

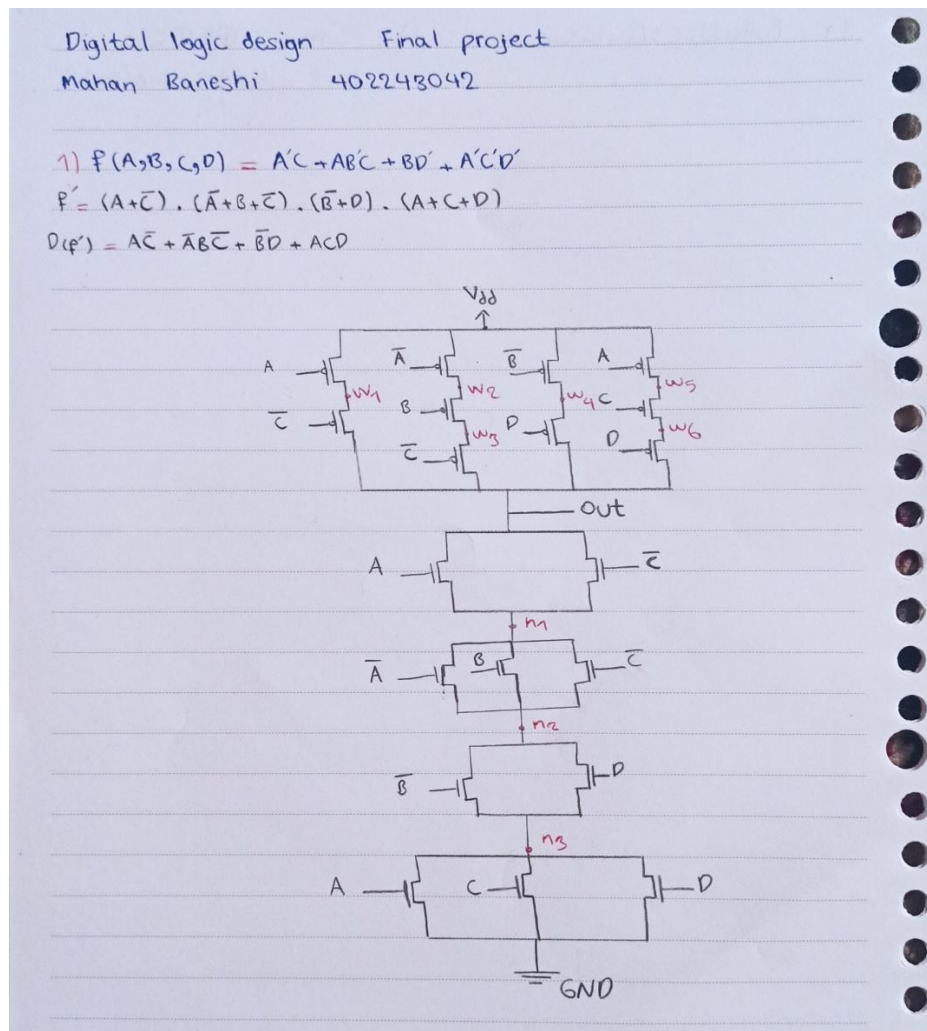
به نام خدا
گزارش کار پروژه مدار منطقی پاییز 1403

ماهان بانشی

سوال 1:

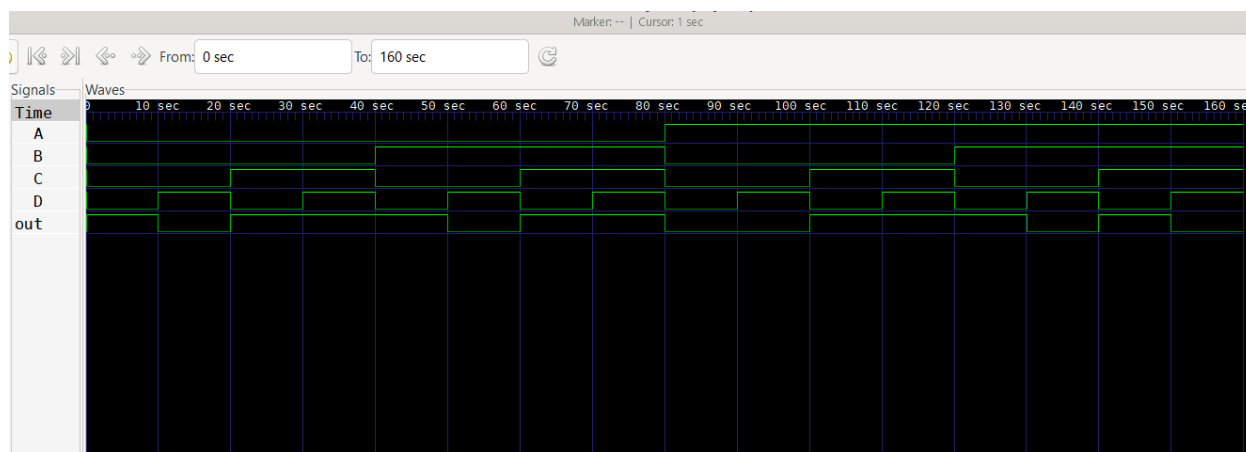
الگوریتم حل سوال:

چون گفته شده است که در سطح ترانزیستور پیاده سازی کنید ابتدا باید شکل کاغذی مدار را با منطق CMOS پیاده سازی میکنیم.



سپس آن شکل را با تعریف تعدادی ترانزیستور PMOS و NMOS بصورت کد در می آوریم.
سپس برای تست بنچ هم 16 حالتی که 4 متغیر دارند را با تاخیر های مشخص می دهیم و نمودار سیگنالی آن را از gtkwave می خوانیم.

نمایش خروجی:

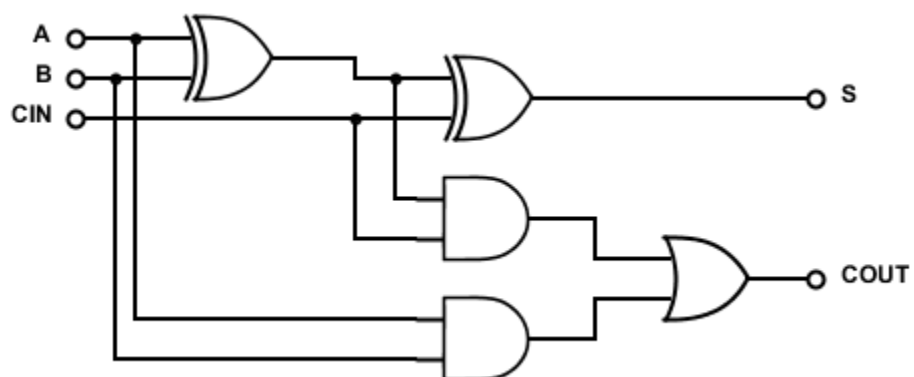


سوال 2:

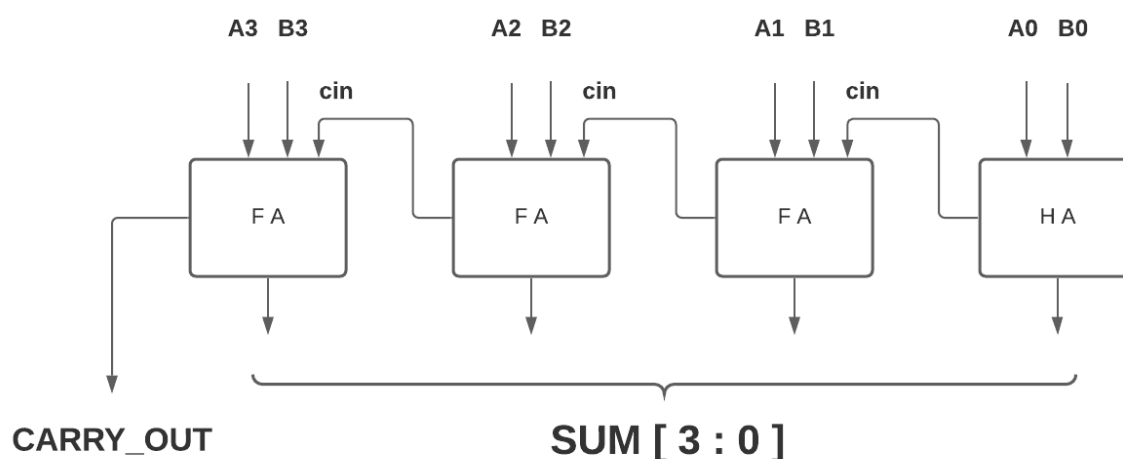
پیاده سازی جمع کننده ی 4 بیتی در سطح gate level:

الگوریتم حل سوال:

ابتدا یک ماژول جمع کننده ی یک بیتی (full adder) در یک تابع طراحی می کنیم. برای تعریف این تابع خروجی های sum و cout را بر حسب ضوابطی از ورودی ها که a و b و Cin بدست می آوریم.



سپس 4 تا full adder را پشت سر هم قرار می دهیم تا جمع کننده 4 بیتی شود. در سمت راست ترین full adder مقدار Cin را صفر قرار می دهیم. در بقیه ی full adder ها Cin را به مقدار Cout در full adder قبلی متصل می کنیم.



مسئله ی Overflow: چون اعداد علامت دار هستند سمت چپ ترین بیت علامت است که اگر صفر باشد عدد مثبت و اگر یک باشد عدد منفی است. سرزیر در صورتی رخ می دهد که دو عدد هم علامت باشند ولی در جمع شدنشان بیت علامت برعکس شود.

سرریز شدت یا نشدن را هم به راحتی با یک ضابطه بدست می آوریم. (سر ارائه توضیح داده خواهد شد)

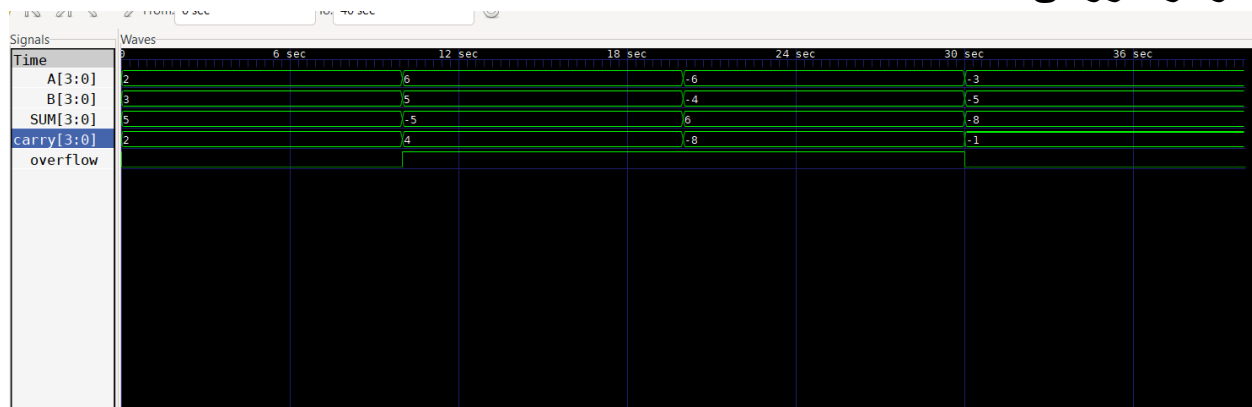
خروجی تست بنچ:

```

[Running] q2_tb.v
VCD info: dumpfile q2.vcd opened for output.
A = 0010, B = 0011, SUM = 0101, Overflow = 0
A = 0110, B = 0101, SUM = 1011, Overflow = 1
A = 1010, B = 1100, SUM = 0110, Overflow = 1
A = 1101, B = 1011, SUM = 1000, Overflow = 0
q2_tb.v:42: $finish called at 40 (1s)
[Done] exit with code=0 in 0.17 seconds

```

نمودار خروجی:



سوال 3:

هدف: پیاده سازی یک ALU شش بیتی با قابلیت انجام 6 عملیات در سطح dataflow برای این کار ابتدا هر یک از اعمالی که ALU انجام می دهد را به صورت یک ماژول تعریف می کنیم که دشواری خاصی ندارد. سپس برای هر ماژول یک نمونه می سازیم و آن را به عملگر خواسته شده پاس می دهیم.

در ضمن در این سوال برای راحت شدن کارمان ورودی های A و B و همچنین Out را بصورت علامت دار تعریف کرده ایم.
نمودار خروجی:



خروجی کنسول:

```
[Running] q3_tb.v
VCD info: dumpfile q3.vcd opened for output.
Operator: 00, A = 15, B = 5, Out = -2
Operator: 01, A = 12, B = 8, Out = -28
Operator: 10, A = 7, B = -4, Out = 4
Operator: 11, A = -9, B = 2, Out = 20
q3_tb.v:42: $stop called at 40 (1s)
** VVP Stop(0) **
** Flushing output streams.
** Current simulation time is 40 ticks.
```