

# ازمایش اول

موضوع: جمع کننده تک بیتی و

نمایش عدد روی 7segment

تاریخ آزمایش: ۱۴۰۲/۷/۱۹

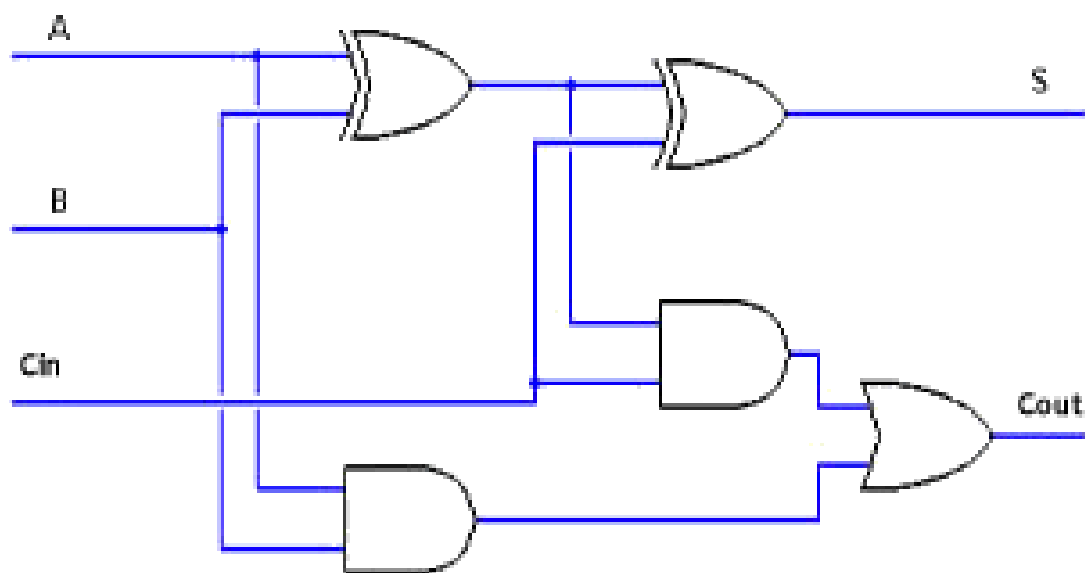
استاد: دکتر جوادی

جواد فرجی (۹۹۵۲۲۰۰۵)

محمد رحمانی (۹۷۵۲۱۲۸۸)

## آزمایش الف:

این آزمایش به صورت دستی انجام شده و با مشخص کردن ورودی ها و خروجی ها و مپ کردن این خروجی و ورودی به پایه های موجود در برد fpga انجام شده



برای مپ کردن ورودی ها و خروجی ها نیاز به اضافه کردن فایل ucf به پروژه داریم.

## اجرای برنامه بر روی برد fpga:

1. Synthesize
2. Implement design
3. Generate programming

در این سه مرحله گزینه run را میزنیم و در صورتی که مشکل خاصی در برنامه وجود نداشته باشد و به باگ نخوریم به مرحله بعد می‌رویم.

## 4. Impact

با استفاده از این برنامه، فایل باینری ساخته شده را به programmer انتقال می‌دهیم و programmer این برنامه را روی بردهای fpga اجرا میکند.

آزمایش ب:

ورودی و خروجی:

```
seg_btn : in std_logic_vector (3 downto 0);  
seg_data : out std_logic_vector (6 downto 0));  
seg_sel : out bit;
```

1 بیت برای این که کدام یک از 7seg ها روشن شود. ۷ بیت برای دیتایی که قرار است در 7seg نشان داده شود و ۴ بیت برای این که با کلید ها، عدد موجود روی 7seg را مشخص کنیم.

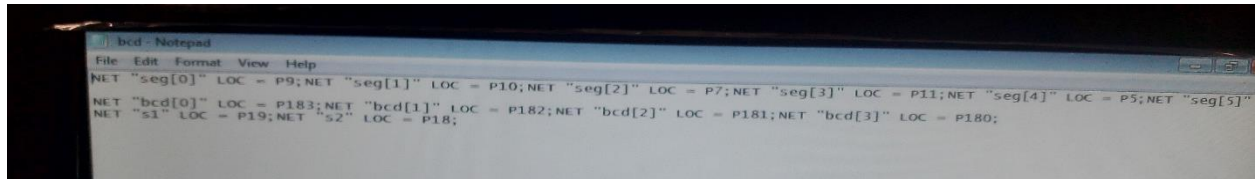
نمایش اطلاعات: (process)

یک پروسه برای این که هر کدام از حالت هایی که کلید ها به وجود می آورند، چه عددی نشان بدهند.

```
process(seg_btn)  
begin  
    case seg_btn is  
        when "0001" => seg <= "0000110"; -- Number 1  
        when "0010" => seg <= "1011011"; -- Number 2  
        when "0011" => seg <= "1001111"; -- Number 3  
        when "0100" => seg <= "1100110"; -- Number 4  
        when "0101" => seg <= "1101101"; -- Number 5  
        when "0110" => seg <= "1111101"; -- Number 6  
        when "0111" => seg <= "0000111"; -- Number 7  
        when "1000" => seg <= "1111111"; -- Number 8  
        when "1001" => seg <= "1101111"; -- Number 9  
        when others => seg <= "0000000";  
    end case;  
end process;
```

## مپ کردن خروجی ها به روی fpga:

برای مپ کردن روی برد های fpga، با استفاده از داکيومنت موجود، این خطوط را داخل فایل ucf قرار میدهیم:



```
File Edit Format View Help
NET "seg[0]" LOC = P9; NET "seg[1]" LOC = P10; NET "seg[2]" LOC = P7; NET "seg[3]" LOC = P11; NET "seg[4]" LOC = P5; NET "seg[5]" LOC = P12;
NET "bcd[0]" LOC = P183; NET "bcd[1]" LOC = P182; NET "bcd[2]" LOC = P181; NET "bcd[3]" LOC = P180;
NET "s1" LOC = P19; NET "s2" LOC = P18;
```

## اجرای برنامه بر روی برد fpga:

1. Synthesize
2. Implement design
3. Generate programming

در این سه مرحله گزینه run را میزنیم و در صورتی که مشکل خاصی در برنامه وجود نداشته باشد و به باگ نخوریم به مرحله بعد می‌رویم.

## 4. Impact

با استفاده از این برنامه، فایل باینری ساخته شده را به programmer انتقال می‌دهیم و programmer این برنامه را روی بردهای fpga اجرا میکند.