📝 گزارش آزمایش ۳

آزمایشگاه طراحی سیستم‌های دیجیتال

> *Alireza Habibzadeh* `99109393`
> *April 12, 2022*

پیاده‌سازی مقایسه‌کننده‌ی یک‌بیتی

برای پیاده‌سازی مقایسه‌کننده‌ی یک بیتی باید از روش `assign` استفاده کنیم. اتفاقا این روش راحت‌تر نیز هست. کافی است خروجی‌های لازم را به صورت تابعی (functional) بر حسب ورودی‌ها بنویسیم.

در حالتی که مقایسه‌کننده ورودی‌های آبشاری نداشته باشد جدول درستی آن مطابق جدول زیر است:

G	E	L	B	A
0	1	0	0	0
0	0	1	1	0
1	0	0	0	1
0	1	0	1	1

اما در حالت آبشاری کافی است حالت‌های تساوی را تغییر دهیم. در این حالات باید به نتیجه‌ی مقایسه‌ی کم‌ارزش‌تر رجوع کنیم. در نهایت عبارات می‌شوند:

```
assign E = (A == B) & E0;
assign L = A == B ? L0 : A < B;
assign G = A == B ? G0 : A > B;
```

در عبارات بالا ورودی‌های `L0`، `E0`، و `G0` نتیجه‌ی مقایسه‌ی کم‌ارزش‌تر هستند. عبارت اول می‌گوید برای تساوی باید هم نتیجه‌ی مقایسه‌ی کم‌ارزش تساوی باشد هم بیت‌های ورودی. برای بیت‌های بزرگ‌تری و کوچک‌تری ابتدا در صورت حالت تساوی به مقایسه‌ی کم‌ارزش ارجاع داده شده و در صورت عدم تساوی مستقیما شرط مقایسه خروجی داده می‌شود.

در نهایت کد مقایسه‌کننده‌ی یک‌بیتی می‌شود:

```
module comparator(
    input A,
    input B,
    input E0,
    input L0,
    input G0,
    output E,
    output L,
    output G
    );

      assign E = (A == B) & E0;
      assign L = A == B ? L0 : A < B;
      assign G = A == B ? G0 : A > B;
endmodule
```

پیاده‌سازی مقایسه‌کننده‌ی ۵بیتی

ورودی‌های اولین رقم را حالت تساوی می‌دهیم چرا که در صورت تساوی ارقام با ارزش‌تر، این رقم باید تصمیم‌گیرنده باشد پس باید فکر کند ارقام کم‌ارزش‌تر (که وجود ندارند) مساوی هستند. حال کافی است ۵ مقایسه‌کننده را به صورت آبشاری به هم وصل کنیم. برای این کار به چهار سری `wire` برای هر یک از ورودی/خروجی‌های `L`، `E`، و `G` نیاز داریم. برای هر یک از آن‌ها یک آرایه `wire` به طول ۴ تعریف کردم که در مقایسه‌کننده‌ها به ترتیب استفاده شده‌اند.

همچنین از آن‌جایی که ورودی‌های ما به صورت `[4:0]` هستند یعنی little endian می‌باشند و برای مثال `A[0]` کم‌ارزش‌ترین بیت است که باید به مقایسه‌کننده‌ی ابتدای زنجیر داده شود و...

```
module five_bit_comparator(
    input [4:0] A,
    input [4:0] B,
    output E,
    output L,
    output G
    );
```

```
    wire [3:0] E0;
    wire [3:0] L0;
    wire [3:0] G0;

    comparator c0(A[0], B[0], 1'b1, 1'b0, 1'b0, E0[0], L0[0], G0[0]);
    comparator c1(A[1], B[1], E0[0], L0[0], G0[0], E0[1], L0[1], G0[1]);
    comparator c2(A[2], B[2], E0[1], L0[1], G0[1], E0[2], L0[2], G0[2]);
    comparator c3(A[3], B[3], E0[2], L0[2], G0[2], E0[3], L0[3], G0[3]);
    comparator c4(A[4], B[4], E0[3], L0[3], G0[3], E, L, G);
endmodule
```

تست

برای بررسی صحت عملکرد مدار از test bench زیر استفاده شده است. البته بیشتر قسمت‌های آن به صورت خودکار توسط نرم‌افزار تولید می‌شود و کافی است داخل بلاک `initial` را تغییر دهیم.

```
module five_bit_comparator_tb;
    // Inputs
    reg [4:0] A;
    reg [4:0] B;
    // Outputs
    wire E;
    wire L;
    wire G;

    // Instantiate the Unit Under Test (UUT)
    five_bit_comparator uut (
        .A(A),
        .B(B),
```

```
        .E(E),
        .L(L),
        .G(G)
    );

    initial begin
        A = 0;
        B = 0;

        // Wait 100 ns for global reset to finish
        #100;
        $monitor("A = %d, B = %d, (A == B) = %d, (A < B) = %d, (A > B) = %d", A, B, E, L, G);

        A = 5'd12; B = 5'd13; #10;
        A = 5'd15; B = 5'd15; #10;
        A = 5'd0; B = 5'd9; #10;
        A = 5'd7; B = 5'd7; #10;
        A = 5'd0; B = 5'd0; #10;
        A = 5'd1; B = 5'd2; #10;
        A = 5'd2; B = 5'd1; #10;
        A = 5'd14; B = 5'd15; #10;
        $stop;
    end
endmodule
```

شبیه‌سازی

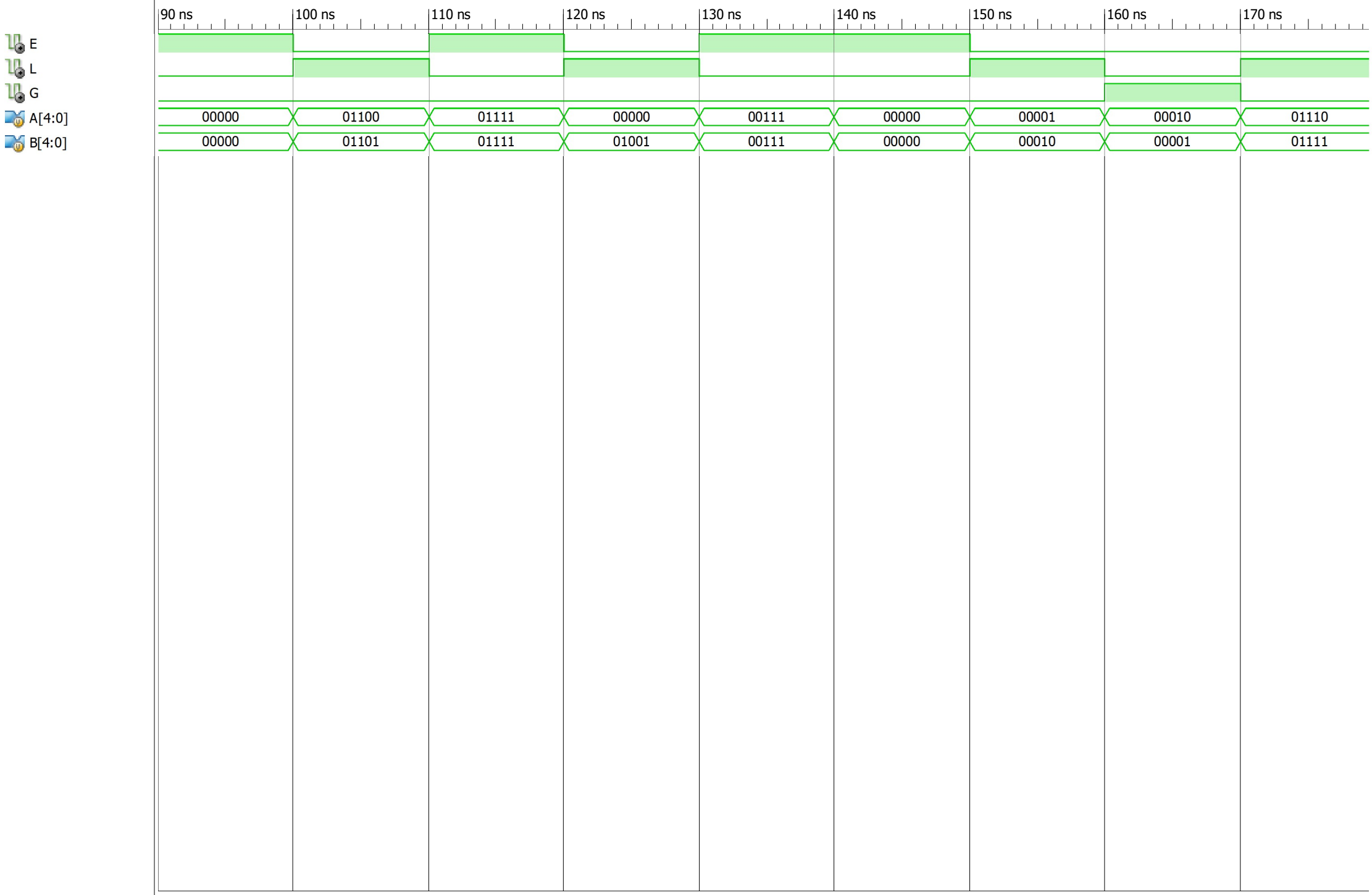
پس از شبیه‌سازی، دستور `$monitor` که در ابتدای بلاک `initial` قرار داده بودم خروجی زیبای زیر را در کنسول تولید می‌کند:

```
This is a Full version of ISim.
Time resolution is 1 ps
```

```
Simulator is doing circuit initialization process.
Finished circuit initialization process.
A = 12, B = 13, (A == B) = 0, (A < B) = 1, (A > B) = 0
A = 15, B = 15, (A == B) = 1, (A < B) = 0, (A > B) = 0
A =  0, B =  9, (A == B) = 0, (A < B) = 1, (A > B) = 0
A =  7, B =  7, (A == B) = 1, (A < B) = 0, (A > B) = 0
A =  0, B =  0, (A == B) = 1, (A < B) = 0, (A > B) = 0
A =  1, B =  2, (A == B) = 0, (A < B) = 1, (A > B) = 0
A =  2, B =  1, (A == B) = 0, (A < B) = 0, (A > B) = 1
A = 14, B = 15, (A == B) = 0, (A < B) = 1, (A > B) = 0
Stopped at time : 180 ns :  in File "C:/HW2-99109393-project/five_bit_comparator_tb.v" Line 88
```

پس مگر این که بخواهیم برای $n^2 = 32^2 = 1024$ حالت ممکن خروجی را بررسی کنیم می‌توان نتیجه گرفت که مدار ما درست کار می‌کند.

همچنین موج مدار برای این تست نیز در صفحه‌ی بعد آمده است.


90 ns
100 ns
110 ns
120 ns
130 ns
140 ns
150 ns
160 ns
170 ns
E
L
G
A[4:0]
00000
01100
01111
00000
00111
00000
00001
00010
01110
B[4:0]
00000
01101
01111
01001
00111
00000
00010
00001
01111