

کد نویسی ترتیبی

- Process
- دستورات شرطی (IF)
- CASE

کد نویسی ترتیبی

- Process (Sensitivity list)

• با لیست حساسیت وارد حلقه می شود.

مثال

نوشتن کد Multiplexor

همانطور که می دانید

- ساخت MUX یعنی انتخاب از بین چند حالت
- در اینجا دو روش را می گوییم:
 1. استفاده از IF
 2. استفاده از CASE

کد MUX روش اول

```
LP0 : process (di, sel)
begin
    if (sel = "000") then
        do <= di(0);
    elsif (sel = "001") then
        do <= di(1);
    elsif (sel = "010") then
        .
        .
    elsif (sel = "110") then
        do <= di(6);
    else
        do <= di(7);
    end if;
end process;
```

کد MUX روش دوم

```
LP1 : process (di, sel)
begin
  case sel is
    when "000" => do <= di(0);
    when "001" => do <= di(1);
    .
    .
    when "110" => do <= di(6);
    when others => do <= di(7);
  end case;
end process;
end RTL2;
```

فرق دو روش در چیست؟

- دستور IF شرط ها را به ترتیب نگاه می کند و با برآورده نشدن هر شرط به سراغ شرط بعدی می رود.
- ولی دستور CASE شروط را همزمان نگاه می کند

مثال بعدی

Flip-flop و Latch

Latch

```
process (clock, reset,input)
begin
    if reset='1' then
        output <= '0';
    elsif clock='1' then
        output <=input;
    end if;
end process;
```

Flip-Flop

```
process (clock, reset)
begin
    if reset='1' then
        output <= '0';
    elsif (clock'event and clock='1') then
        output <=input;
    end if;
end process;
```

Counter

• کد زیر غلط است:

```
process (clock, reset)
begin
  if reset='1' then
    count <= (others => '0');
  elsif clock='1' and clock'event then
    if clock_enable='1' then
      count <= count + 1;
    end if;
  end if;
end process;
```

پورت خروجی را نمیتوان به روز کرد

- `count <= count + 1;`

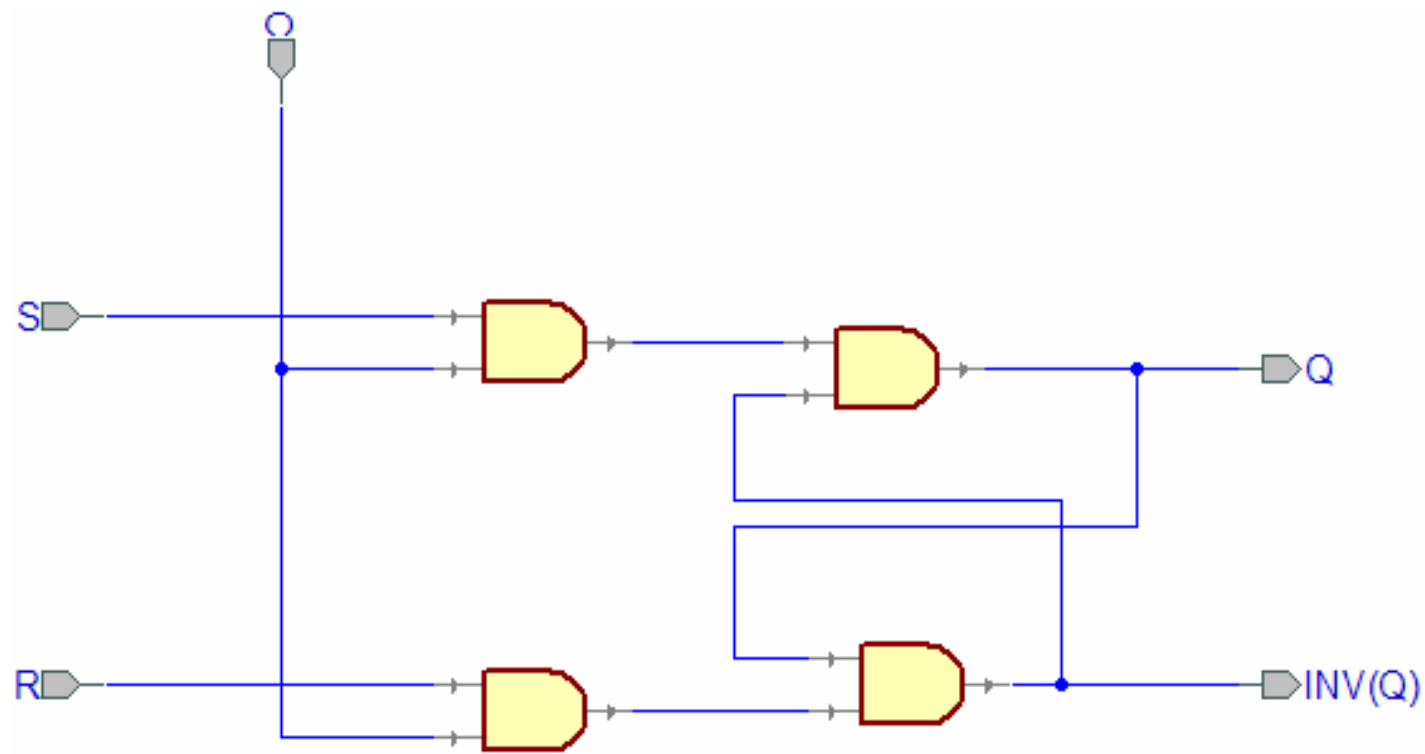
راه حل : استفاده از سیگنال

- عملیات مورد نظر ما روی سیگنال انجام شده و نهایتاً به خروجی منتقل می شود

```
process(CLK)
begin
    if rising_edge(CLK) then
        if CLR = '1' then
            count_sig <= (others => '0');
        else
            count_sig <= count_sig + 1;
        end if;
    end if;
end process;

Q <= count_sig;
```

نکته: به مدار زیر نگاه کنید



- در این مدار پورت Q که خروجی است و به ورودی وصل شده

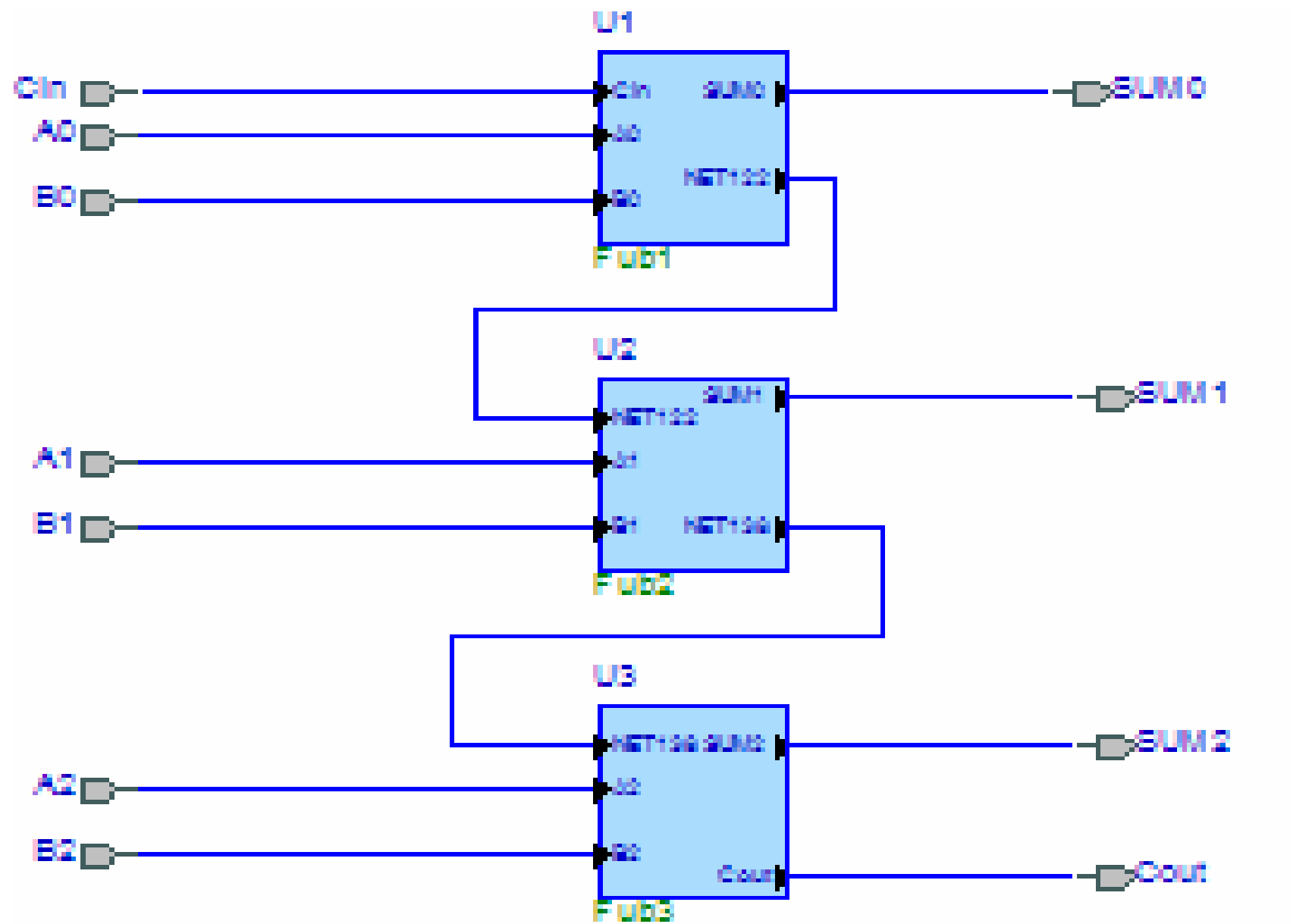
- در VHDL مجاز نیستیم که پورت خروجی را به ورودی مصل کنیم

- راه حل: استفاده از سیگنال به عنوان واسط

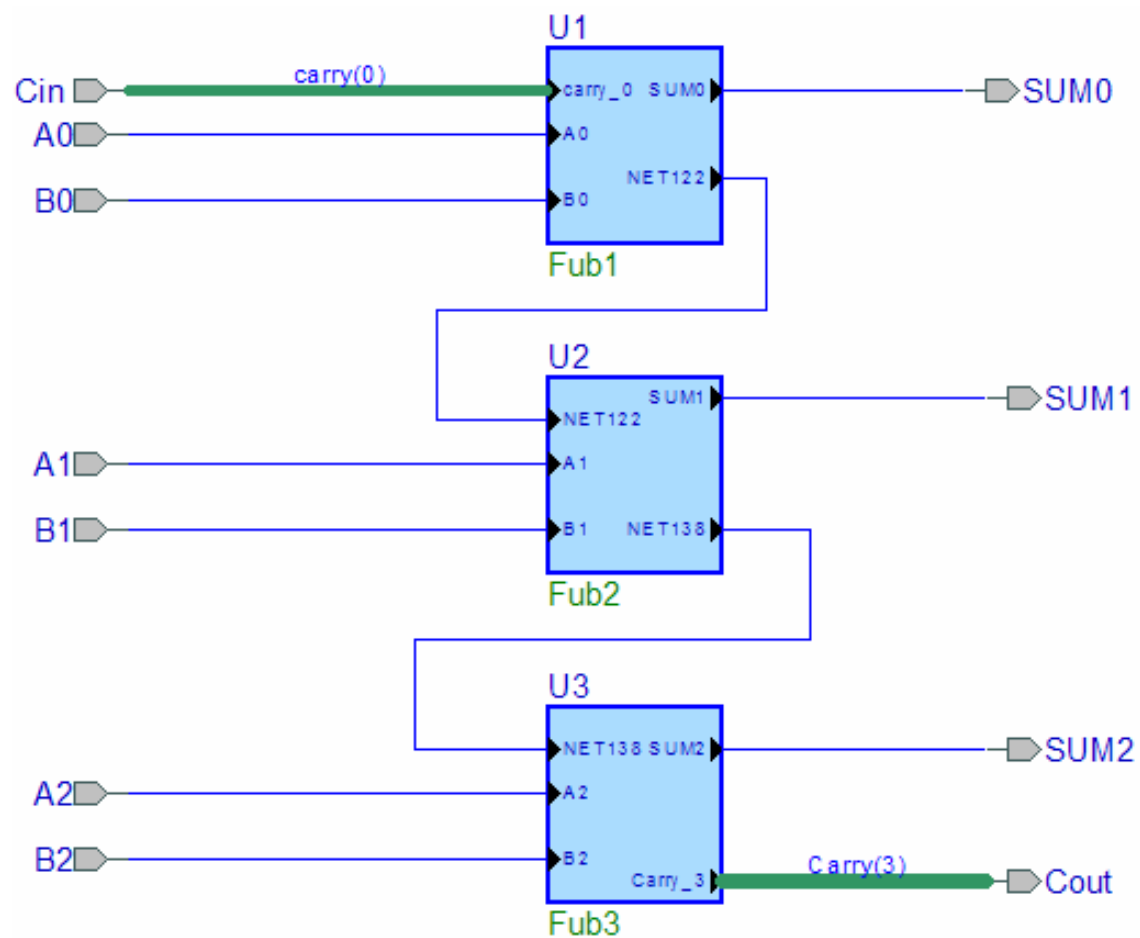
تعریف حلقه در VHDL

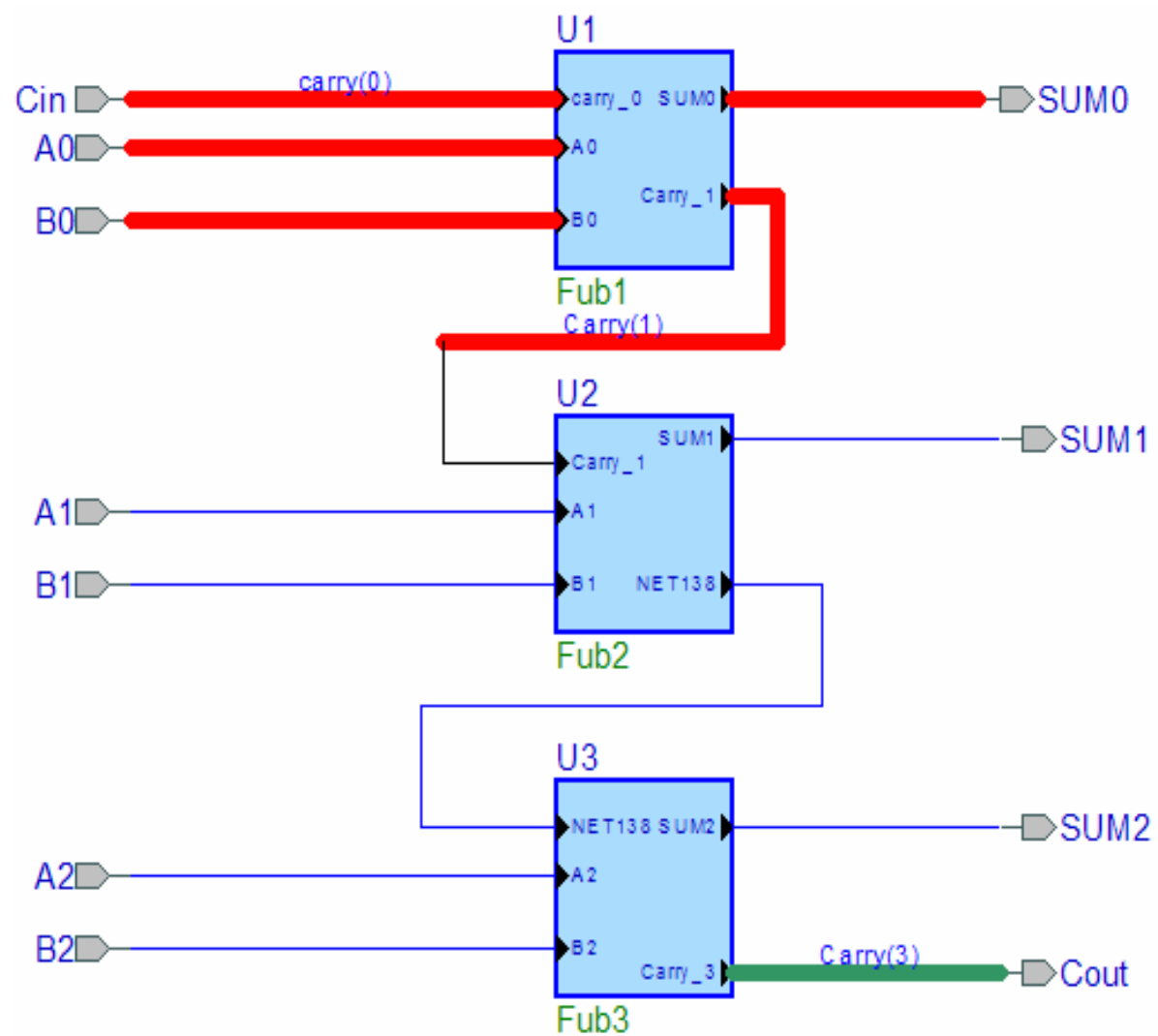
ساخت جمع کننده چند بیتی از جمع کننده تک بیتی

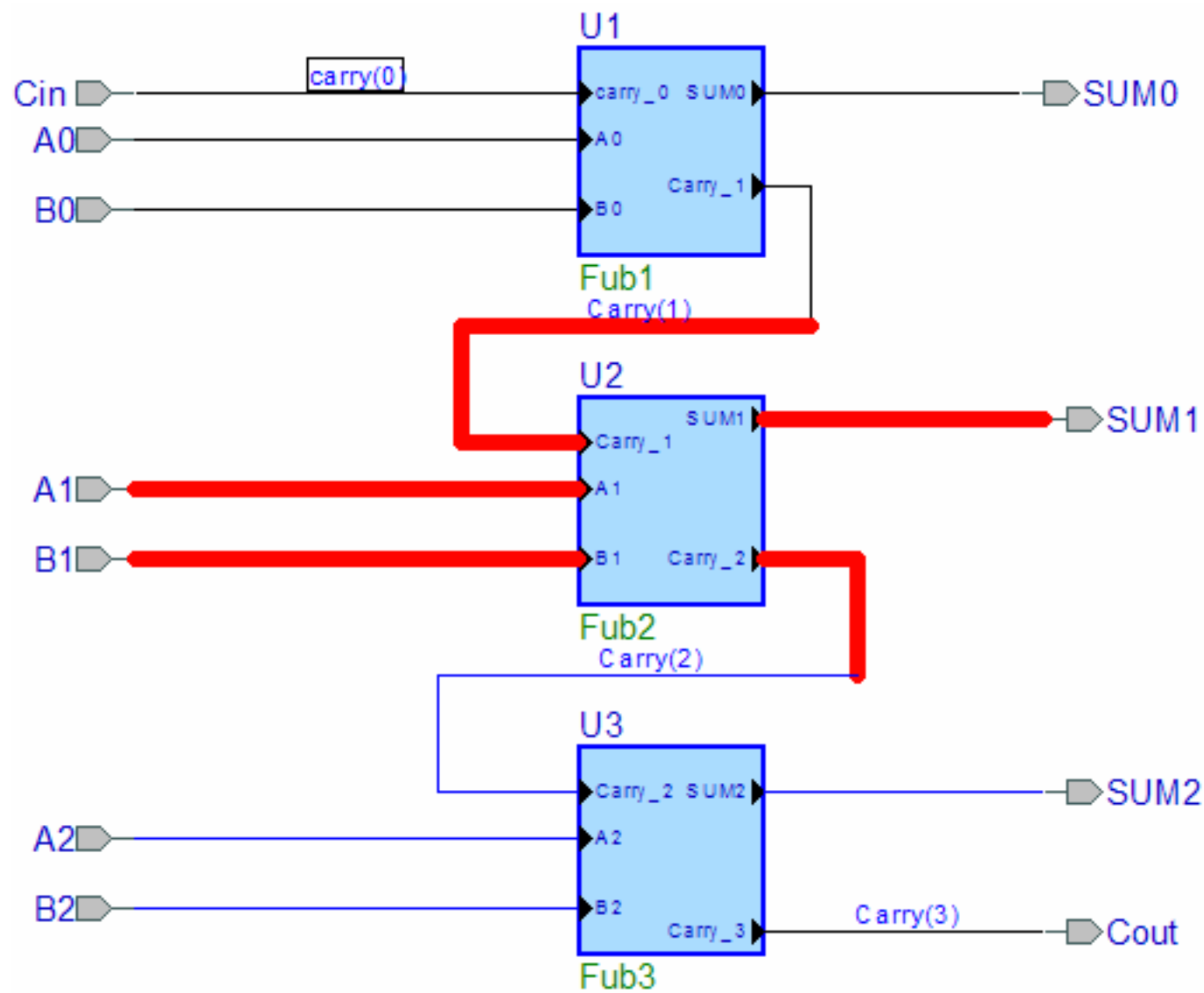
• مدار زیر را در نظر بگیرید

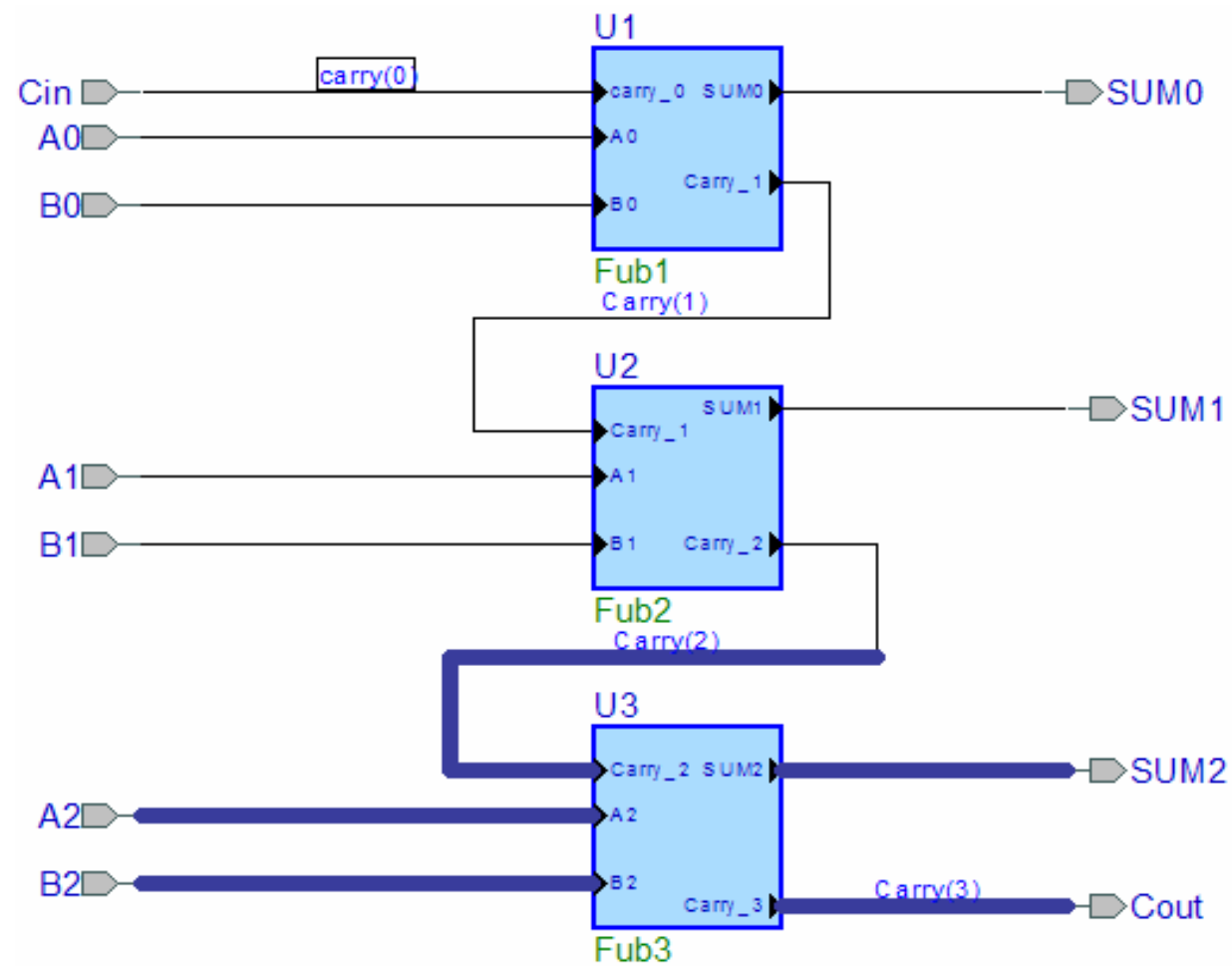


فرض کنید می خواهیم این مدار را با مراحل زیر Port Map کنیم.









اعمال بالا مرحله ای است.

• دستور generate