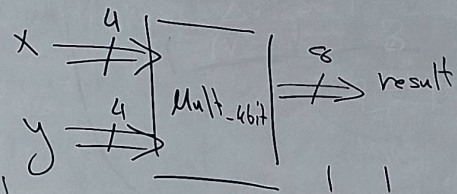


# Multiplier

$x \rightarrow y(10)_{10}$   
 $y \rightarrow 0000$   
 $r_1$



```

      1 1
      | |
      | | 1 1 0 0 r3
      | | 1 1 0 0 r4
  
```

---

1 0 1 1 0 1 0 0 → r6

```

      1 1
      | |
      | | 1 1 1 r1
      | | 1 1 0 r2
  
```

---

1 0 1 1 0 1 → r5

```

      1
      | |
      | | 0 1 1 0 1 r5
      | 0 1 1 0 1 0 0 r6
  
```

---

1 1 1 0 0 0 0 1 → result

```

      3 ← 0
      | | | |
      * | | | |
      | | | |
  
```

---

```

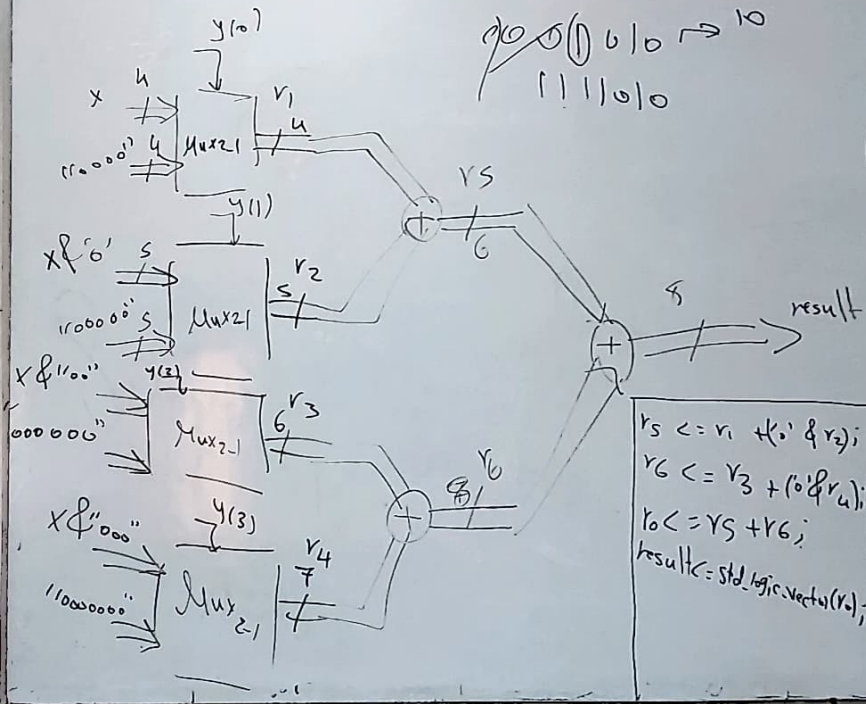
      3 3 1 1 1 → r1
      2 3 1 1 1 0 → r2
      | 1 1 1 0 0 → r3
      | 1 1 1 0 0 0 → r4
  
```

---

1 1 1 0 0 0 0 1

1010 → 10 ms  
→ 6 sign

~~10101010~~ → 10  
1111010



entity Mult\_4bit is

Port( $x, y$  : in std\_logic\_vector(3 downto 0);  
result : out std\_logic\_vector(7 downto 0));  
end Mult\_4bit;

architecture rtl of Mult\_4bit is  
Signal  $r_1$  : unsigned(3 downto 0);  
Signal  $r_2$  : unsigned(4 downto 0);  
Signal  $r_3, r_5$  : unsigned(5 downto 0);  
Signal  $r_4$  : unsigned(6 downto 0);  
begin  
 $r_6, r_0$  : unsigned(7 downto 0);  
 $r_1 <= \text{unsigned}(x)$  when  $y(0) = '1'$  else (others = '0');  
 $r_2 <= (\text{unsigned}(x) \& '0')$  when  $y(1) = '1'$  else (others = '0');  
 $r_3 <= (\text{unsigned}(x) \& '00')$  when  $y(2) = '1'$  else (others = '0');  
 $r_4 <= (\text{unsigned}(x) \& '000')$  when  $y(3) = '1'$  else (others = '0');

$r_5 <= r_1 + r_2;$   
 $r_6 <= r_3 + r_4;$   
 $r_0 <= r_5 + r_6;$   
result <= std\_logic\_vector(r\_0);