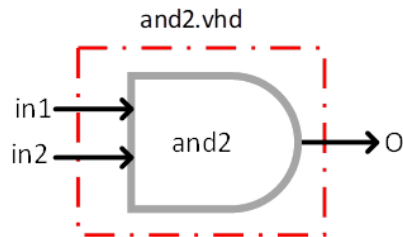


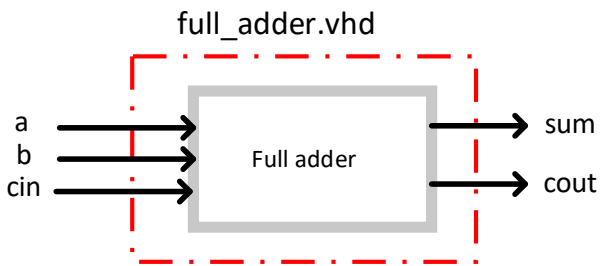
توضیحات آزمایش پنجم

هدف از این این آزمایش طراحی انواع ضرب‌کننده و تحلیل و مقایسه عملکرد آنها است. پیاده‌سازی سخت‌افزاری تمامی واحدها در سطح تجرید گیت صورت گیرد. برای طراحی ضرب‌کننده از ماژول‌های half_adder و full_adder و ماژول گیت and استفاده کنید. نمودار بلوکی انواع ضرب‌کننده در دستور کار آورده شده است.

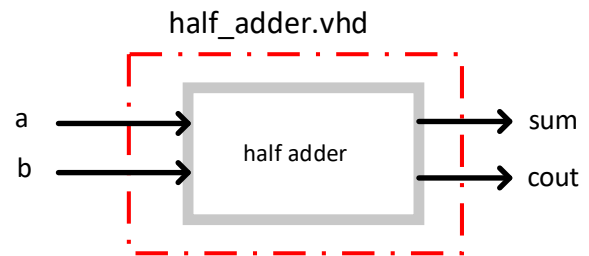
نام entity و پورت‌های ورودی و خروجی ماژول‌های مورد نظر به صورت زیر باشد. تهیه testbench الزامی است.



شکل (۱) ماژول گیت and دو ورودی و پورت‌های ورودی و خروجی



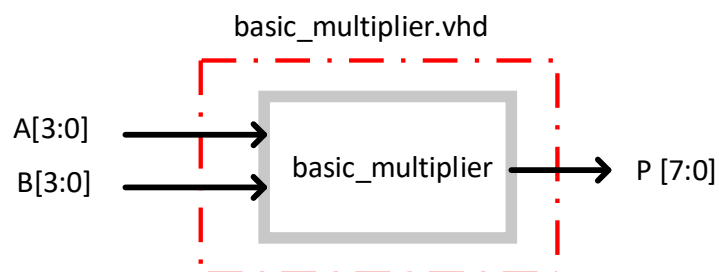
شکل (۳) ماژول full_adder و پورت‌های ورودی و خروجی



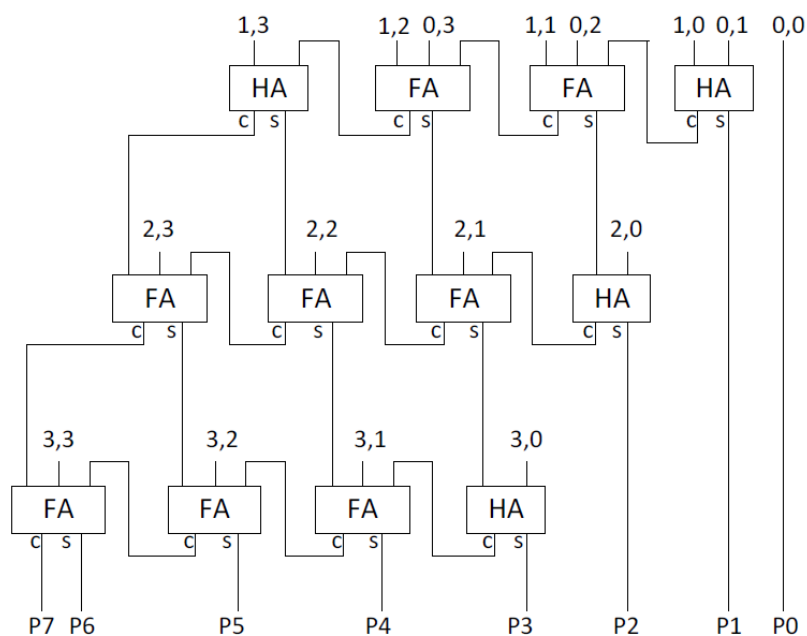
شکل (۲) ماژول half_adder و پورت‌های ورودی و خروجی

(۱) ضرب‌کننده معمولی

برای طراحی این ضرب‌کننده از ماژول‌های and2، full_adder و half_adder استفاده کنید.



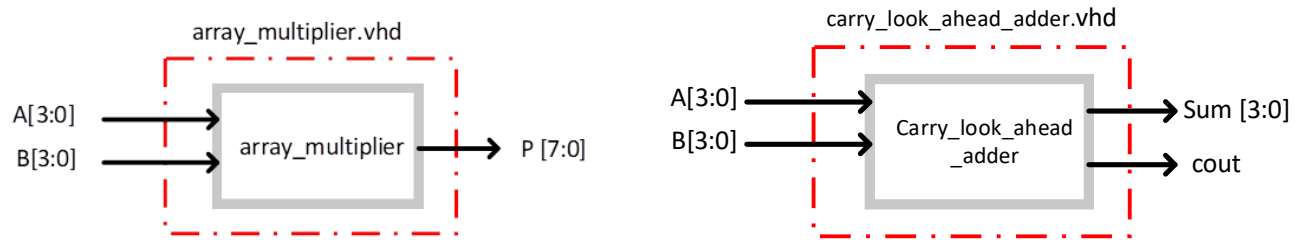
شکل ۴) ضرب‌کننده معمولی و پورت‌های ورودی و خروجی



شکل ۵) نمودار بلوکی ضرب‌کننده معمولی

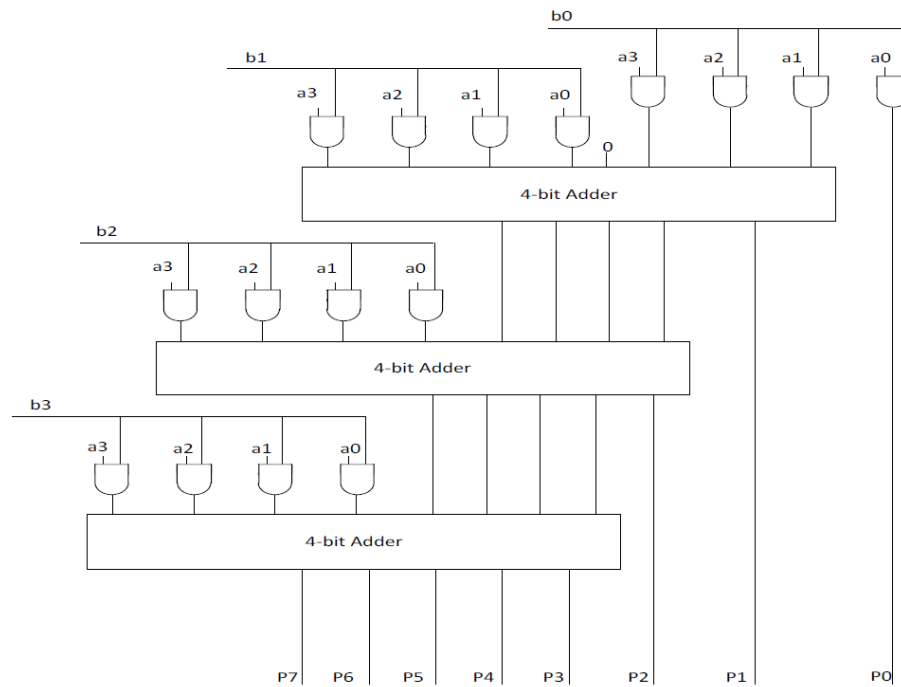
۲) ضرب کننده آرایه‌ای

در این ضرب کننده از ماژول and2 و واحد جمع کننده ۴ بیتی carry_look_ahead_adder و که در آزمایش چهارم طراحی شده است، استفاده کنید.



شکل ۷) ضرب کننده آرایه‌ای و پورت‌های ورودی و خروجی

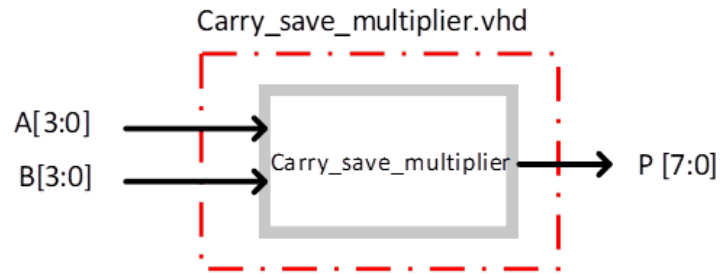
شکل ۶) carry lookahead adder و پورت‌های ورودی و خروجی



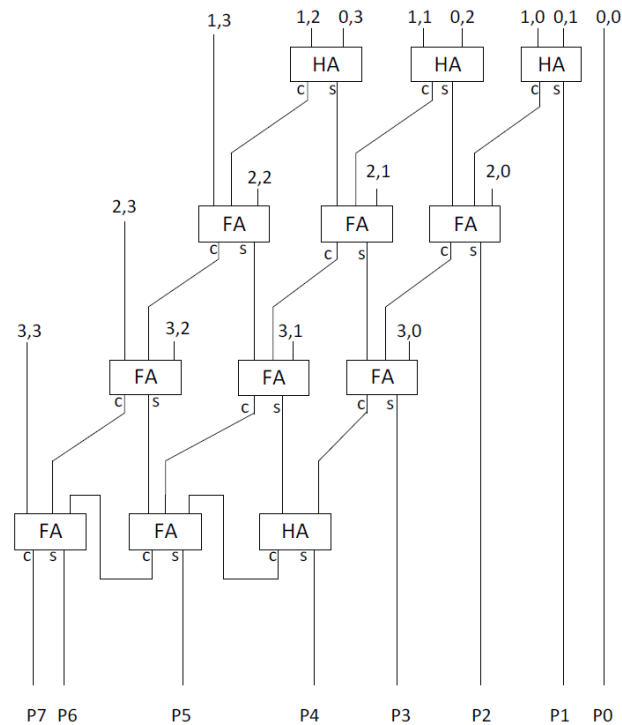
شکل ۸) نمودار بلوکی ضرب کننده آرایه‌ای

۳) ضرب کننده carry save adder

برای طراحی این ضرب کننده از ماژول های and2، full_adder و half_adder استفاده کنید.

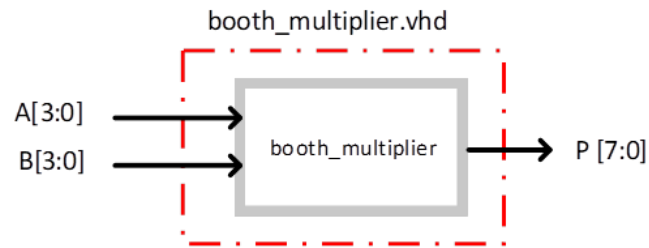


شکل ۹) ضرب کننده carry_save_adder و پورت های ورودی و خروجی



شکل ۱۰) نمودار بلوکی ضرب کننده carry save adder

۴) ضرب کننده booth (اختیاری)



شکل ۱۱) ضرب کننده booth و پورت های ورودی و خروجی

concatenate کردن سیگنالها در VHDL:

```
Signal S: std_logic_vector (3 downto 0) := "1010";  
Signal C : std_logic := '1';  
A_1 <= '0' & C & S(3 downto 1) & "01";    -- A <= "0110101"
```

نحوه نمایش شماتیک در Modelsim

```
vsim -vopt work.full_adder  
vsim -vopt -debugDB work.full_adder
```