



پروژه وریلاگ مدار منطقی

مهلت تحویل : 1401/04/03



بهار 1401

- پیاده‌سازی پروژه به صورت انفرادی یا تیم‌های 2 نفری خواهد بود.
- پوشه پروژه را فشرده (zip یا rar) کرده و در بخشی که در سامانه vu ایجاد شده بارگذاری کنید. اسم فایل باید به فرمت StuNumber1_StuNumber2_VP (در صورت گروهی زدن پروژه) و یا StuNumber_VP باشد. هردو عضو یک گروه باید فایل پروژه را بارگذاری کنند.
- دریافت فایل‌های پروژه از طریق روش‌هایی مانند تلگرام، واتساپ و ... امکان پذیر نیست.
- در صورت مشاهده هرگونه تقلب چه در بخش اصلی و چه در بخش امتیازی، نمره کل پروژه تمام نمراتی که تقلب آنها اثبات شده، منفی صد (100-) لحاظ می‌شود.
- زمان ارائه پروژه پس از اتمام زمان ارسال فایل‌ها اعلام خواهد شد.

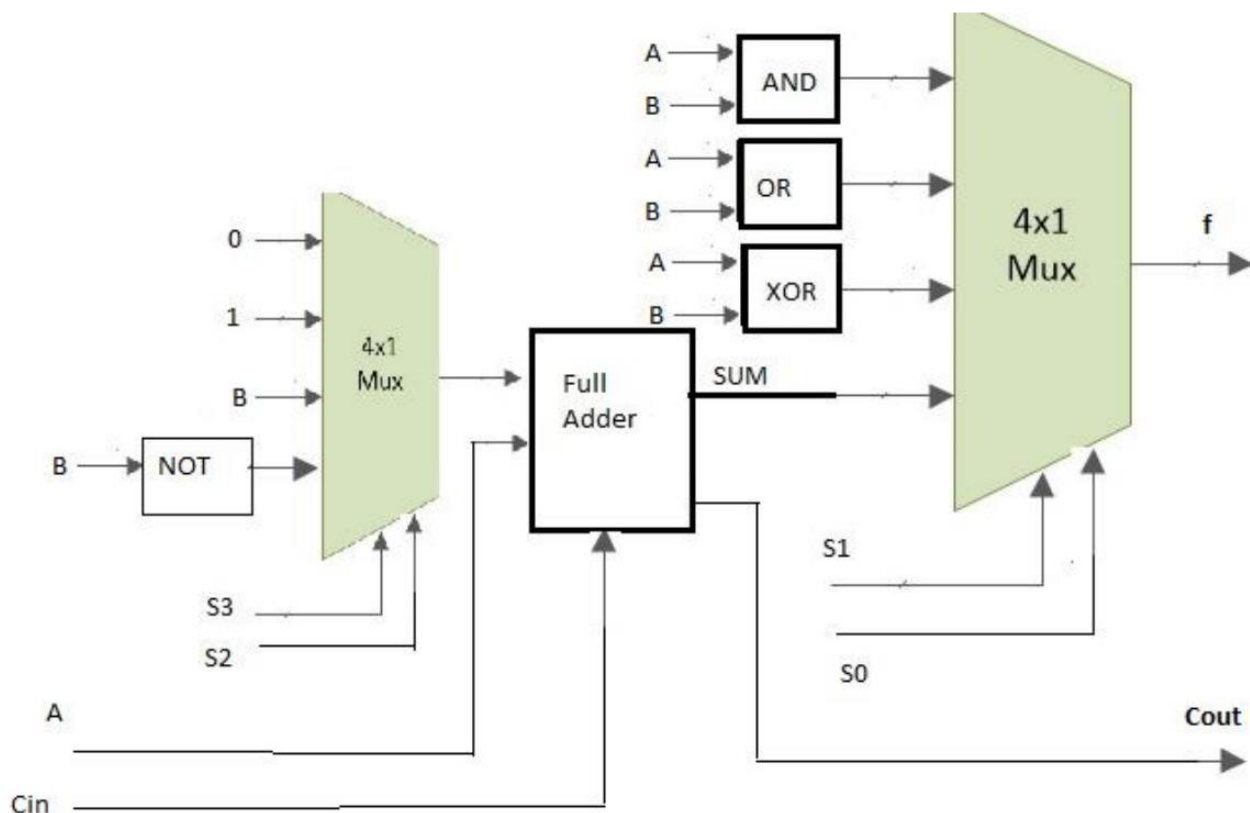
مقدمه: واحد محاسبه و منطق (Arithmetic Logic Unit) یا ALU، بخش مهمی از یک پردازنده را تشکیل می‌دهد که اعمال پایه منطقی و حسابی را بر روی داده‌ها (عملوندها) به انجام می‌رساند. در این پروژه می‌خواهیم یک ALU ساده طراحی کنیم.

فاز اول: یک تمام جمع کننده به صورت ساختاری (gate-level) و یک مالتی‌پلکسر 4 به 1 (دارای دو خط انتخاب) به صورت رفتاری (Behavioral) در وریلاگ طراحی کنید.

فاز دوم: یک ALU ساده تک بیتی به شکل زیر و با نمونه‌گیری از قطعات و ماژولهای طراحی شده قبل و به شکل ساختاری طراحی کنید.

ورودی‌ها: A,B,Cin,S[3:0]

خروجی‌ها: Cout,f



خطوط انتخاب $S[0:3]$ را به صورت vector تعریف نمایید. این خطوط طبق جدول زیر خروجی ALU را تعیین می کنند.

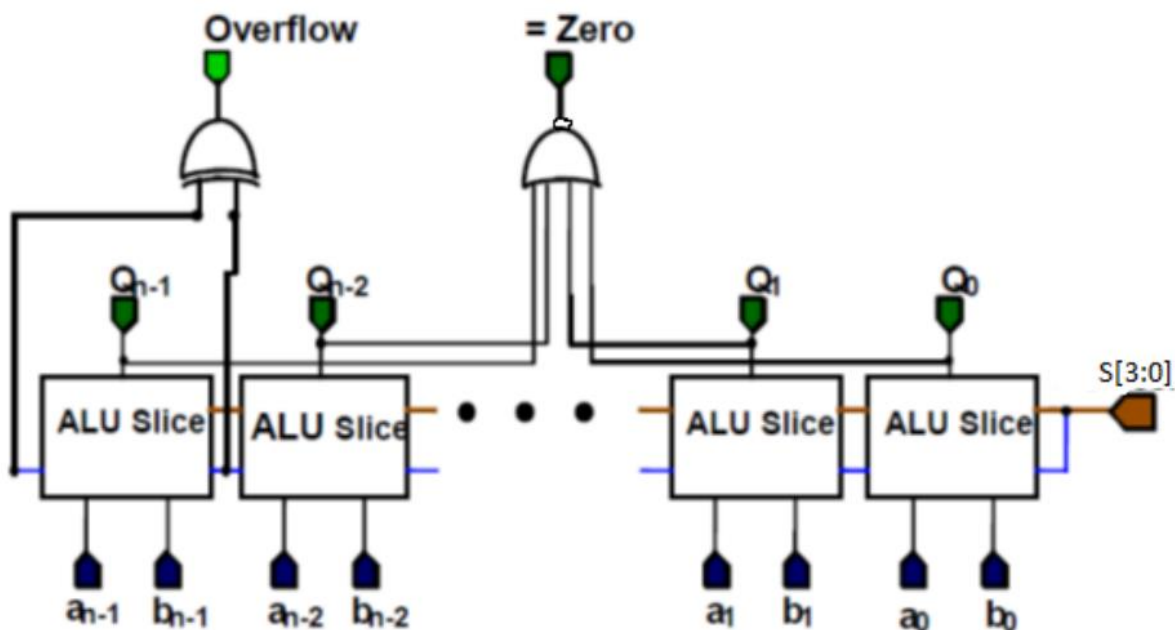
"جدول عملکرد ALU"

S3	S2	S1	S0	Cin	f	Explanation
0	0	0	0	X	$A.B$	AND
0	0	0	1	X	$A+B$	OR
0	0	1	0	X	$A \text{ XOR } B$	XOR
0	0	1	1	0	A	transfer input A
0	0	1	1	1	$A+1$	increment A
0	1	1	1	0	$A-1$	Decrement A
0	1	1	1	1	A	transfer input A
1	0	1	1	0	$A+B$	ADD
1	0	1	1	1	$A+B+1$	ADD plus 1
1	1	1	1	0	$A+B'$	ADD to 1's Complement of B
1	1	1	1	1	$A+B'+1$	ADD to 2's Complement of B (A-B)

فاز سوم: با نمونه‌گیری از چهار ALU طراحی شده در فاز دوم، و یک گیت NOR چهار ورودی و یک XOR دو ورودی، یک ALU چهاربیتی به شکل زیر طراحی کنید. (به صورت ساختاری طراحی شود).

ورودی‌ها: دو vector چهاربیتی A و B، یک بیت Cin، یک vector چهاربیتی S[3:0] که به خطوط S[3:0] از هر برش تک بیتی ALU متصل می‌شود. برای اولین برش ALU، ورودی Cin کل به Cin متصل می‌شود، و برای بقیه برش‌ها، Cin به Cout مرحله قبل وصل می‌شود.

خروجی‌ها: یک بردار (vector) چهاربیتی F، یک خروجی Zero (که در صورت صفر بودن F، مقدار Zero برابر 1 می‌شود). و خروجی overflow و Cout.



- ورودی Cin و خروجی Cout در شکل بالا رسم نشده است.

❖ نمره اضافه: پیاده سازی فاز سوم به صورت رفتاری، دارای نمره اضافه است.

❖ نمره اضافه: نوشتن تست بنچ (test bench) برای نمایش کارهای مختلف ALU دارای نمره اضافه است.

موفق باشید - تیم طراحی پروژه