

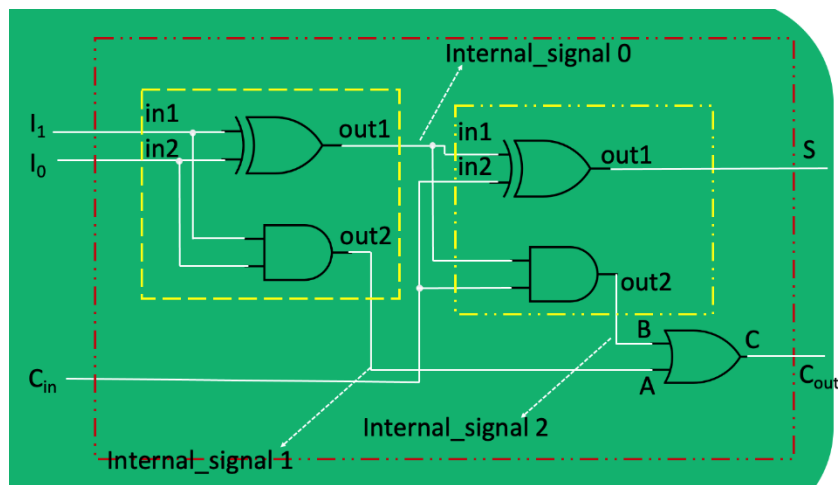
به نام خدا

گزارش کار آزمایش اول

علی نوروزیپیگی – فرهاد امان

هدف آزمایش: پیاده سازی full adder با زبان VHDL

تئوری آزمایش: مدار full adder به شکل زیر است:



این مدار از وصل کردن یک گیت or و دو half adder تشکیل شده است که مدار half adder خود نیز از یک گیت xor و یک گیت and ساخته می شود.

برای ساختن گیت های پایه and , xor , or مطابق با روش زیر هر موجودیت را تشکیل داده و خاصیت آن را تعریف می کنیم و سپس با نمونه سازی مدارهای بزرگتر را تشکیل می دهیم:

entity NAME is

port (signal\_names : mode);

end NAME;

architecture Structural of NAME is

--definition

end Structural;

در مدارهای بزرگ‌تر در قسمت architecture برای این‌که از موجودیت‌های دیگر نمونه‌سازی کنیم، قبل از تعریف، آن‌ها را به صورت component نشان داده و بعد از آن‌ها نمونه می‌گیریم و در صورت نیاز به سیگنال‌های میانی، آن‌ها را نیز تعریف می‌کنیم:

```
entity NAME is
```

```
.....
```

```
end NAME;
```

```
architecture Structural of NAME is
```

```
    component base_gate is
```

```
        port(signal_names : mod);
```

```
    end component NAME;
```

```
    --internal signals' declaration if needed
```

```
begin
```

```
    --definition
```

```
end Structural;
```

دستور کار آزمایش:

ابتدا اسم موجودیت را انتخاب کرده و سپس port های آن را مشخص می کنیم. برای port ها با استفاده از in یا out خروجی یا ورودی بودن را مشخص کرده و همچنین mode آن را std\_logic قرار می دهیم.

با توجه به شکل، ابتدای گیت های پایه and xor or را می سازیم. چون این گیت ها پایه هستند نیاز به نمونه سازی از مدارهای دیگر ندارند. پس بعد از تعریف موجودیت و مشخص کردن ورودی و خروجی برای هر کدام از آن ها، مستقیم رفتار آن ها را تعریف می کنیم.

برای این منظور سیگنال خروجی C را با روش زیر و با استفاده از کلیدواژه مناسب به رفتار مدنظر assign می کنیم:

and\_gate:

C <= A and B;

or\_gate:

C <= A or B;

xor\_gate:

C <= A xor B;

تمام کدهای بالا درون بلاک architecture قرار می گیرند که در تئوری آزمایش اشاره شد.

اما برای مدارهای half adder و full adder ، نیاز است که گیت های پایه مورد نیاز را درون آن ها مشخص کنیم. پس بعد از مشخص کردن نام و port ها، ابتدا گیت های xor و and را برای half adder و گیت or و half adder را برای full adder نمونه سازی می کنیم.

در half adder، حالا باید port map انجام دهیم و ورودی و خروجی گیت های نمونه سازی شده و مدار اصلی را به هم وصل کنیم. برای این منظور ورودی های گیت نمونه سازی شده را به شکل زیر به ورودی و خروجی مدار اصلی وصل می کنیم:

xor\_gate\_instance0 : xor\_gate port map (A => in1, B => in2, C => out1);

and\_gate\_instance0 : and\_gate port map (A => in1, B => in2, C => out1);

کار ساخت مدار half adder اینجا تمام می شود.

ساخت مدار full adder نیاز به یک گیت or و دو half adder دارد که طبق روش گفته در بالا component ها را نمونه سازی می‌کنیم. اما تفاوتی که در ساخت این مدار وجود دارد این است که برای این مدار نیاز به سیگنال های درونی داریم که با استفاده از کلیدواژه signal آن را درست می‌کنیم. سپس در port map این سیگنال ها را به ورودی ها و خروجی های مربوطه متصل می‌کنیم:

```
signal internal_signal0, internal_signal1, internal_signal2 : std_logic;
```

```
begin:
```

```
half_adder_instance0 : half_adder port map (in1=>I1, in2=>I0, out1=>internal_signal0,  
out2=>internal_signal1);
```

```
half_adder_instance1 : half_adder port map (in1=>internal_signal0, in2=>Cin, out1=>S,  
out2=>internal_signal2);
```

```
or_gate_instance0 : or_gate port map (A=>internal_signal1, B=>internal_signal2, C=>Cout);
```

فایل پروژه نیز به همراه این گزارش آپلود شده است.