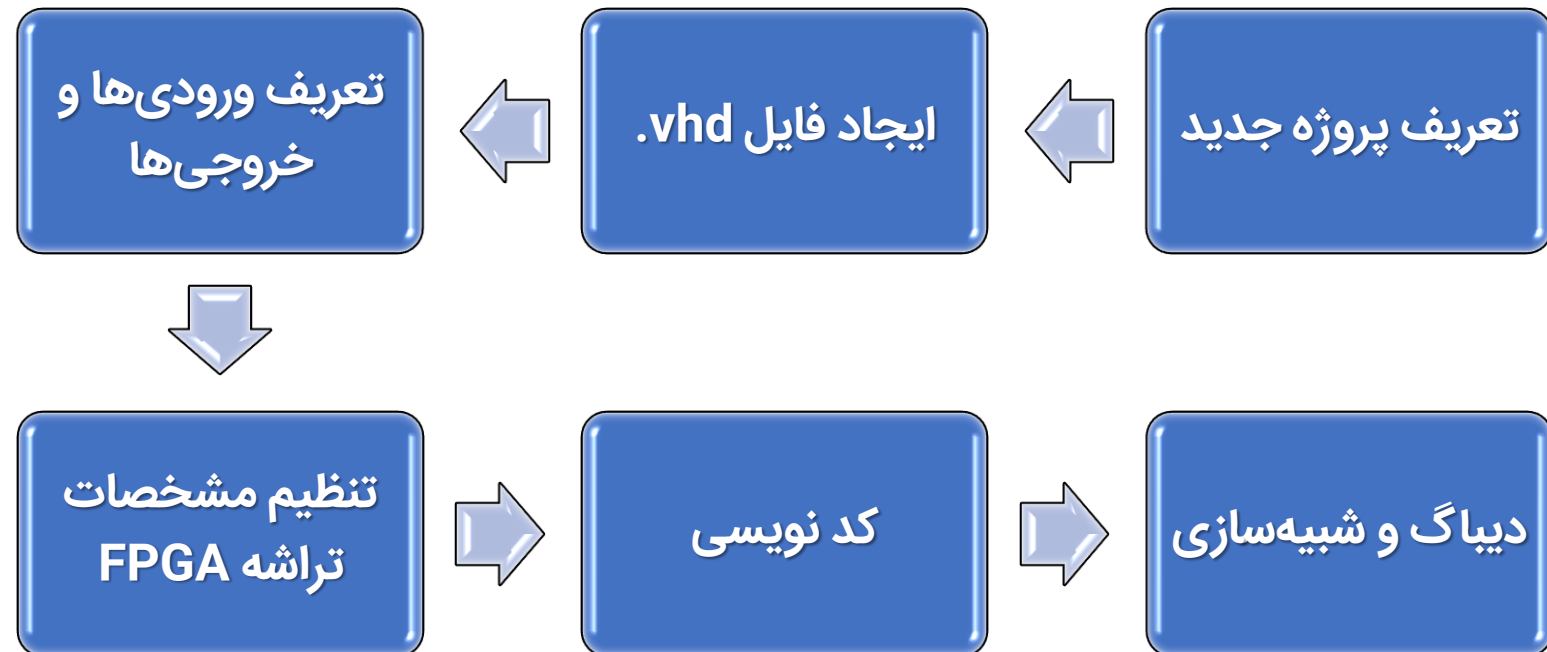
```
6  
7  
8  
9  
10 process ( )  
11 begin
```

بدنه کد ترتیبی

```
12  
13  
14  
15 end process;
```

```
16  
17  
18 end Behavioral;  
19
```

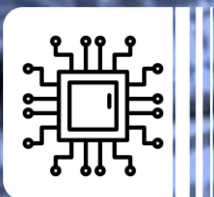
روند کار در محیط ISE



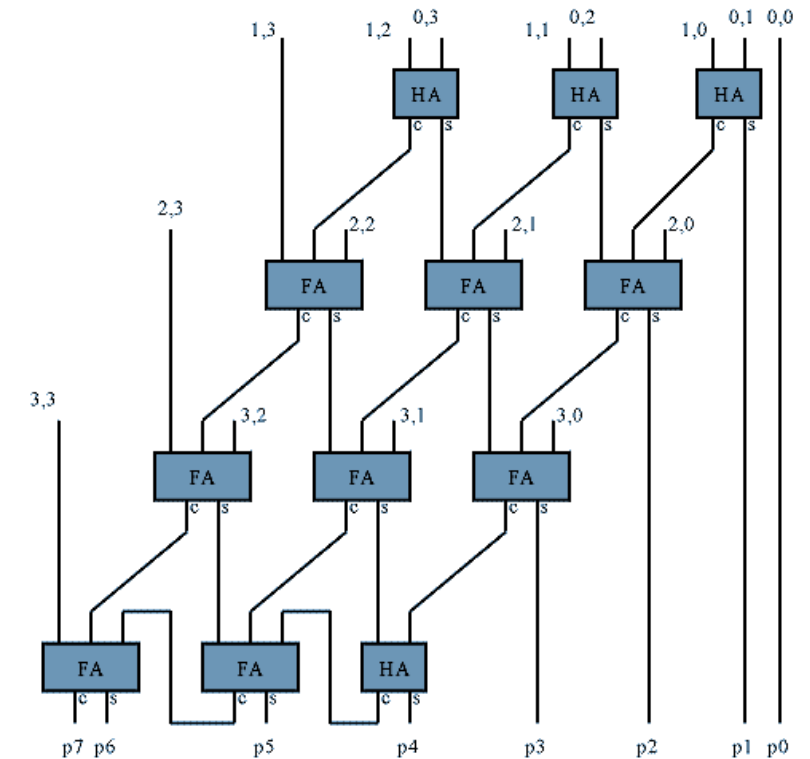
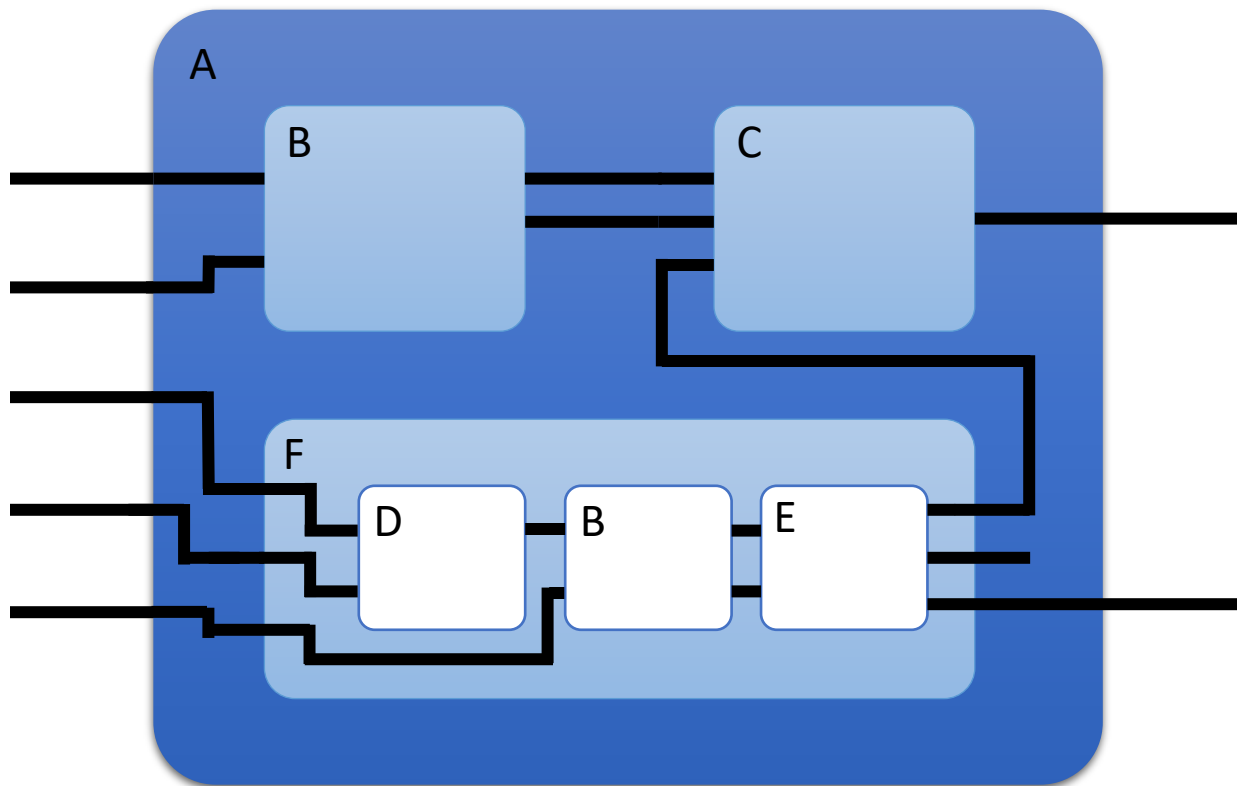
طراحی مدارات به صورت ماژولار

استفاده از ماژول‌هایی که از قبل ایجاد کرده‌ایم برای ساخت مدارات بزرگ‌تر:

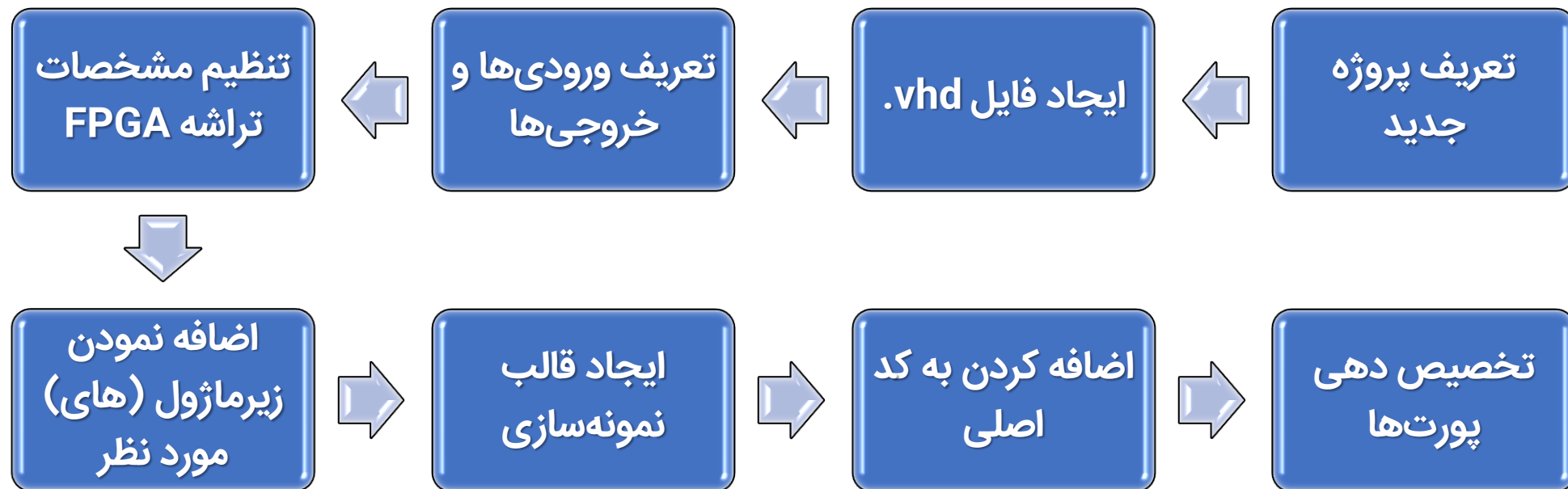
- خلاصه شدن کد
- ساده‌تر شدن روند طراحی مدار
- سهولت در عیب‌یابی کد
- تطبیق‌پذیری بیشتر مدار

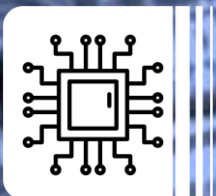


طراحی مدارات به صورت ماژولار



طراحی مدارات به صورت ماژولار



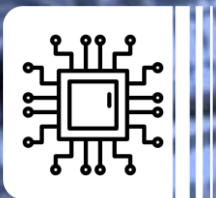


شبیه سازی

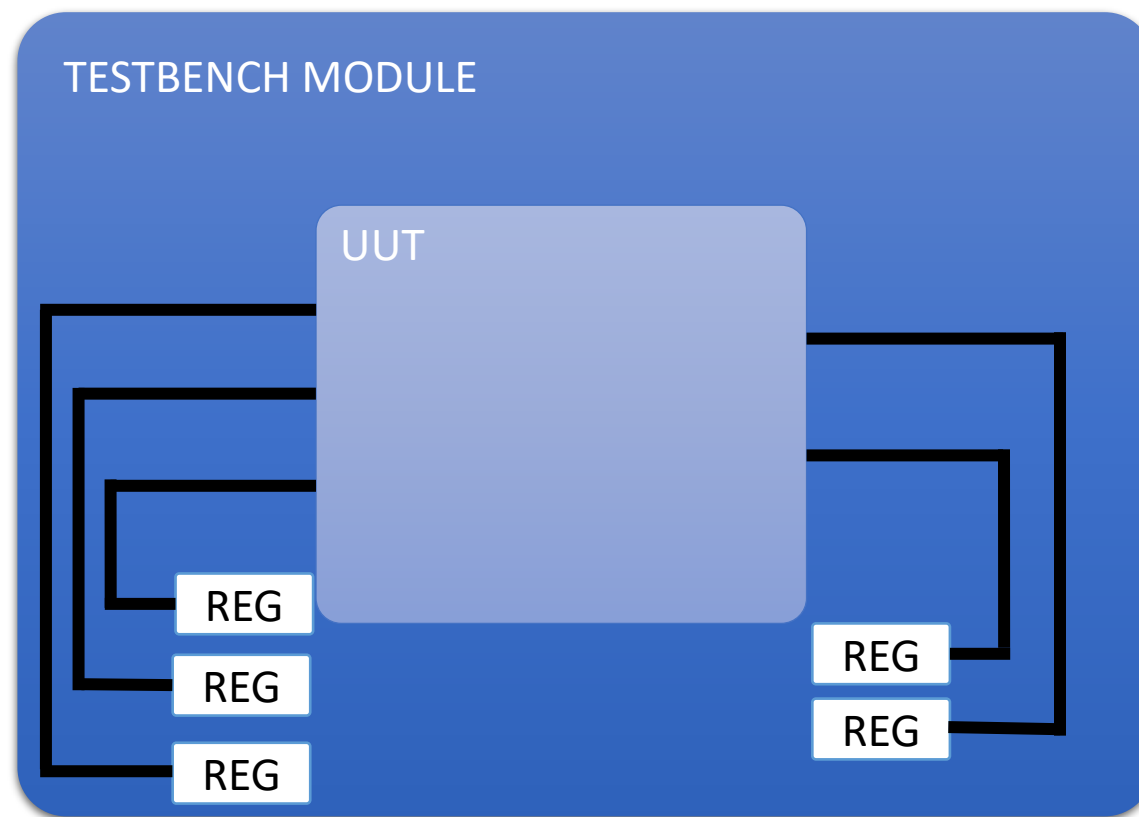
• اطمینان از صحت عملکرد مدار

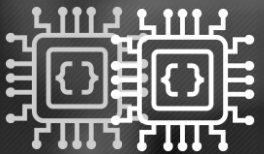
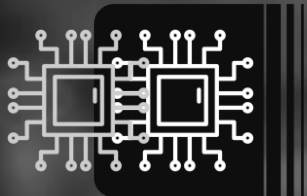


Model*Sim*



نوشتن کد TESTBENCH





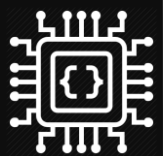
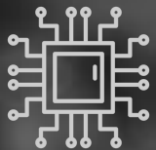
```
LIBRARY ieee;
USE ieee.std_logic_1164.ALL;

ENTITY test IS
END test;

ARCHITECTURE behavior OF test IS

    -- Component Declaration for the Unit Under Test (UUT)
    COMPONENT FA_1
    PORT (
        A : IN std_logic;
        B : IN std_logic;
        Cin : IN std_logic;
        S : OUT std_logic;
        Cout : OUT std_logic );
    END COMPONENT;

    --Inputs
    signal A : std_logic := '0';
    signal B : std_logic := '0';
    signal Cin : std_logic := '0';
```



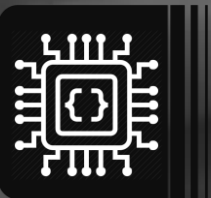
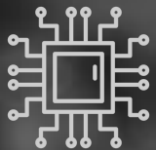
BEGIN

```
--Inputs
signal A : std_logic := '0';
signal B : std_logic := '0';
signal Cin : std_logic := '0';

--Outputs
signal S : std_logic;
signal Cout : std_logic;

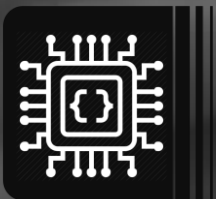
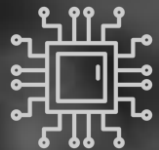
-- No clocks detected in port list. Replace <clock> below with
-- appropriate port name
constant <clock>_period : time := 10 ns;

-- Instantiate the Unit Under Test (UUT)
 uut: FA_1 PORT MAP (
    A => A,
    B => B,
    Cin => Cin,
    S => S,
    Cout => Cout );
```

```
-- Clock process definitions
<clock>_process :process
begin
    <clock> <= '0';
    wait for <clock>_period/2;
    <clock> <= '1';
    wait for <clock>_period/2;
end process;

-- Stimulus process
stim_proc: process
begin
    -- hold reset state for 100 ns.
    wait for 100 ns;
    wait for <clock>_period*10;
    -- insert stimulus here
    A <= '1';
    B <= '1';
    Cin <= '1';
```



```
begin
    -- hold reset state for 100 ns.
    wait for 100 ns;
    wait for <clock>_period*10;
    -- insert stimulus here
    A <= '1';
    B <= '1';
    Cin <= '1';

    wait;
end process;

END;
```

دستور کار آزمایش امروز؟

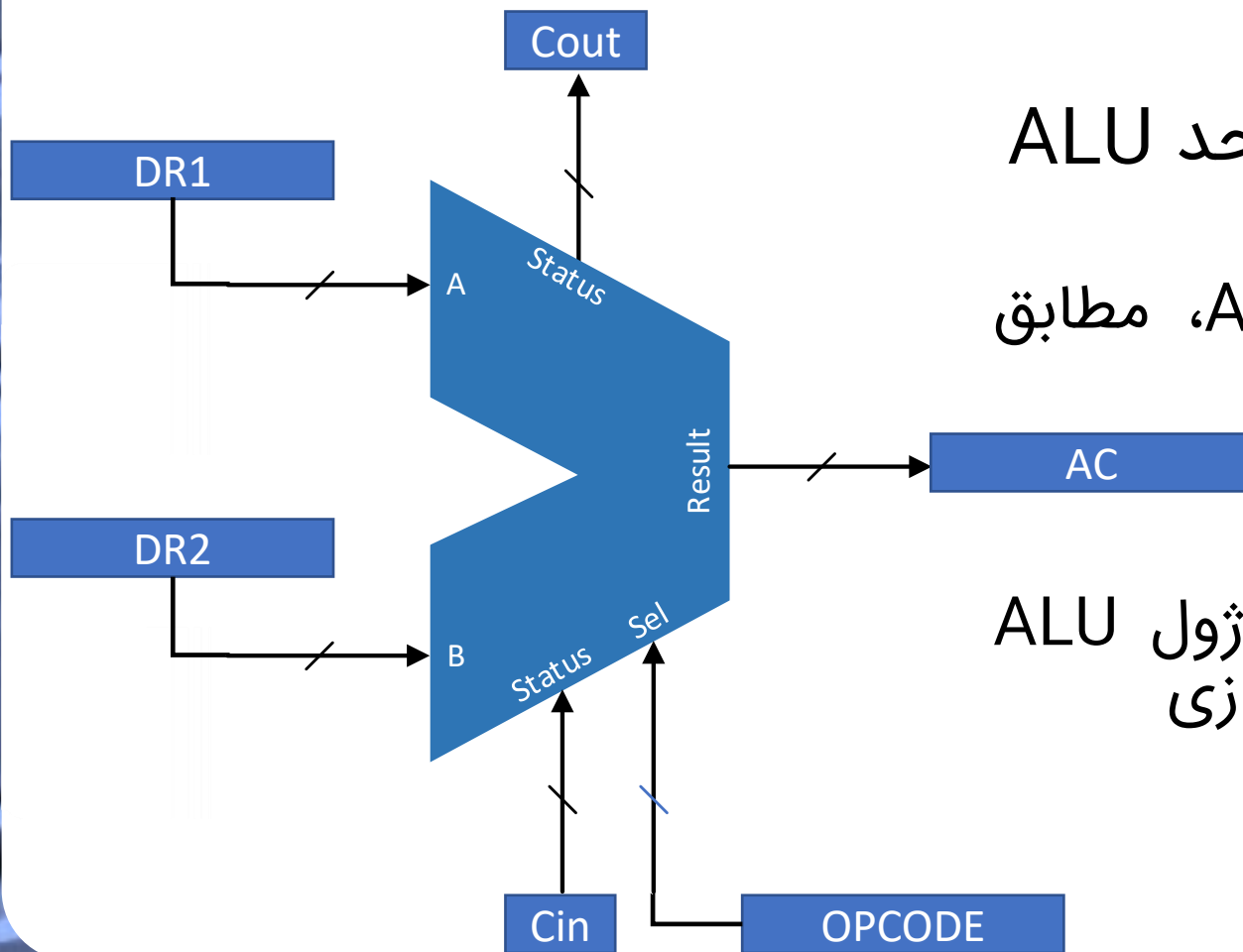
• طراحی و پیاده‌سازی یک واحد ALU

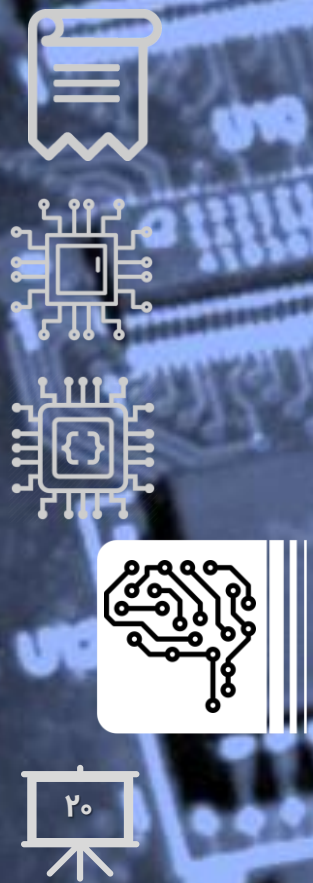
• طراحی تمام واحدهای ALU، مطابق اسلاید بعد

• کد نویسی هر ماژول

• شبیه‌سازی همه ماژول‌ها

• طراحی و پیاده‌سازی تاپ‌ماژول ALU به همراه کدنویسی و شبیه‌سازی





Selector (OPCODE)					عمل	تابع
S_3	S_2	S_1	S_0	Cin		
0	0	0	0	0	$F = A$	انتقال
0	0	0	0	1	$F = A + 1$	افزایش
0	0	0	1	0	$F = A + B$	جمع
0	0	0	1	1	$F = A + B + 1$	Cin جمع با
0	0	1	0	0	$F = A + \bar{B}$	B تفریق با
0	0	1	0	1	$F = A + \bar{B} + 1$	تفریق
0	0	1	1	1	$F = A - 1$	A کاهش
0	1	0	0	X	$F = A \wedge B$	AND
0	1	0	1	X	$F = A \vee B$	OR
0	1	1	0	X	$F = A \oplus B$	XOR
0	1	1	1	X	$F = \bar{A}$	متمم A
1	0	X	X	X	$F = \text{shr } A$	شیفت A به راست
1	1	X	X	X	$F = \text{shl } A$	شیفت A به چپ

از توجه شما سپاسگزارم.