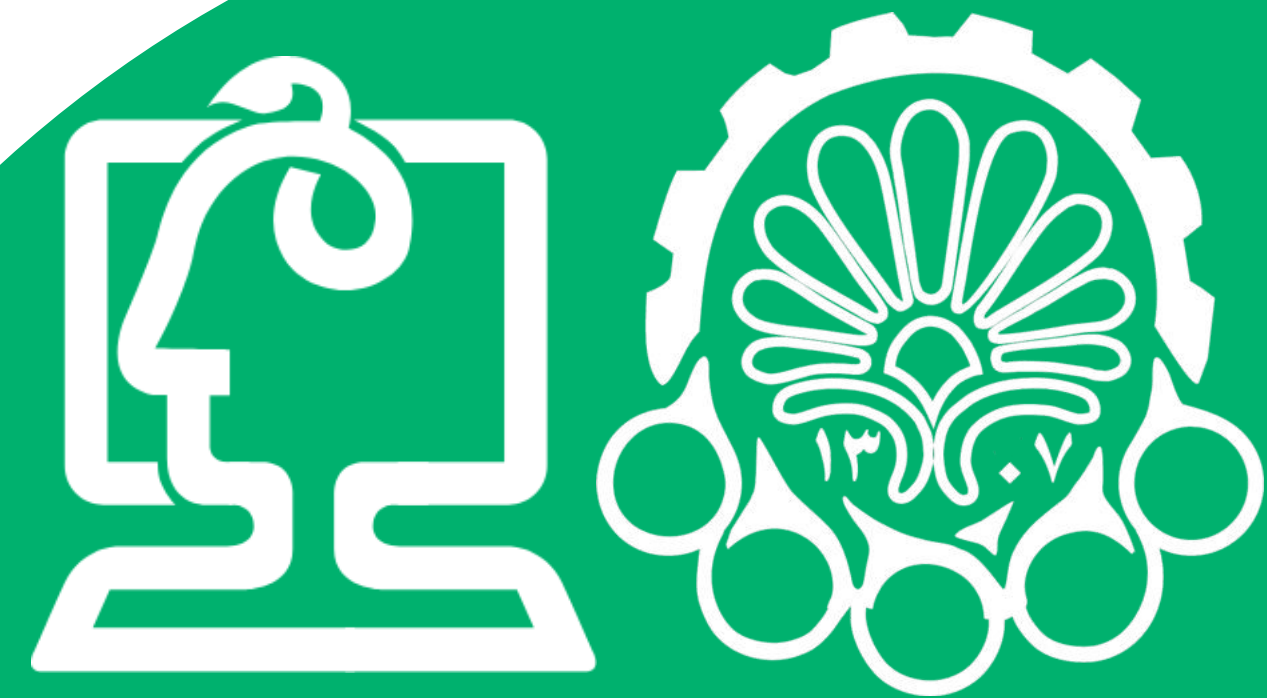


آزمایشگاه معماری کامپیوتر

نیمسال اول ۱۴۰۰

جلسه دوم





توصیف گیت And به صورت برداري

```
library IEEE;  
use IEEE.std_logic_1164.all;  
Entity and_gate_vector is  
Port(  
  A, B: in std_logic_vector(3 downto 0);  
  C : out std_logic_vector(3 downto 0)  
);  
End entity and_gate_vector;
```



مقدار دهی به std_logic_vector

```
Port(  
  A: in std_logic_vector(9 downto 0);  
  C: out std_logic_vector (9 doento 0)  
);  
---  
Signal b: std_logic_vector(9 downto 0);  
---  
B(9 downto 1) <= B(8 downto 0); -- shift  
B(0) <= '1';  
B(2 downto 0) <= "110";  
B <= A;  
B(9 downto 8) <= A(2 downto 1);  
B(4) <= B(5);  
B <= (others => '0');  
B(5 downto 0) <= A(9 downto 7) & "011";
```



استفاده از when-else و with-select

```
Signal q: std_logic_vector(3 downto 0);  
Signal t: std_logic_vector(3 downto 0);
```

```
Q<= "0000" when t="1010" else  
    "1011" when t="0010" else  
    "1111";
```

```
with t select  
q<= "0000" when "1010",  
    "1011" when "0010",  
    "1111" when others;
```



```
library ieee;  
use ieee.std_logic_1164.all;
```

```
entity Decoder_vec is  
port(  
input: in std_logic_vector(2 downto 0);  
output: out std_logic_vector(7 downto 0)  
);  
end entity Decoder_vec;
```

architecture beh of Decoder_vec is

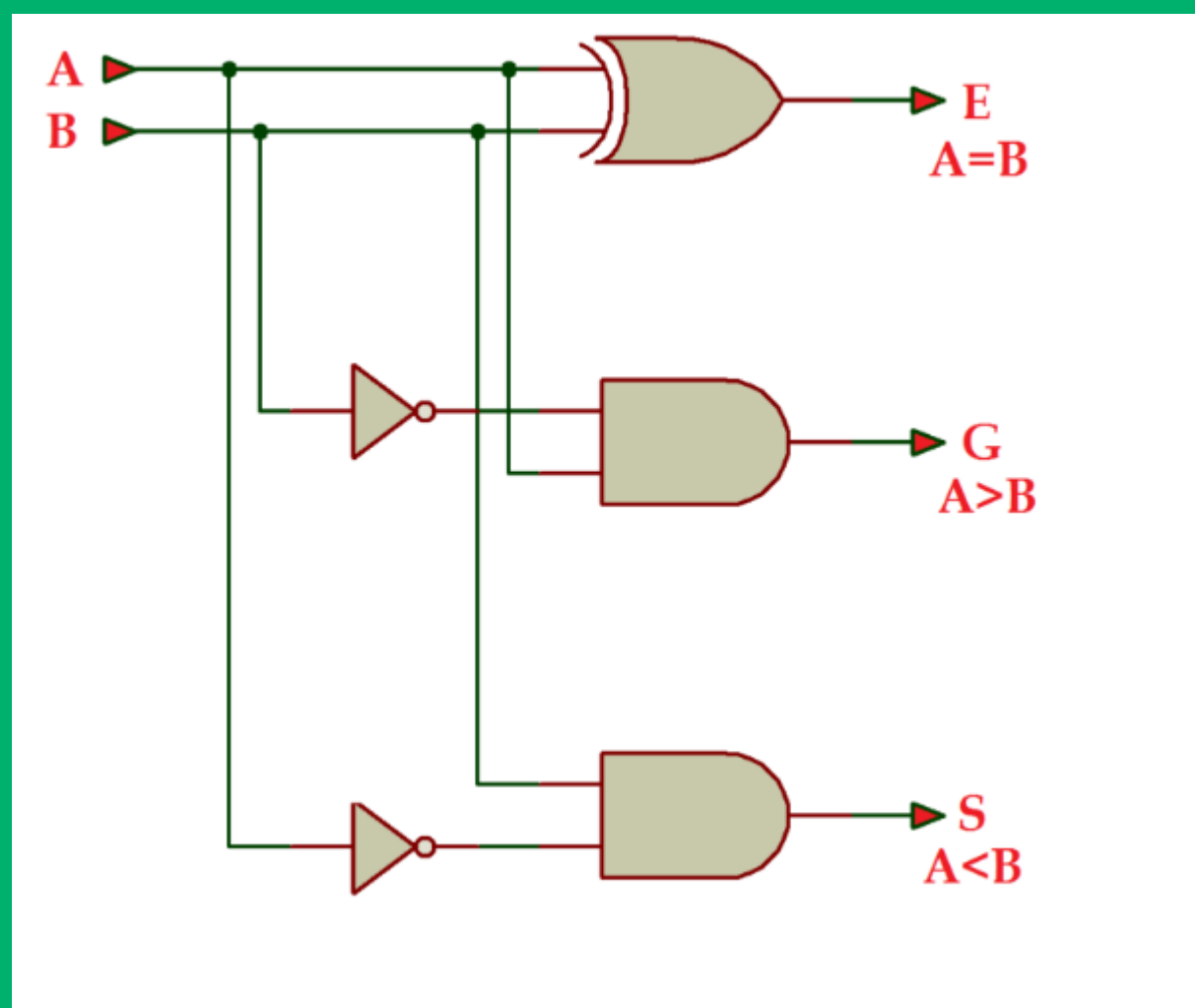
```
begin  
output(0) <= '1' when input="000" else '0';  
output(1) <= '1' when input="001" else '0';  
output(2) <= '1' when input="010" else '0';  
output(3) <= '1' when input="011" else '0';  
output(4) <= '1' when input="100" else '0';  
output(5) <= '1' when input="101" else '0';  
output(6) <= '1' when input="110" else '0';  
output(7) <= '1' when input="111" else '0';  
end beh;
```



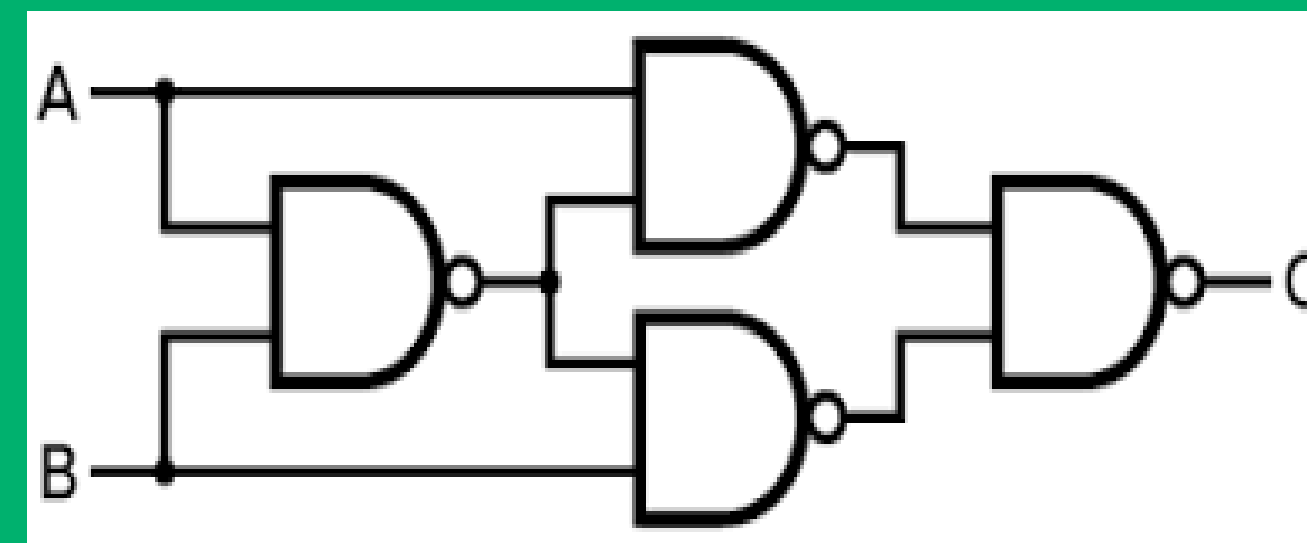
تکلیف شماره دو

قسمت اول:

طراحی مدار مقایسه کننده تک بیتی و استفاده از گیت xor که با گیت های nand پیاده سازی شده



مدار مقایسه کننده تک بیتی



گیت xor پیاده سازی شده با گیت های nand



تکلیف شماره دو

قسمت دوم:

با استفاده از یک $Mux4*1$ و حداقل تعداد گیت دو عدد دوبیتی ($A=a_1a_0, B=b_1b_0$) را مقایسه کرده و اگر $A>B$ خروجی یک شود.



تکلیف شماره دو

قسمت سوم:

با استفاده از دیکودر ۲ به ۴ و گیت‌های and و or یک مالتی پلکسر ۴ به یک طراحی و پیاده سازی کنید.

با استفاده از دیکودر و گیت‌های nand یک جمع کننده بسازید.