



پیش گزارش آزمایش سوم

اعضای گروه:

صادق محمدیان: ۴۰۱۱۰۹۴۷۷

متین محمدی: ۴۰۱۱۱۰۳۲۹

امیرحسین ملک محمدی: ۴۰۱۱۰۶۵۷۷

قسمت اول:

برای این قسمت ابتدا ماژول `cascadable_1_bit_comparator` را طراحی می کنیم که شمای کلی آن به شکل زیر است:

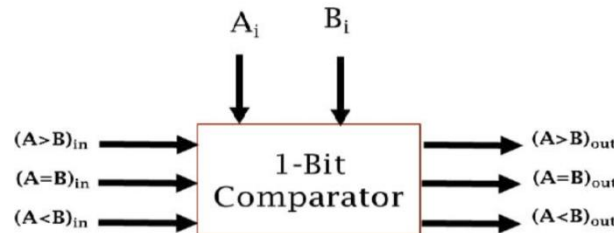


Fig 3: 1-Bit Cascaded Comparator

سپس با کنار هم قرار دادن ۴ `cascadable_1_bit_comparator` یک 4bit comparator می سازیم که شمای کلی آن به شکل زیر است:

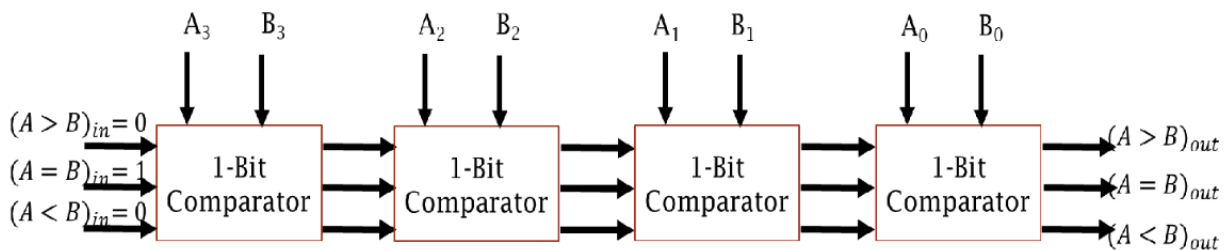


Fig 2: 4-Bit Comparator using 1-bit Cascaded Comparator

ماژول `cascadable_1_bit_comparator`:

```
1 module CASCADABLE_1_BIT_COMPARATOR(input la, input eq, input le, input a, input b, output laa, output eqq, output lee);
2
3     assign eqq = eq & (a==b);
4     assign laa = (eq & (a>b)) | la;
5     assign lee = (eq & (a<b)) | le;
6
7 endmodule
```

ماژول `four_bit_comparator`:

```
1 module FOUR_BIT_COMPARATOR(input [3:0]a, input [3:0]b, output la, output eq, output le);
2     CASCADABLE_1_BIT_COMPARATOR comp1(0,1,0,a[3],b[3],laa1,eqq1,lee1);
3     CASCADABLE_1_BIT_COMPARATOR comp2(laa1,eqq1,lee1,a[2],b[2],laa2,eqq2,lee2);
4     CASCADABLE_1_BIT_COMPARATOR comp3(laa2,eqq2,lee2,a[1],b[1],laa3,eqq3,lee3);
5     CASCADABLE_1_BIT_COMPARATOR comp4(laa3,eqq3,lee3,a[0],b[0],la,eq,le);
6 endmodule
```

روش کار به این صورت است که با مقایسه پر ارزش ترین بیت شروع می کنیم و بیت به بیت مقایسه می کنیم و با استفاده از نتایج مقایسه بیت قبلی مقدار های la, eq, le را آپدیت می کنیم. باید توجه کنیم مقادیر ورودی laa, eqq, lee ورودی به اولین cascadable_1_bit_comparator باید به ترتیب 0,1,0 باشند.

ماژول تست آن به صورت زیر می باشد:

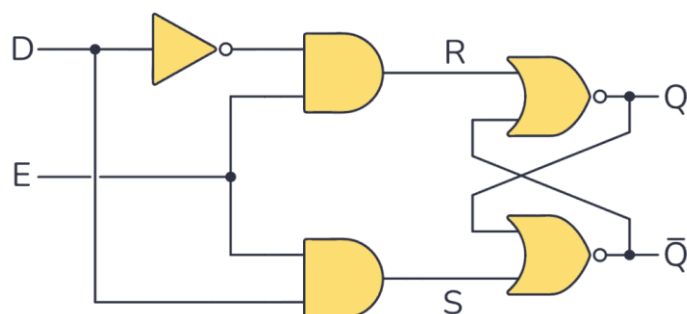
```
1 module tb();
2     reg [3:0]a;
3     reg [3:0]b;
4     wire la,eq,le;
5
6     FOUR_BIT_COMPARATOR comp(a,b,la,eq,le);
7
8     initial
9     begin
10        a=4'b1111 ;
11        b=4'b1011 ;
12        #10
13        $display("a = ", a , " b = ", b , "=>" , " la = " , la , " eq = " , eq , " le = " , le);
14        a=4'b1011 ;
15        b=4'b1011 ;
16        #10
17        $display("a = ", a , " b = ", b , "=>" , " la = " , la , " eq = " , eq , " le = " , le);
18        a=4'b1001 ;
19        b=4'b1011 ;
20        #10
21        $display("a = ", a , " b = ", b , "=>" , " la = " , la , " eq = " , eq , " le = " , le);
22        a=4'b1011 ;
23        b=4'b0000 ;
24        #10
25        $display("a = ", a , " b = ", b , "=>" , " la = " , la , " eq = " , eq , " le = " , le);
26
27    end
28
29 endmodule
```

و نتیجه حاصل از شبیه سازی نیز به شکل زیر است:

```
# a = 15 b = 11=> la = 1 eq = 0 le = 0
# a = 11 b = 11=> la = 0 eq = 1 le = 0
# a = 9 b = 11=> la = 0 eq = 0 le = 1
# a = 11 b = 0=> la = 1 eq = 0 le = 0
```

قسمت دوم:

طراحی ما باید شامل یک ماژول باشد و ما باید نتیجه مقایسه تا بیت قبلی را نگه داریم بدین منظور می توانیم از Latch استفاده کنیم.



| E or Clk | D | Q | Q' |
|----------|---|-------|----|
| 0 | 0 | Latch | |
| 0 | 1 | Latch | |
| 1 | 0 | 0 | 1 |
| 1 | 1 | 1 | 0 |

با توجه به نتیجه مقایسه تا بیت قبلی و بیت های ورودی جدید ۳ حالت رخ می دهد:

- (۱) **عدد اول بزرگتر از عدد دوم باشد:** در اینصورت با ورود بیت های جدید همچنان عدد اول بزرگتر می ماند.
- (۲) **عدد اول کوچکتر از عدد دوم باشد:** در اینصورت با ورود بیت های جدید همچنان عدد اول کوچکتر می ماند.
- (۳) **دو عدد مساوی باشند:** در این صورت

اگر بیت های ورودی عدد اول بزرگتر باشد عدد اول بزرگتر خواهد بود

اگر دو بیت برابر باشند عدد نهایی بزرگتر خواهد بود

و در صورتی که بیت عدد دوم بزرگتر باشد عدد اول کوچکتر خواهد بود.

طراحی ما برای این ماژول به صورت زیر خواهد بود:

```
1  module SERIAL_COMPARATOR(input clk, input rst, input a, input b,
2      output la, output eq,output le);
3
4
5      wire la_d, eq_d, le_d;
6
7      assign la_d = (~rst) & ((eq & (a > b)) | la);
8      assign eq_d =  rst | ((eq & (a == b)) & (~rst));
9      assign le_d = (~rst) & ((eq & (a < b)) | le);
10
11
12      //latches
13      wire la_bar, eq_bar ,le_bar;
14
15      assign la = ~(la_bar & ~(clk & la_d));
16      assign la_bar = ~(la & ~(clk & ~(la_d)));
17
18      assign eq = ~(eq_bar & ~(clk & eq_d));
19      assign eq_bar = ~(eq & ~(clk & ~(eq_d)));
20
21      assign le = ~(le_bar & ~(clk & le_d));
22      assign le_bar = ~(le & ~(clk & ~(le_d)));
23
24
25
26  endmodule
```

ماژول آزمون ما نیز به صورت زیر خواهد بود:

```
module SERIAL_COMPARATOR_TB();
    reg rst, clk, a, b;
    wire la, eq, le;

    SERIAL_COMPARATOR cfomparator(clk, rst, a, b, la, eq, le);

    always begin
        #5 clk = ~clk;

    end
    initial begin
        clk = 0;
        rst = 1;
        #10
        $display("time: %d, reset: %b a_in: %b, b_in: %b, la: %b, eq: %b, le: %b", $time, rst, a, b, la, eq, le);
        rst = 0;
        a = 1;
        b = 0;
        #10
        $display("time: %d, reset: %b a_in: %b, b_in: %b, la: %b, eq: %b, le: %b", $time, rst, a, b, la, eq, le);
        a = 0;
        b = 1;
        #10
        $display("time: %d, reset: %b a_in: %b, b_in: %b, la: %b, eq: %b, le: %b", $time, rst, a, b, la, eq, le);
        a = 0;
        b = 1;
        #10
        $display("time: %d, reset: %b a_in: %b, b_in: %b, la: %b, eq: %b, le: %b", $time, rst, a, b, la, eq, le);
        rst = 1;
        #10
        $display("time: %d, reset: %b a_in: %b, b_in: %b, la: %b, eq: %b, le: %b", $time, rst, a, b, la, eq, le);
        rst = 0;
        a = 1;
        b = 1;
        #10
```

```

        $display("time: %d, reset: %b a_in: %b, b_in: %b, la: %b, eq: %b, le:
%b", $time, rst, a, b, la, eq, le);
        a = 0;
        b = 1;
        #10
        $display("time: %d, reset: %b a_in: %b, b_in: %b, la: %b, eq: %b, le:
%b", $time, rst, a, b, la, eq, le);
        a = 1;
        b = 0;
        #10
        $display("time: %d, reset: %b a_in: %b, b_in: %b, la: %b, eq: %b, le:
%b", $time, rst, a, b, la, eq, le);

    end

endmodule

```

و نتایج آن به صورت زیر است که نشان می دهد کد ما به درستی کار می کند:

```

time:          10, reset: 1 a_in: x, b_in: x, la: 0, eq: 1, le: 0
time:          20, reset: 0 a_in: 1, b_in: 0, la: 1, eq: 0, le: 0
time:          30, reset: 0 a_in: 0, b_in: 1, la: 1, eq: 0, le: 0
time:          40, reset: 0 a_in: 0, b_in: 1, la: 1, eq: 0, le: 0
time:          50, reset: 1 a_in: 0, b_in: 1, la: 0, eq: 1, le: 0
time:          60, reset: 0 a_in: 1, b_in: 1, la: 0, eq: 1, le: 0
time:          70, reset: 0 a_in: 0, b_in: 1, la: 0, eq: 0, le: 1
time:          80, reset: 0 a_in: 1, b_in: 0, la: 0, eq: 0, le: 1

```

ابتدا دو سریال 100 و 011 را مقایسه می کنیم و سپس بعد از ریست کردن مدار دوسریال 101 و 110 را تست می کنیم.

