

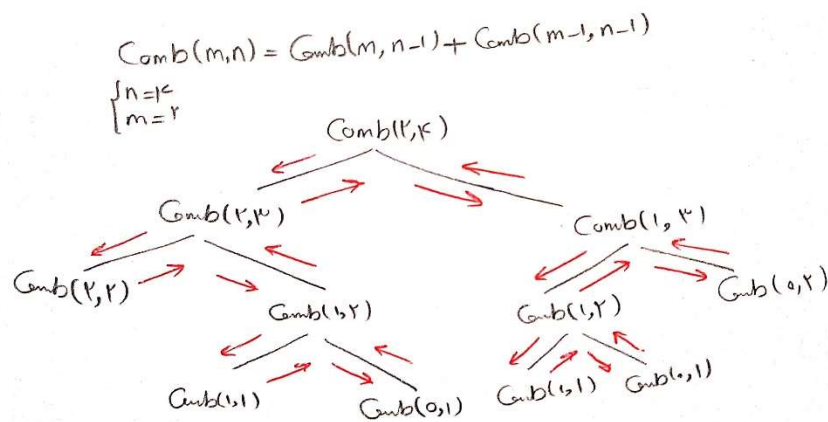
به نام خدا

پرnian فاضل 810198516

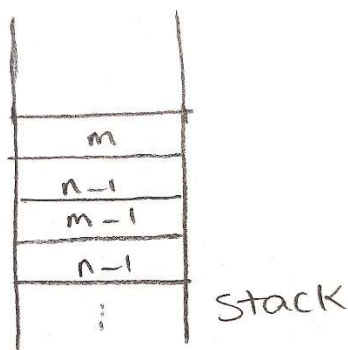
امتحان میان ترم درس طراحی سیستم ها و دیجیتال

دکتر مدرسی

ایده‌ی کلی من برای حل این سوال استفاده از روشی که در تمرین شماره 3 گفته شد می باشد. به این صورت که در مسیر داده یک مایجول Accumulator داریم که وقتی به شرط تابع بازگشتی میرسیم چون مقدار بازگشتی آن 1 است، یک واحد به Accumulator اضافه می کنیم. در اینجا شر بازگشت هنگامی است که  $m==0$  یا  $m==n$  باشد. برای پوش کردن در استک من برای هر  $n$  و  $m$  که به عنوان ورودی در تابع بازگشتی هستند، مقادیر  $m$  و  $n-1$  و  $m-1$  و  $n-1$  را در استک پوش می کنم. برای توضیح دلیل این روش مثال زیر را که به صورت درختی تابع بازگشتی را برای  $Comb(2,4)$  نشان می دهد را در نظر بگیرید:



درواقع در هر مرحله از وارد شدن به تابع بازگشتی 2 مقدار بالای استک را پاپ میکنم و این 2 مقدار را به عنوان 2 ورودی تابع جدید بازگشتی در نظر میگیرم. این کار (پاپ کردن 2 مقدار بالا) را تا جایی ادامه می دهم که به یکی از شروط بازگشت تابع برسم، که در این صورت علاوه بر پاپ کردن آن 2 مقدار از تابع بازگشتی مقدار Accumulator را یک واحد زیاد میکنم. برای واضح تر شدن منظورم از وضعیت استک در هر تابع بازگشتی بع از پاپ کردن 2 مقدار  $n$  و  $m$  به عنوان ورودی های آن تابع مقادیر زیر به استک اضافه می شوند و بالای استک به صورت زیر خواهد بود:



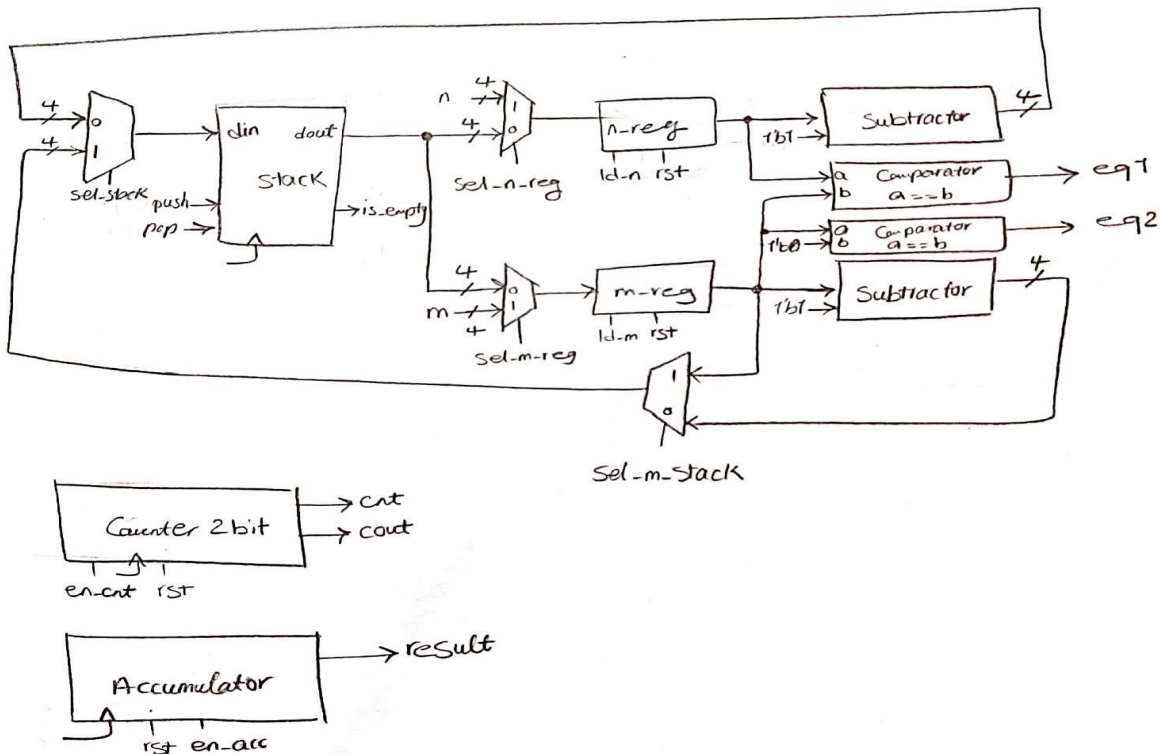
```

8
9  #include <iostream>
10
11 using namespace std;
12
13
14 int comb (int m, int n){
15
16     if (m==0 || m==n)
17
18         return 1;
19     else
20
21         return comb(m, n-1) + comb(m-1, n-1);
22 }

```

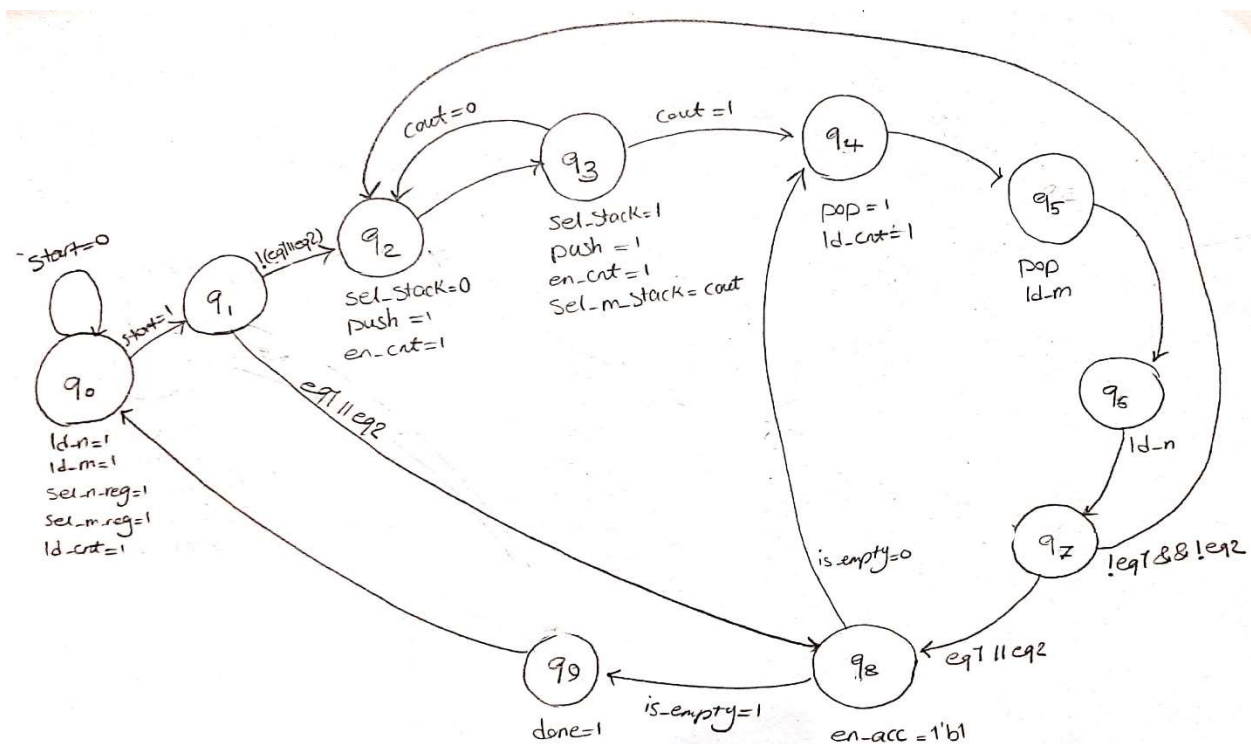
حال کنترلر و مسیر داده را نشان می دهیم:

مسیر داده:



در این طراحی برای مسیر داده من 2 رجیستر برای ذخیره کردن 2 مقدار بالای استک در نظر گرفتیمو همچنین دو subtractor تا بتوانم پارامتر های تابع بازگشتی بعدی را ایجاد کنم. این 2 ورودی را از طریق 2 مالتی پلکسر با سلکت های sel\_stack و sel\_m\_reg به استک پوش می کنم. برای استک 2 سیگنال کنترلی ورودی push و pop و یک سیگنال خروجی is\_empty در نظر گرفتم که نشان میدهد آیا استک خالی است یا نه. برای چک کردن اینکه شرایط بازگشت تابع برقرار است یا نه از 2 مقایسه کننده با خروجی های eq1 و eq2 استفاده کردم که این ها ورودی کنترلر هستند. ماجول accumulator نشان داده شده که جواب نهایی مقدار ذخیره شده در آن است. همچنین برای 2 بار پوش کردن ورودی های تابع بازگشتی بعدی از یک شمارنده با 2 بیت استفاده کردم که از خروجی c\_out آن رد کنترلر برای مشخص کردن اینکه شمارنده یک دور شمارش کرده استفاده میکنم. توجه کنید که تعداد بیت خروجی در Accumulator را برابر 13 بیت در نظر گرفتم چون بیشترین مقدار برای حالت  $m=7$  و  $n=8$  رخ میدهد که به باینری به 13 بیت نیاز دارد.

## کنترلر:



در طراحی این کنترلر من از سیگنال های خروجی مسیر داده استفاده کردم مثلا برای مشخص کردن اینکه آیا شرط تابع بازگشتی برقرار است یا خیر که با سیگنال های eq1 و eq2 این کار را انجام دادم. در اینجا تا زمانی

که سیگنال start یک نشده flow انجام نمیشود و همچنین هنگامی که مقدار Accumulator جواب نهایی مسئله را دارد سیگنال done را یک می کنم. پوش کردن ورودی های توابع بازگشتی در استیت های q2 و q3 انجا میشوند. در استیت q8 و q7 پس از پاپ کردن 2 مقدار بالای استک شرط های تابع بازگشتی و اینکه آیا استک خالی است را چک میکنم.

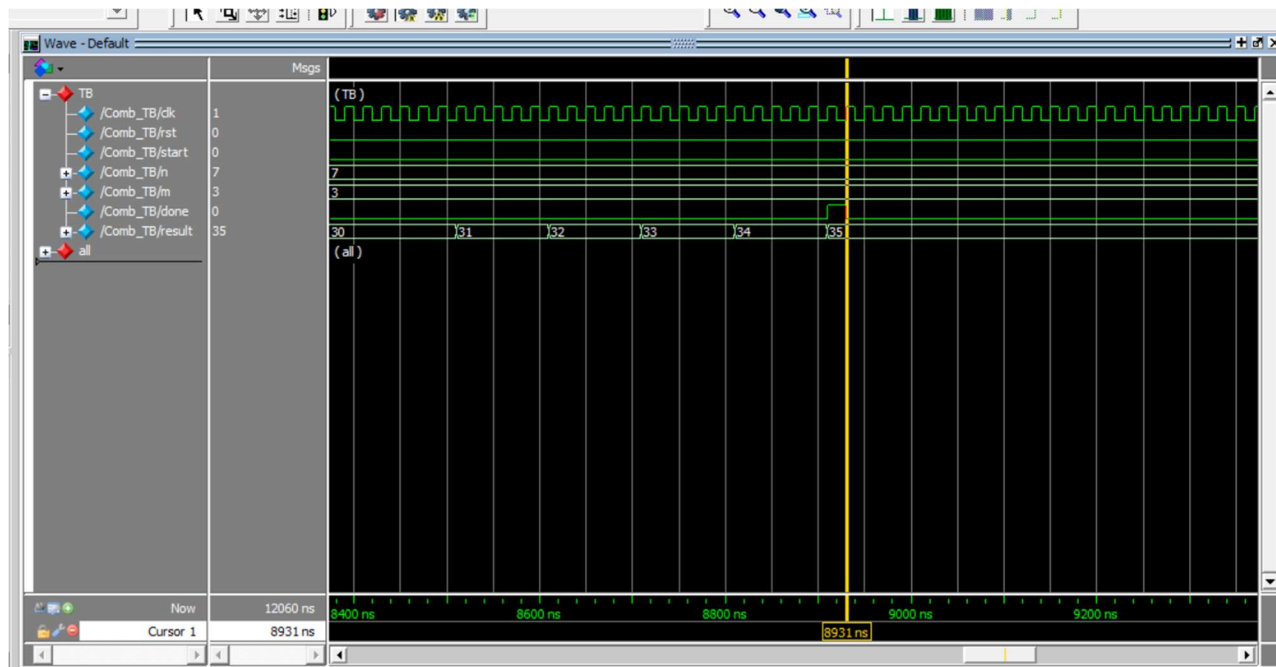
خروجی waveform چند نمونه تست:

n=6 , m=2:



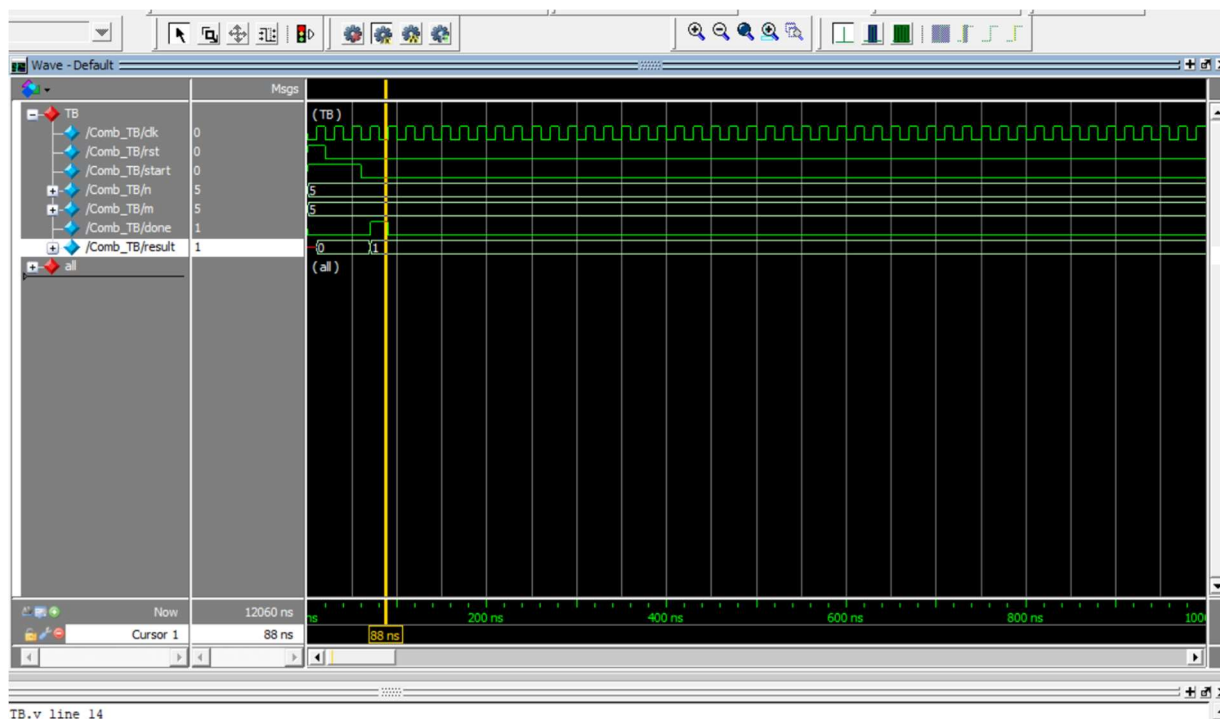
همانطور که مشاهده میشود مقدار 15 هنگامی که سیگنال done یک شده روی سیگنال خروجی قرار میگیرد.

$n=7, m=3$



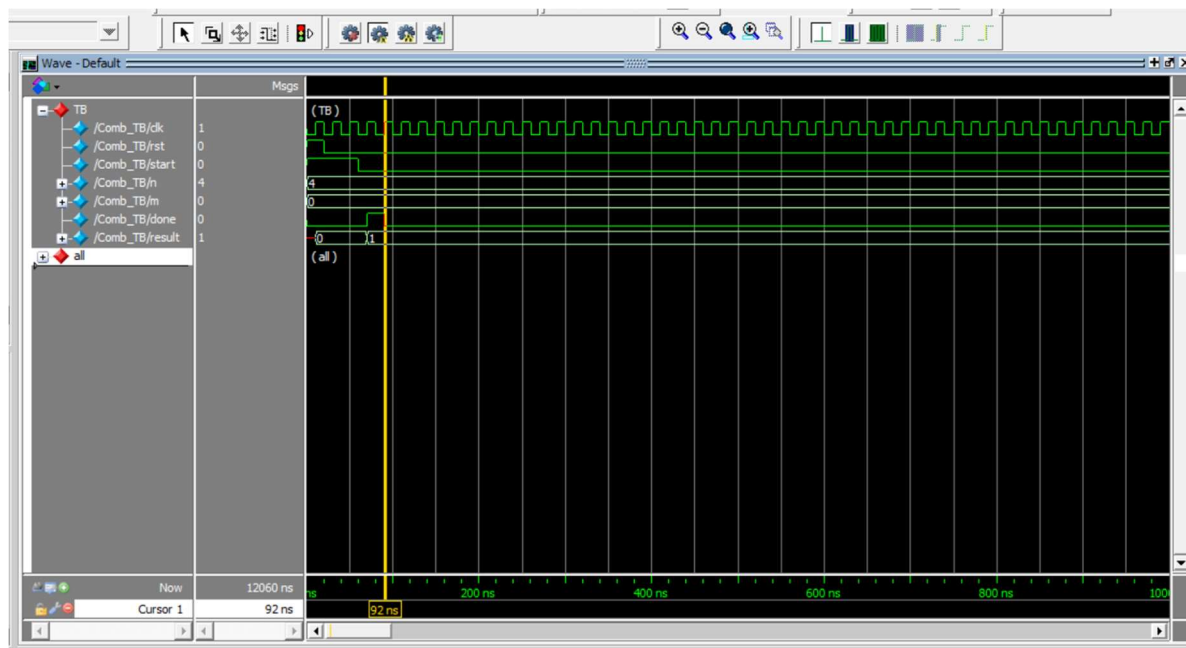
همانطور که مشاهده میشود مقدار 35 هنگامی که سیگنال done یک شده روی سیگنال خروجی قرار میگیرد.

$n=5, m=5$ :



همانطور که مشاهده میشود مقدار 1 هنگامی که سیگنال done یک شده روی سیگنال خروجی قرار میگیرد.

n=4, m=0:



همانطور که مشاهده میشود مقدار 1 هنگامی که سیگنال done یک شده روی سیگنال خروجی قرار میگیرد.

توجه شود که در موارد بالا من مقدار خروجی با radix برابر unsigned نشان دادم.