

آز سیستم های دیجیتال

دکتر اجلائی

مبینا حیدری، عاطفه قندهاری، نیکا قادری
بهار ۱۴۰۳



آزمایش سوم

توصیف جریان داده

تاریخ گزارش: ۲۶ فروردین ۱۴۰۳

۱ شرح آزمایش

این آزمایش به ساخت مقایسه گر می پردازد در بخش اول باید با استفاده از مدار ترکیبی یک مقایسه گر چهار بیتی ساخت در بخش دوم با کمک فقط یک module باید مقایسه گری ساخت که تا کلاک فعال حاصل مقایسه را اعلام کند و همینطور امکان reset نیز دارد

ورودی ها شامل a, b که در بخش اول ۴ بیتی و در بخش دوم یک بیتی است و reset clock، که فقط در بخش دوم مورد استفاده قرار می گیرد و خروجی ها شامل e(equal), g(greater), l(less)، است و هر یک که برقرار باشد ۱ میشود و در غیر این صورت صفر میشود.

۲ بخش اول: مقایسه گر ۴ بیتی

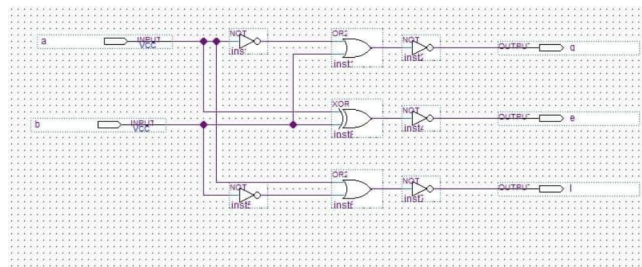
- ابتدا یک ماژول برای مقایسه هر بیت به نام comparator میسازیم. خروجی ها را بر اساس جدول زیر به دست می آوریم. در نتیجه کد

a	b	g	l	e
0	0	0	0	1
0	1	0	1	0
1	0	1	0	0
1	1	0	0	1

شکل ۱: جدول مقایسه

verilog این ماژول به صورت زیر است:

```
module comparator (
    input a,
    input b,
    output g,
    output e,
    output l
);
    assign g = ~(~a | b);
    assign e = ~(a ^ b);
    assign l = ~((~b) | a);
endmodule
```



شکل ۲: شماتیک comparator

- حال نیاز به یک ماژول داریم که ورودی e, g, l دارد که نشان میدهد عدد تا بیت قبل بزرگتر، کوچکتر و یا مساوی بوده و همچنین ورودی a, b را نیز دارد که بیت فعلی می باشد و در صورتی که lin یا gin یک باشد به همان صورت lout و یا gout یک میشود ولی اگر مساوی باشد دو بیت a, b در خروجی تاثیر میگذارد نام این ماژول را cascablecomparator می گذاریم و کد Verilog آن به صورت زیر است:
 - در این بخش باید دو عدد ۴ بیتی با هم مقایسه شود و حاصل مقایسه در خروجی نشان داده شود. برای این کار یک ماژول داریم که متشکل از ۴ ماژول cascablecomparator که خروجی هر ماژول به عنوان ورودی به ماژول بعدی داده میشود.
- کد verilog این ماژول به این صورت است:

```

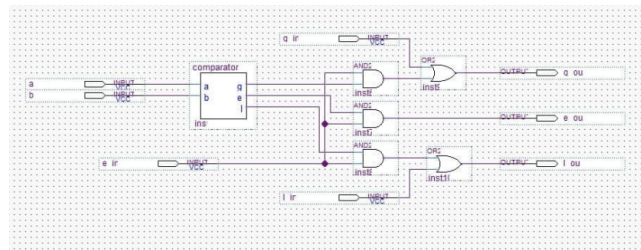
module cascable_comparator (
    input a,
    input b,
    input g_in,
    input e_in,
    input l_in,
    output g_out,
    output e_out,
    output l_out
);

comparator COM(
    .a(a),
    .b(b),
    .g(g_in),
    .e(e_in),
    .l(l_in)
);

assign g_out = g_in | (e_in & g);
assign e_out = e_in & e;
assign l_out = l_in | (e_in & l);

endmodule

```



شکل ۳: شماتیک cascable comparator

```

module four_bit_comparator (
    input [3:0] a,
    input [3:0] b,
    input g_in,
    input e_in,
    input l_in,
    output g_out,
    output e_out,
    output l_out
);

cascable_comparator COM3(
    .a(a[3]),
    .b(b[3]),
    .g_in(g_in),
    .e_in(e_in),
    .l_in(l_in),
    .g_out(g3),
    .e_out(e3),
    .l_out(l3)
);

cascable_comparator COM2(
    .a(a[2]),
    .b(b[2]),
    .g_in(g3),
    .e_in(e3),
    .l_in(l3),
    .g_out(g2),
    .e_out(e2),
    .l_out(l2)
);

cascable_comparator COM1(
    .a(a[1]),
    .b(b[1]),
    .g_in(g2),
    .e_in(e2),
    .l_in(l2),
    .g_out(g1),
    .e_out(e1),
    .l_out(l1)
);

endmodule

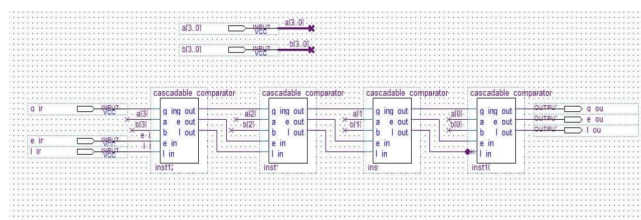
```

```

cascable_comparator COM0(
    .a(a[0]),
    .b(b[0]),
    .g_in(g1),
    .e_in(e1),
    .l_in(l1),
    .g_out(g_out),
    .e_out(e_out),
    .l_out(l_out)
);

endmodule

```



شکل ۴: four bit comparator

tb comparator bit for مازولی است که برای مقدار دهی به a, b استفاده میشود و تمام مقدار ممکن ۴ بیتی را برای این دو ورودی در نظر میگیرد:

```

module four_bit_comparator_TB ();
reg [4:0] a, b;
wire g, e, l;

four_bit_comparator COM(
    .a(a[3:0]),
    .b(b[3:0]),
    .g_in(1'b0),
    .e_in(1'b1),
    .l_in(1'b0),
    .g_out(g),
    .e_out(e),
    .l_out(l)
);

initial begin
    $monitor("%d, %d, g = %d, e = %d, l = %d", a, b, g, e, l);
    for (a = 0; a < 16; a = a + 1) begin
        for (b = 0; b < 16; b = b + 1) begin
            #10;
        end
    end
    $stop;
end
endmodule

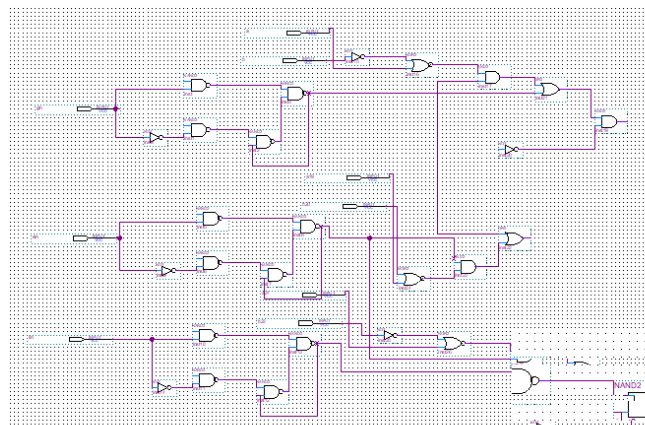
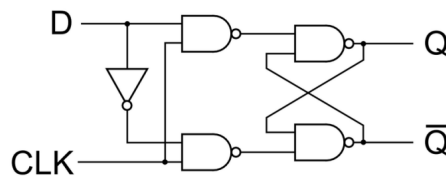
```

مدار نهایی در کوارتوس سنتز شد و summary آن به شکل زیر است:

Flow Summary	
Flow Status	Successful - Tue Apr 06 11:26:08 2021
Quartus II 64-Bit Version	13.0.1 Build 232 06/12/2013 SP 1 SJ Web Edition
Revision Name	Tootooleh
Top-level Entity Name	four_bit_comparator
Family	Cyclone IV GX
Total logic elements	12 / 14,400 (< 1 %)
Total combinational functions	12 / 14,400 (< 1 %)
Dedicated logic registers	0 / 14,400 (0 %)
Total registers	0
Total pins	14 / 81 (17 %)
Total virtual pins	0
Total memory bits	0 / 552,960 (0 %)
Embedded Multiplier 9-bit elements	0
Total GXB Receiver Channel PCS	0 / 2 (0 %)
Total GXB Receiver Channel PMA	0 / 2 (0 %)
Total GXB Transmitter Channel PCS	0 / 2 (0 %)
Total GXB Transmitter Channel PMA	0 / 2 (0 %)
Total PLLs	0 / 3 (0 %)
Device	EP4CGX15BF14C6
Timing Models	Final

۳ بخش دوم: مقایسه گر سریالی

- در این بخش فقط به ماژول نیاز داریم و چون میخواهیم نتایج را ذخیره کنیم مدار ترتیبی است و با کمک flop flip d میسازیم:



شکل ۵: شماتیک

- در این ماژول باید از ۳dff استفاده می کنیم ورودی dff ها به ترتیب lin gin ein هستند و خروجی ان l,g,e هستند هر یک از سیگنال های ورودی توسط یک مدار ترکیبی به دست می اید مانند قسمت قبل ، برای مقایسه اعداد ۴ بیتی در این ماژول از بیت پر ارزش شروع میشود. چون

سیگنال reset داریم با استفاده از or به حالت اولیه برمیگردیم کد وریلاگ به این صورت می باشد:

```
module serial_comparator (
    input a,
    input b,
    input reset,
    input clock,
    output g,
    output e,
    output l
);

wire g_d, g_i1, g_i2, g_not;
wire e_d, e_i1, e_i2, e_not;
wire l_d, l_i1, l_i2, l_not;

/* g reg (g_d is input) */
assign g_i1 = ~(clock & g_d);
assign g_i2 = ~(clock & ~g_d);
assign g = ~(g_i1 & g_not);
assign g_not = ~(g_i2 & g);
/* g reg */

/* e reg (e_d is input) */
assign e_i1 = ~(clock & e_d);
assign e_i2 = ~(clock & ~e_d);
assign e = ~(e_i1 & e_not);
assign e_not = ~(e_i2 & e);
/* e reg */

/* l reg (l_d is input) */
assign l_i1 = ~(clock & l_d);
assign l_i2 = ~(clock & ~l_d);
assign l = ~(l_i1 & l_not);
assign l_not = ~(l_i2 & l);
/* l reg */

assign g_d = ~reset & (g | (e & ~((~a) | b)));
assign e_d = reset | (e & ~(a ^ b));
assign l_d = ~reset & (l | (e & ~((~b) | a)));

endmodule
```

- ماژول serial_comparator_tb تمام ورودی های ممکن را چک میکند کد وریلاگ آن به صورت زیر است:

```
module serial_comparator_TB ();

reg [4:0] a, b;
reg reset = 1, clock = 0;
integer i;
wire g, e, l;

always #10 clock = ~clock;

serial_comparator COM(
    .a(a[i]),
    .b(b[i]),
    .reset(reset),
    .clock(clock),
    .g(g),
    .e(e),
    .l(l)
);

initial begin
    for (a = 0; a < 16; a = a + 1) begin
        for (b = 0; b < 16; b = b + 1) begin
            #20 reset = 0;
            for (i = 3; i >= 0; i = i - 1) begin
                #20;
            end
            $display("%d, %d, g = %d, e = %d, l = %d", a, b, g, e, l);
            reset = 1;
        end
    end
    $stop;
end

endmodule
```

- با کوآرتوس آن را سنتز میکنیم:

Flow Summary	
Flow Status	Successful - Tue Apr 06 11:30:26 2021
Quartus II 64-Bit Version	13.0.1 Build 232 06/12/2013 SP 1 SJ Web Edition
Revision Name	Tootooleh
Top-level Entity Name	serial_comparator
Family	Cyclone IV GX
Total logic elements	7 / 14,400 (< 1 %)
Total combinational functions	7 / 14,400 (< 1 %)
Dedicated logic registers	0 / 14,400 (0 %)
Total registers	0
Total pins	7 / 81 (9 %)
Total virtual pins	0
Total memory bits	0 / 552,960 (0 %)
Embedded Multiplier 9-bit elements	0
Total GXB Receiver Channel PCS	0 / 2 (0 %)
Total GXB Receiver Channel PMA	0 / 2 (0 %)
Total GXB Transmitter Channel PCS	0 / 2 (0 %)
Total GXB Transmitter Channel PMA	0 / 2 (0 %)
Total PLLs	0 / 3 (0 %)
Device	EP4CGX15BF14C6
Timing Models	Final