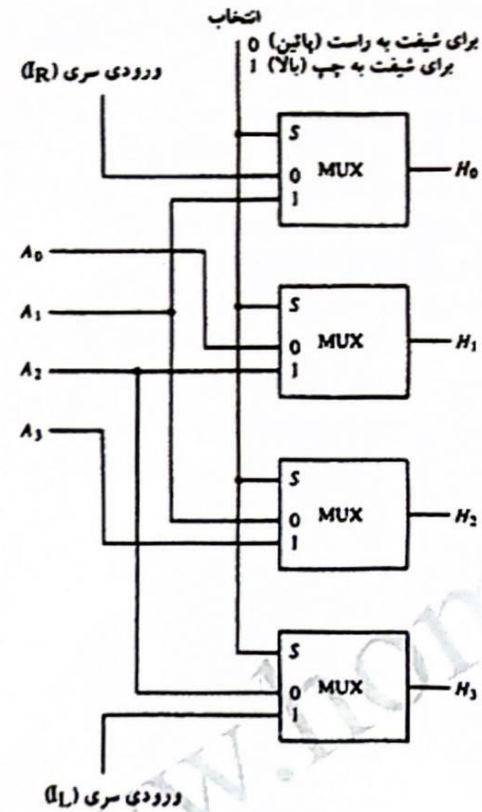


گزارش کار: *ali ghasemzadeh*

در این پروژه، ما چندین ماژول تعریف کرده‌ایم *Verilog* در زبان که شامل یک مالتی‌پلکسر، یک شیفت‌دهنده (*multiplexer*)، و یک (*shift_left*) به چپ شیفت‌دهنده به راست و یک ماژول (*shift_right*) است. (*main_module*) اصلی در ادامه به توضیح هر کدام از این ماژول‌ها می‌پردازیم:



جدول تابع				
انتخاب	خروجی			
	H_0	H_1	H_2	H_3
0	I_R	A_0	A_1	A_2
1	A_1	A_2	A_3	I_L

شکل ۴-۱۲ شیفت ترکیبی ۴ بیتی

۴-۷ واحد حساب، منطق و شیفت

ماژول *multiplexer*

یک *multiplexer* ماژول مالتی پلکسر ۲ به ۱ با عرض ۴ بیتی را پیاده سازی می کند. ورودی های این ماژول شامل دو (s) یک سیگنال انتخاب و $(in0$ و $inM)$ ورودی ۴ بیتی است. (out) خروجی ۴ بیتی به inM برابر با ۱ باشد، اگر s خروجی متصل می شود و به $in0$ غیر این صورت در خروجی متصل می شود.

```
module multiplexer (  
    input wire s,  
    input wire [3:0] in0,  
    input wire [3:0] in1,  
    output wire [3:0] out  
);  
  
assign out = s ? in1 : in0;  
  
endmodule
```

ماژول *shift_left*

یک شیفت‌دهنده *shift_left* ماژول
به چپ برای داده‌های ۴ بیتی
است. این ماژول دارای یک
ورودی (*in`*) و خروجی ۴ بیتی
است. (*out`*) خروجی ۴ بیتی
این ماژول بیت‌های ورودی را
به سمت چپ یک مکان شیفت
می‌دهد.

```
module shift_left (  
    input wire [3:0] in,  
    output wire [3:0] out  
);  
  
assign out = in << 1;  
  
endmodule
```

ماژول *shift_right*

یک *shift_right* ماژول
شیفت‌دهنده به راست برای
داده‌های ۴ بیتی است. این ماژول
(*in*) دارای یک ورودی ۴ بیتی
(*out*) و یک خروجی ۴ بیتی
است. این ماژول بیت‌های
ورودی را به سمت راست یک
مکان شیفت می‌دهد.

```
module shift_right (  
    input wire [3:0] in,  
    output wire [3:0] out  
);  
  
assign out = in >> 1;  
  
endmodule
```

ماژول *main_module*

ماژول *main_module* ماژول اصلی پروژه است که با استفاده از چهار مالتیپلکسر چندین ورودی را مدیریت می‌کند. این ماژول دارای ورودی‌های ۴ بیتی (*A0*، *A1*، *A2*، *A3*) و یک سیگنال (*IR*، *IL*) است. خروجی‌های این ماژول شامل چهار خروجی بیتی (*H0*، *H1*، *H2*، *H3*) است. در این ماژول، با توجه به ورودی‌های مختلف به خروجی (*S* سیگنال) ها متصل می‌شود.

```
module main_module (  
    input wire [3:0] A0,  
    input wire [3:0] A1,  
    input wire [3:0] A2,  
    input wire [3:0] A3,  
    input wire [3:0] IR,  
    input wire [3:0] IL,  
    input wire s,  
    output wire [3:0] H0,  
    output wire [3:0] H1,  
    output wire [3:0] H2,  
    output wire [3:0] H3  
);  
  
wire [3:0] mux1_out;  
wire [3:0] mux2_out;  
wire [3:0] mux3_out;  
wire [3:0] mux4_out;
```

```
multiplexer mux1 (  
    .s(s),  
    .in0(IR),  
    .in1(A1),  
    .out(mux1_out)  
);  
  
multiplexer mux2 (  
    .s(s),  
    .in0(A0),  
    .in1(A2),  
    .out(mux2_out)  
);  
  
multiplexer mux3 (  
    .s(s),  
    .in0(A1),  
    .in1(A3),  
    .out(mux3_out)  
);  
  
multiplexer mux4 (  
    .s(s),  
    .in0(A2),  
    .in1(IL),  
    .out(mux4_out)  
);
```

main_module

```
assign H0 = mux1_out;  
assign H1 = mux2_out;  
assign H2 = mux3_out;  
assign H3 = mux4_out;  
  
endmodule
```