

به نام خدا  
آزمایشگاه طراحی سیستم های دیجیتال  
آزمایش چهارم: توصیف رفتاری



اعضای گروه:

401105775 روزین تقی زادگان

401106096 رادین شاهدایی

401106125 باربد شهرآبادی

در این آزمایش قصد داریم با استفاده از زبان ورایلاگ یک استک با 8 خانه 4 بیتی طراحی کنیم  
سیگنال های ورودی و خروجی استک به شرح زیر میباشد:

|                 |          |   |
|-----------------|----------|---|
| <b>Inputs:</b>  | Clk      | Clock signal                              |
|                 | RstN     | Reset signal                              |
|                 | Data_In  | 4-bit data into the stack                 |
|                 | Push     | Push Command                              |
|                 | Pop      | Pop Command                               |
| <b>Outputs:</b> | 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 |

همچنین علاوه بر این سیگنال ها دو متغیر کمکی دیگر هم در برنامه تعریف کردیم:

stack\_data:

یک آرایه 8 تایی از نوع متغیر reg که وظیفه ذخیره سازی داده های push شده به استک را دارد.

sp:

نشانه گر استک (stack pointer) که همواره به اولین خانه خالی استک اشاره میکند. در ابتدا این مقدار برابر با 0 است و هر بار که مقداری به استک push شد یکی اضافه میشود و هنگامی که pop اتفاق بیفتد کم میشود. زمانی هم که استک پر باشد برابر با 8 میشود.

میخواهیم که پشته ما برای کار های push, pop و reset بصورت زیر عمل کند:

push:

اگر که استک پر نباشد، ابتدا مقدار data\_in به استک اضافه شده و سپس به مقدار sp یکی اضافه شود.

pop:

اگر که استک خالی نباشد، از مقدار sp یکی کم میشود و بالاترین داده استک به data\_out میرود.

reset:

تمامی داده های داخل استک خالی شده (برابر صفر شوند) و مقدار sp برابر صفر شود.  
شبه کد برنامه بشکل زیر میشود:

$(push == 1)(full == 0): stackData[sp] = dataIn, sp = sp + 1$

$(pop == 1)(empty == 1): sp = sp - 1, dataOut = stackData[sp]$

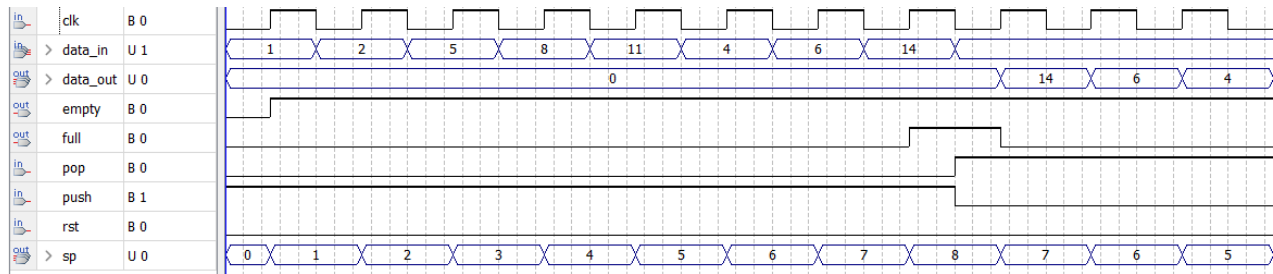
$(rst == 1): sp = 0, stackData = 0$

$full = (sp == 8), empty = !(sp == 0)$

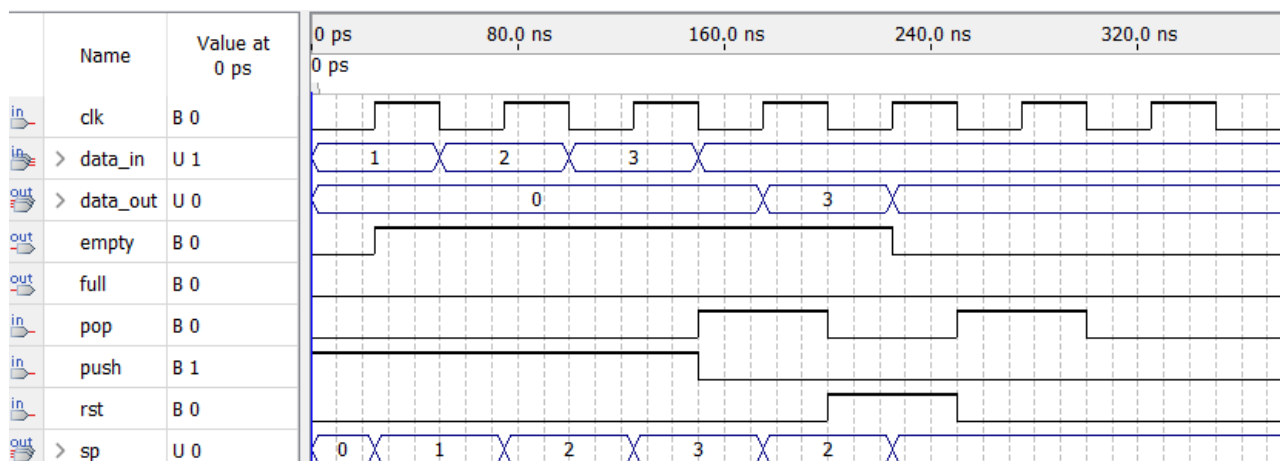
کد وریلاگ برنامه بشکل زیر میباشد:

```
module Stack(  
    input [3:0] data_in,  
    input push,  
    input pop,  
    input rst,  
    input clk,  
    output reg [3:0] data_out,  
    output reg full,  
    output reg empty,  
    output reg [3:0] sp  
);  
  
reg [3:0] stack_data[7:0];  
integer i;  
always @(posedge clk) begin  
    if(rst == 1) begin  
        for(i = 0; i < 8; i = i + 1) begin  
            stack_data[i] = 0;  
        end  
        sp = 0;  
        empty = 0;  
        full = 0;  
        data_out = 0;  
    end  
    if(push == 1 && full == 0) begin  
        stack_data[sp] = data_in;  
        sp = sp + 1;  
    end  
    if(pop == 1 && empty == 1) begin  
        sp = sp - 1;  
        data_out = stack_data[sp];  
    end  
    if(sp == 8) begin  
        full = 1;  
    end else begin  
        full = 0;  
    end  
    if(sp == 0) begin  
        empty = 0;  
    end else begin  
        empty = 1;  
    end  
end  
endmodule
```

حال برنامه را تست میکنیم:



در هشت کلاک اول هشت عدد مختلف را به استک push میکنیم. مشاهده میشود که سیگنال full برابر با یک میشود و مقدار نشانه گر هم هر بار یکی اضافه میشود. پس از آن سیگنال pop برابر 1 میشود و از آن پس در هر کلاک بالاترین عدد داخل استک در data\_out نمایش داده میشود و sp کم میشود.



در این تست در سه کلاک اول سه عدد را به استک وارد میکنیم. سپس یک بار pop میکنیم و آخرین عدد وارد شده در استک، نمایش داده میشود و sp یکی کم میشود. پس از آن استک را ریست میکنیم و همانطور که مشاهده میشود sp برابر با صفر شده و داده های داخل استک خالی میشود. پس از آن وقتی pop کنیم چون چیزی در استک نیست مقداری نمایش داده نمیشود.