

آز سیستم های دیجیتال

دکتر اجلالی

مبینا حیدری، عاطفه قندهاری، نیکا قادری
بهار ۱۴۰۳



آزمایش چهارم

طراحی *stack*

تاریخ گزارش: ۲۵ فروردین ۱۴۰۳

۱ شرح آزمایش

در این آزمایش هدف بر آن است که یک پشته چهار بیتی با ظرفیت هشت داده طراحی کنیم. این ساختار در ماژول *stack* پیاده سازی شده است. سپس در یک تست بنچ عملکرد این حافظه را مورد بررسی قرار داده ایم. در زیر ورودی ها و خروجی ها آورده شده اند:

Inputs:	Clk	Clock signal
	RstN	Reset signal
	Data_In	4-bit data into the stack
	Push	Push Command
	Pop	Pop Command
Outputs:	Data_Out	4-bit output data from stack
	Full	Full=1 indicates that the stack is full
	Empty	Empty=0 indicates that the stack is empty

۲ ماژول ها

۱.۲ *stack*

ساختار این ماژول به صورت زیر می باشد:

```

1
2 module stack (
3     output reg [3:0] Data_Out,
4     output reg Full,
5     output reg Empty
6     input Clk,
7     input RstN,
8     input [3:0] Data_In,
9     input Push,
10    input Pop,
11);
12
13
14 reg [3:0] mem [7:0];
15 reg [3:0] a = 0;
16
17 always @(posedge Clk or negedge RstN) begin
18     if (~RstN) begin
19         a <= 0;
20         Empty <= 1;
21         Full <= 0;
22     end
23     else if (Push & ~Full) begin
24         if (a == 7) Full <= 1;
25         mem[a] <= Data_In;
26         a <= a + 1;
27         Empty <= 0;
28     end
29     else if (Pop & ~Empty) begin
30         if (a == 1) Empty <= 1;
31         Data_Out <= mem[a-1];
32         a <= a-1;
33         Full <= 0;
34     end
35 end
36
37 endmodule

```

ابتدا یک آرایه هشت بیتی به نام *mem* تعریف می کنیم و سپس آن را به صورت یک وکتور چهارتایی گسترش می دهیم. همچنین یک شمارنده به نام *a* تعریف می کنیم که نشان دهنده ظرفیت کنونی پشته می باشد. در یک بلاک *always*، هرگاه که به لبه مثبت کلاک یا لبه منفی سیگنال ریست برخورد کردیم - فرض شده که ریست *activelow* می باشد - با توجه به وضعیت کنونی پشته، در صورت نیاز تغییراتی را اعمال می کنیم که به صورت زیر می باشد:

۱. اگر ریست صفر - فعال - باشد، باید پشته را خالی کنیم. در این شرایط *Full*، *a* باید صفر شوند و چون پشته خالی می شود خروجی *Empty* یک می شود.

۲. اگر بخواهیم پوش کنیم و استک پر نباشد، باید داده جدید در اندیس *a* لود شود و *a* یک واحد افزایش یابد. اگر *a* به هفت رسیده باشد، بدین معنی است که آخرین عنصر در حال وارد شدن است و از این پس باید *Full* یک شود. هم چنین اگر *Empty* از قبل صفر بوده، الان باید یک شود.

۳. هم چنین اگر پاپ یک باشد و پشته هم خالی نباشد، باید آخرین عنصر از پشته خارج شود، برای این کار کافی است که اندیس *a* را کاهش دهیم. هم چنین در صورتی که *a* به یک رسیده باشد، باید فلگ *Empty* را فعال کنیم چراکه آخرین عنصر در حال پاپ شدن می باشد. در اینجا اگر فلگ *Full* از قبل یک بوده باشد، باید صفر شود. در آخر نیز عنصری که پاپ شده را در *Data_out* قرار می دهیم.

۲.۲ testbench

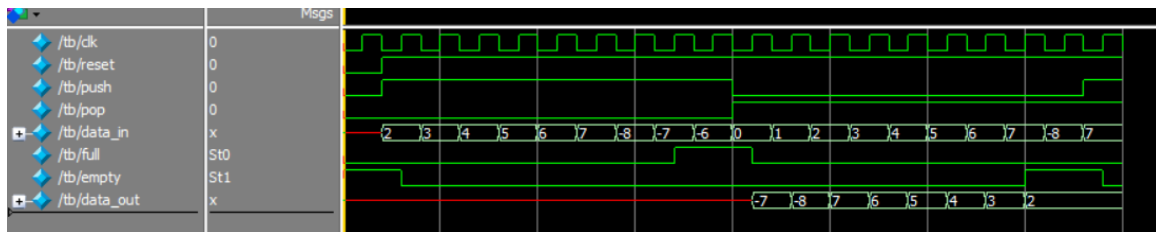
ساختار این ماژول به صورت زیر می باشد:

```

1 module tb();
2
3 reg      clk = 0, reset = 0, push = 0, pop = 0;
4 reg [3:0] data_in;
5 wire     full, empty;
6 wire [3:0] data_out;
7
8
9 stack Stack(
10     .Clk(clk),
11     .RstN(reset),
12     .Push(push),
13     .Pop(pop),
14     .Data_In(data_in),
15     .Data_Out(data_out),
16     .Full(full),
17     .Empty(empty)
18 );
19
20 always #10 clk = ~clk;
21
22 initial begin
23     $monitor("data_out = %d, Full = %d, Empty = %d", data_out, full, empty);
24     #20 reset = 1;
25     push = 1;
26     for (data_in = 2; data_in < 11; data_in = data_in + 1) begin
27         #20;
28     end
29     push = 0;
30     pop = 1;
31     for (data_in = 0; data_in < 9; data_in = data_in + 1) begin
32         #20;
33     end
34     push = 1;
35     data_in = 7;
36     #20;
37     #20 reset = 0;
38     #20;
39     $stop;
40 end
41
42 endmodule
43

```

در این ماژول ابتدا نه بار پوش می کنیم تا فلگ *Full* فعال شود، سپس هر هشت عنصر موجود در پشته را پاپ می کنیم. شرح موج این فرآیند به صورت می باشد:



هم چنین ترمینال شبیه سازی به صورت زیر می باشد:

```
VSIM 12> run
# data_out = x, Full = 0, Empty = 1
# data_out = x, Full = 0, Empty = 0
# data_out = x, Full = 1, Empty = 0
# data_out = 9, Full = 0, Empty = 0
# data_out = 8, Full = 0, Empty = 0
# data_out = 7, Full = 0, Empty = 0
# data_out = 6, Full = 0, Empty = 0
# data_out = 5, Full = 0, Empty = 0
# data_out = 4, Full = 0, Empty = 0
# data_out = 3, Full = 0, Empty = 0
# data_out = 2, Full = 0, Empty = 1
# data_out = 2, Full = 0, Empty = 0
```

۳ سنتز

گزارش سنتز مدار نیز به صورت زیر می باشد:

Flow Summary	
Flow Status	Successful - Sun Apr 14 00:41:21 2024
Quartus II 64-Bit Version	13.1.0 Build 162 10/23/2013 SJ Web Edition
Revision Name	DSDlab-HW4
Top-level Entity Name	stack
Family	Cyclone IV GX
Total logic elements	62 / 14,400 (< 1 %)
Total combinational functions	45 / 14,400 (< 1 %)
Dedicated logic registers	42 / 14,400 (< 1 %)
Total registers	42
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