

به نام خدا



آزمایشگاه طراحی سیستم‌های دیجیتال

آزمایش چهارم: توصیف رفتاری

نویسندگان:

زهره عباسی لیاپی - ۹۸۱۰۵۸۸۱

محمد حسین دولت‌آبادی - ۹۸۱۰۵۷۷۳

استاد:

دکتر اجلالی

دستیار آموزشی:

آقای سیادت‌زاده

فهرست مطالب

۳ توضیح آزمایش
۴ پشته
۵ پیاده‌سازی
۶ سنجش درستی مدار

توضیح آزمایش

در این آزمایش قصد داریم داده‌ساختار پشته را با استفاده از توصیف رفتاری در وریلاگ پیاده‌سازی کنیم. عمق پشته‌ی مد نظر ۸ و پهنای آن ۴ است. (در پیاده‌سازی ما با توجه به استفاده از `parameter` این عدد به راحتی قابل تغییر است و قابلیت استفاده برای پشته‌ای به عمق و پهنای متفاوت هم دارد.)

پشته

ورودی‌ها:

Clk: کلاک مدار

RstN: در صورت فعال بودن این سیگنال، مدار ریست، داده‌ها پاک و پشته خالی می‌شود.

Data_In: مقدار ورودی پشته است که توسط ما داده می‌شود.

Push: در صورت فعال بودن این سیگنال، داده‌ی موجود در Data_In با بررسی شروط لازم در صورت امکان وارد پشته می‌شود.

Pop: در صورتی که داده‌ای درون پشته وجود داشته باشد، با این سیگنال بالاترین داده‌ی موجود در پشته از آن خارج و وارد Data_Out می‌شود.

خروجی‌ها:

Data_Out: مقدار خروجی Pop شده از پشته درون آن قرار می‌گیرد.

Full: در صورتی که پشته پر شده باشد، این سیگنال فعال می‌شود.

Empty: در صورتی که پشته خالی باشد این سیگنال صفر می‌شود.

پیاده‌سازی

پیاده‌سازی با استفاده از توصیف رفتاری، با ریست اکتیو لو و آسنکرون انجام شده است. با هر لبه‌ی بالا یا پایین‌رونده‌ی کلاک خروجی‌ها مشخص می‌شوند. برای حرکت در آرایه از یک متغیر به نام ptr استفاده می‌کنیم که index آرایه را در آن ذخیره کنیم. در صورت ریست شدن مدار، پشته خالی شده و مقادیر full و empty صفر می‌شود. همچنین مقدار ptr صفر می‌شود تا به خانه‌ی اول آرایه اشاره کند. همچنین اگر مقدار Data_out معتبر نباشد، به آن مقدار z را نسبت می‌دهیم. شکل ۱ توضیحات مربوطه را نشان می‌دهد.

```

1
2   `timescale 1ns / 1ps
3
4   module stack(Data_Out, Full, Empty, Clk, RstN, Data_In, Push, Pop);
5       parameter DEPTH = 8;
6       parameter WIDTH = 4;
7
8       output reg Empty, Full;
9       output reg [WIDTH - 1:0] Data_Out;
10
11
12       input [WIDTH - 1:0] Data_In;
13       input Clk, RstN, Push, Pop;
14
15       integer ptr;
16       reg [WIDTH - 1:0] data [0:DEPTH - 1];
17
18       always @(posedge Clk or negedge RstN) begin
19
20           if(!RstN) begin
21               ptr <= 0;
22               Full <= 0;
23               Empty <= 0;
24               Data_Out <= {WIDTH{1'bz}};
25           end
26       end

```

شکل ۱

در غیر این صورت اگر پشته پر نباشد و عملیات push انجام شود، مقدار data_in درون پشته قرار می‌گیرد. و اگر پشته خالی نباشد و یک pop شود، آخرین مقدار وارد شده به پشته، از آن خارج می‌شود و درون data_out قرار می‌گیرد. شکل ۲ نشانگر این توضیحات است.

```

26       else begin
27           Data_Out <= {WIDTH{1'bz}};
28           if(Push && !Full) begin
29               Empty <= 1;
30               Full <= (ptr === DEPTH - 1) ? 1 : 0;
31
32               data[ptr] <= Data_In;
33               ptr <= ptr + 1;
34           end
35           else if(Pop && Empty) begin
36               Full <= 0;
37               Empty <= ptr === 1 ? 0 : 1;
38               Data_Out <= data[ptr - 1];
39
40               ptr <= ptr - 1;
41           end
42       end
43   end
44 end
45
46 endmodule

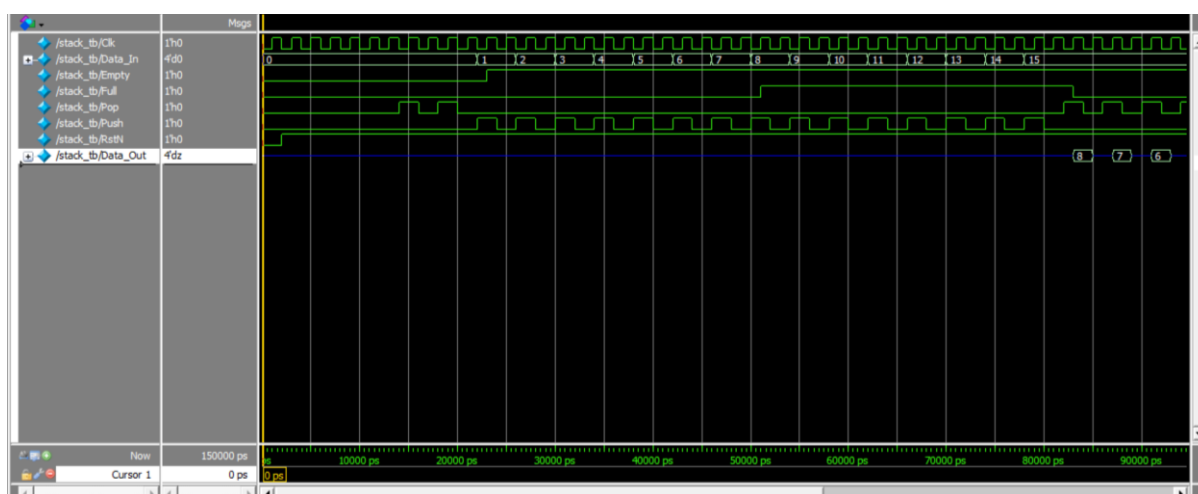
```

شکل ۲

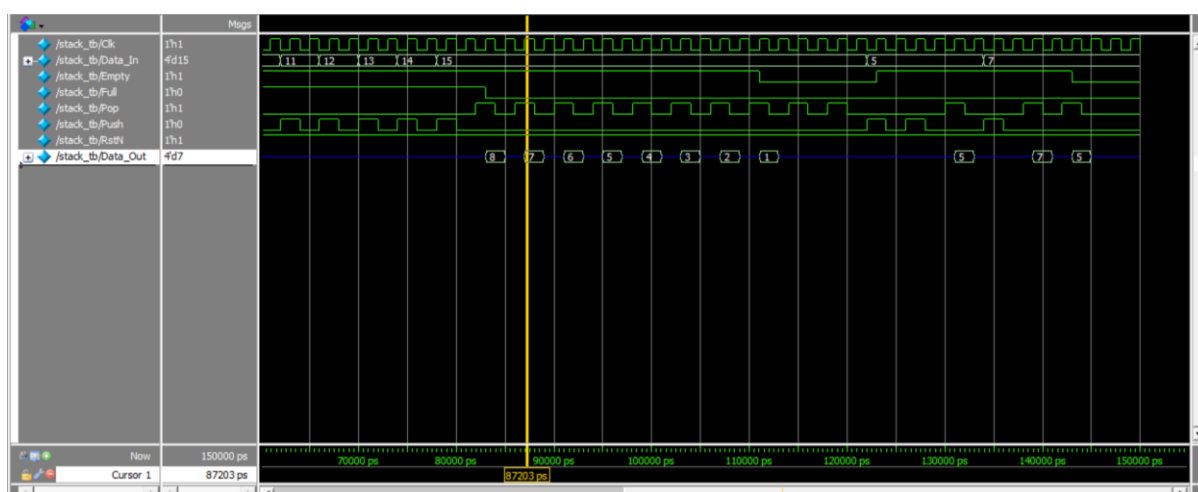
سنجش درستی مدار

با تست‌بنچ طراحی شده برای ماژول استک، می‌توانیم درستی کار آن را بررسی کنیم. ابتدا مدار را ریست می‌کنیم. سپس در حالی که پشته خالی است سعی می‌کنیم از آن عملیات pop را انجام دهیم که چیزی به خروجی نمی‌دهد. سپس اعداد ۱ تا ۱۵ را به ورودی می‌دهیم که بعد از ورود عدد ۸، دیگر چیزی وارد پشته نشده و سیگنال full ۱ می‌شود.

پس از آن با pop کردن به ترتیب اعداد ۸ تا ۱ به خروجی داده می‌شوند و سیگنال empty صفر می‌شود که نشان‌دهنده‌ی خالی بودن پشته است. پس از آن دیگر چیزی از پشته پس از pop شدن خارج نمی‌شود. شکل ۳ و ۴ تمامی موارد مطرح‌شده را به درستی نشان می‌دهند.



شکل ۳



شکل ۴