به نام خدا

دانشگاه صنعتی شریف

دانشکده مهندسی کامپیوتر

آزمایشگاه معماری کامپیوتر

آزمایش پنجم:

طراحی واحد محاسبه

اطلاعات تیم	
نام اعضا	شماره دانشجویی
متین داغیانی	۹۸۱۰۶۴۵۶
بردیا محمدی	۹۸۱۷۱۱۰۴
محمدجواد هزاره	۹۸۱۰۱۰۷۴

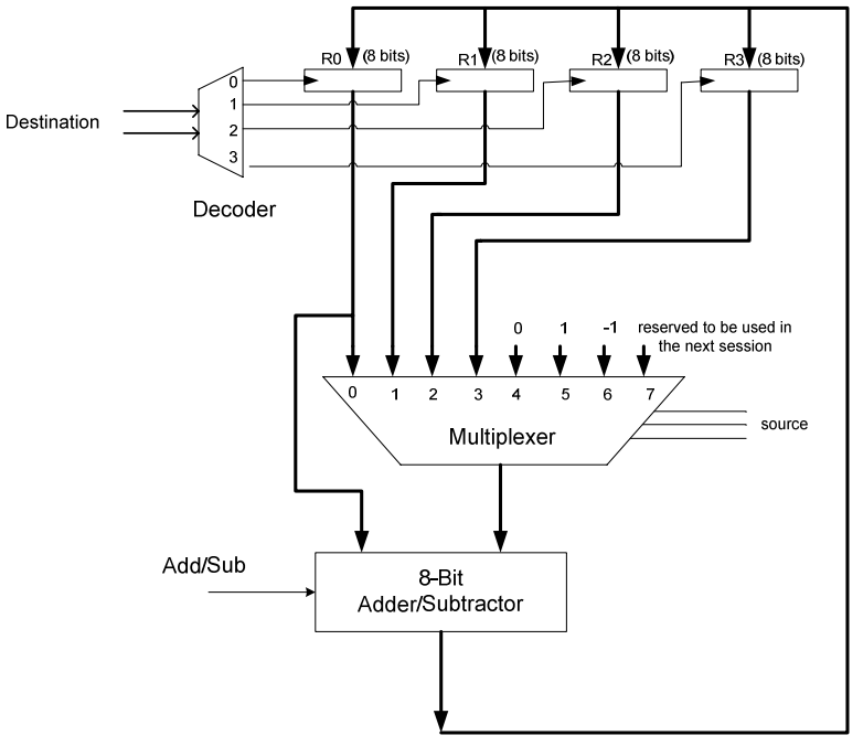
پاییز ۱۴۰۰

فهرست مطالب

فهرست تصاویر

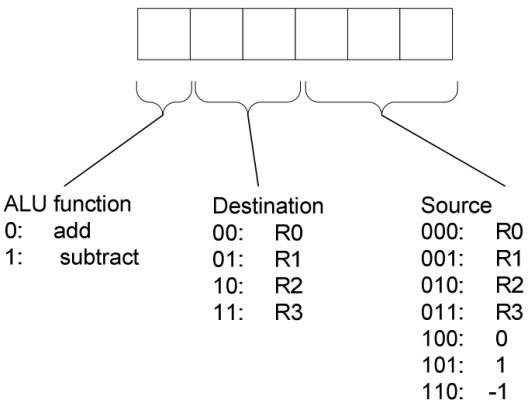
۱ هدف آزمایش

هدف این آزمایش پیاده‌سازی واحد محاسبه‌ی یک کامپیوتر بود. این واحد محاسبه که معماری آن در شکل ۱ آمده است، از توانایی جمع و تفریق هشت بیتی برخوردار است. عملیات‌هایی که این واحد می‌تواند انجام دهد به فرم $d = r \pm s$ بوده که d آدرس ثبات مقصد و s آدرس ثبات مبدا خواهد بود. r نیز آدرسی ثابت است که به یکی از ثبات‌ها اشاره می‌کند. (در این مدار این ثباتِ ثابتْ همان R_0 است.)



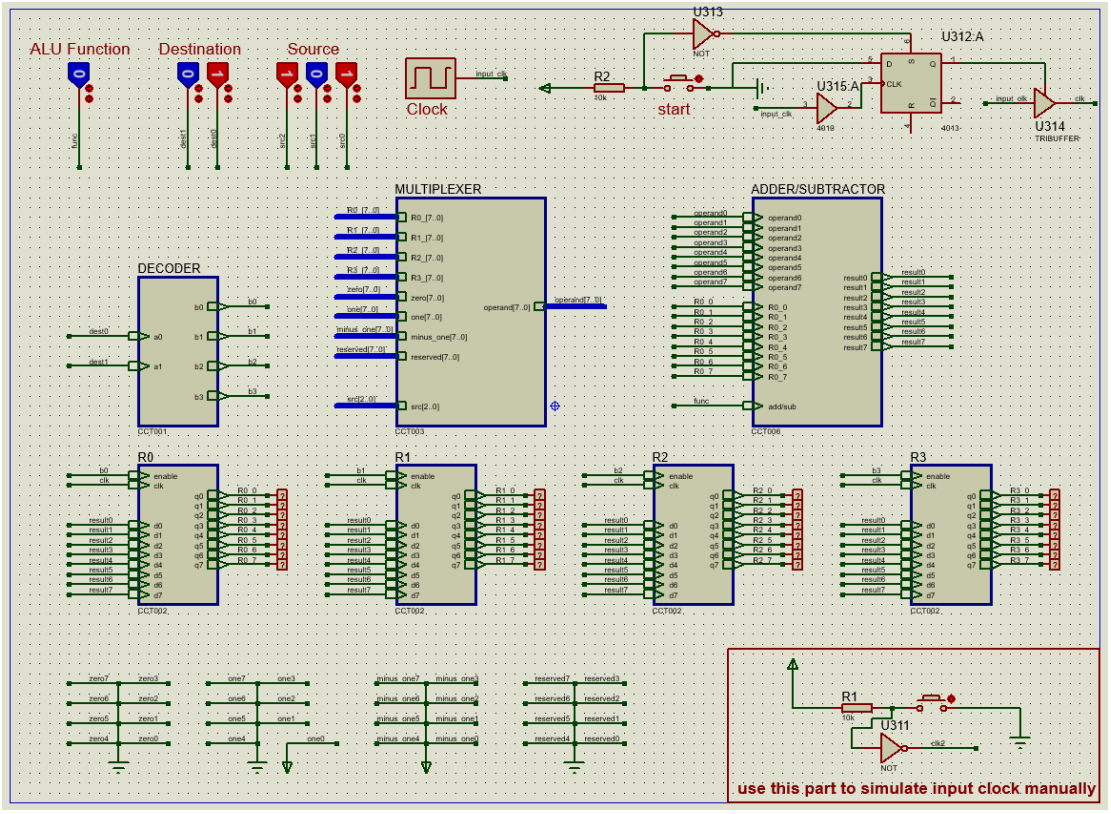
شکل ۱: معماری واحد محاسبه

معماری دستورالعمل‌های این ماشین نیز در شکل ۲ آمده است.



شکل ۲: معماری مجموعه دستورالعمل‌ها

همانطور که در تصویر دیده می‌شود بیت اول دستورالعمل مشخص کننده‌ی نوع عملیات بوده، دو بیت بعدی آدرس ثبات مقصد را مشخص می‌کنند و سه بیت آخر نیز آدرس ثبات مبدا هستند. نمای کلی مدار طراحی شده نیز در شکل ۳ آمده است.



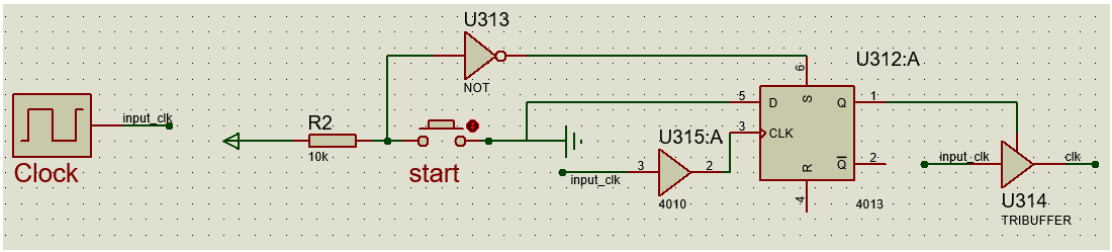
شکل ۳: نمای کلی مدار واحد محاسبه

در بخش بعدی به مراحل طراحی این مدار و بررسی طراحی داخلی ماژول‌های آن خواهیم پرداخت.

۲ طراحی و پیاده‌سازی مدار

ماژول‌های مورد نیاز و شروع به کار مدار

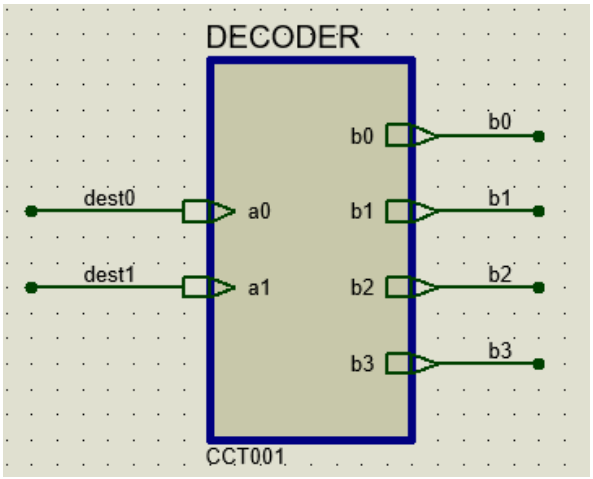
همانطور که در شکل ۱ برای معماری مدار دیده می‌شود، به یک دیکودر برای تعیین آدرس ثبات مقصد، چهار ثبات هشت بیتی که قابلیت بارگذاری داشته باشند، یک مالتی پلکسر برای انتخاب ثبات مبدا و در نهایت ماژول جمع و تفریق کننده نیاز داریم. با فشردن کلید $start$ مدار آغاز به کار کرده و محاسبه‌ی خود را انجام می‌دهد. با توجه به این که مدار ترکیبی است محاسبه‌ی آن یک کلاک به طول می‌انجام بنابراین کلاک داخلی آن را که باعث فعال شدن رجیسترها می‌شود با اندکی تاخیر پس از آمدن کلاک غیرفعال می‌کنیم. این فرآیند در قسمتی از مدار که در شکل زیر آمده است انجام شده است.



شکل ۴: کلید $start$ و شروع به کار مدار

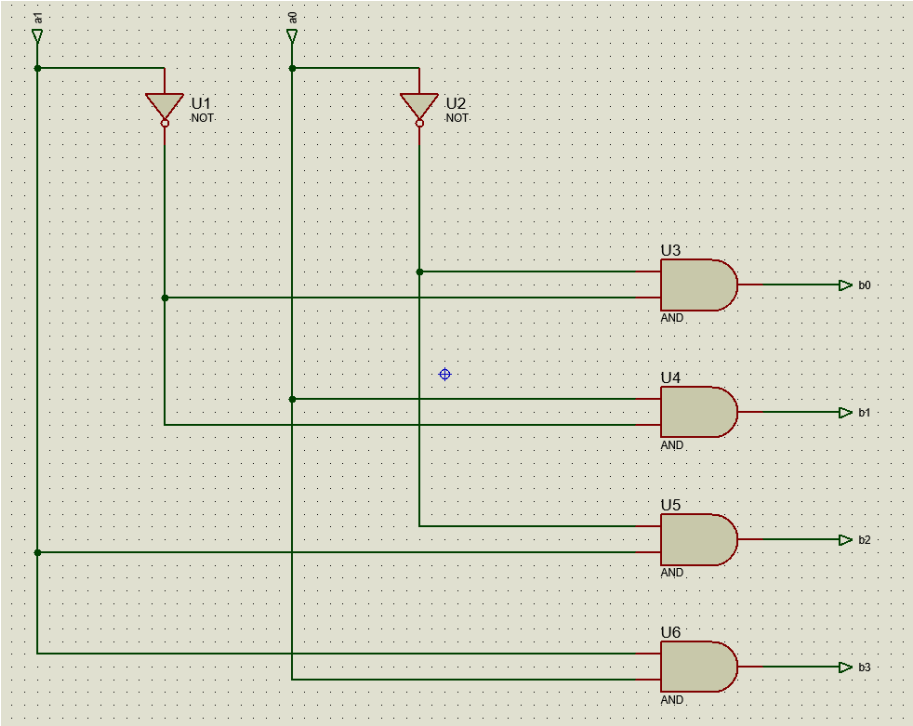
۱.۲ ماژول DECODER

همانطور که قبل‌تر توضیح داده شد، این ماژول وظیفه‌ی تعیین ثبات مقصد را از روی آدرس آن بر عهده دارد. خروجی آن به عنوان سیگنال enable به رجیسترها داده شده تا در صورت فعال بودن این سیگنال، رجیستر موردنظر فعال شده و عملیات نوشتن روی آن انجام شود. رابط ورودی و خروجی این ماژول را در شکل ۵ می‌توان دید.



شکل ۵: ورودی و خروجی‌های DECODER

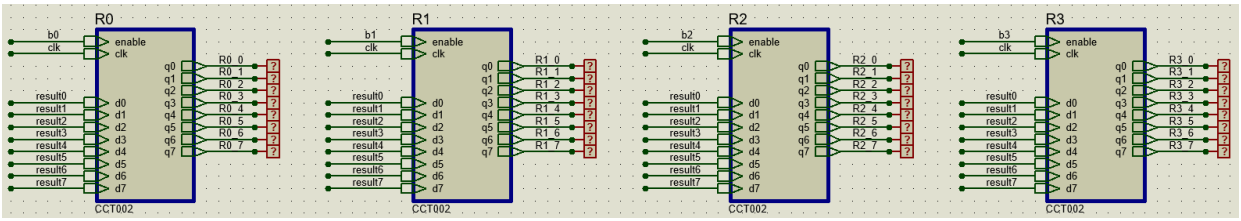
همانطور که دیده می‌شود ورودی‌های آن بیت‌های دوم و سوم دستورالعمل هستند که با نام‌های $dest_0$ و $dest_1$ نام‌گذاری شده‌اند. طراحی داخلی این ماژول نیز به صورت شکل ۶ است که به همان روش رایج ساخت دیکودر می‌باشد.



شکل ۶: طراحی داخلی ماژول DECODER

۲.۲ ثبات‌ها

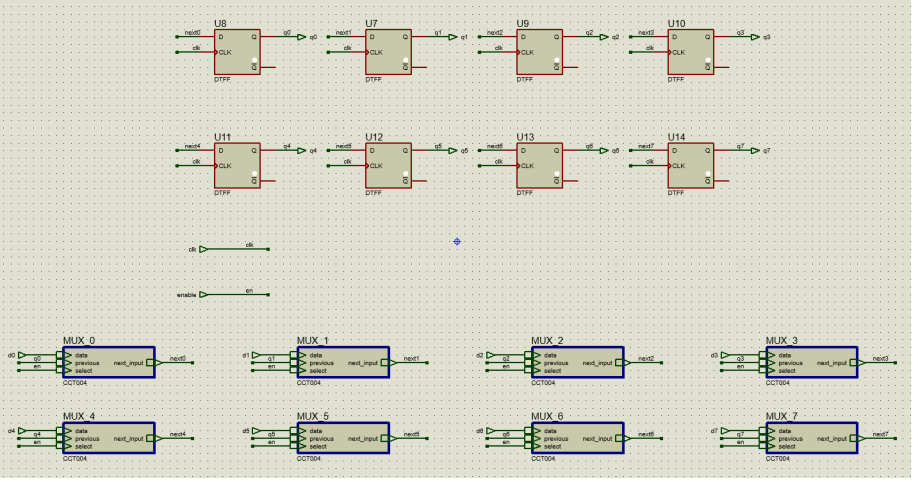
این ماژول‌ها که در مدار با نام R_0 تا R_3 نام‌گذاری شده‌اند، ثبات‌های کامپیوتر را تشکیل می‌دهند. ورودی‌ها و خروجی‌های آن‌ها را در شکل ۷ می‌توان دید.



شکل ۷: ورودی و خروجی‌های ثبات‌ها

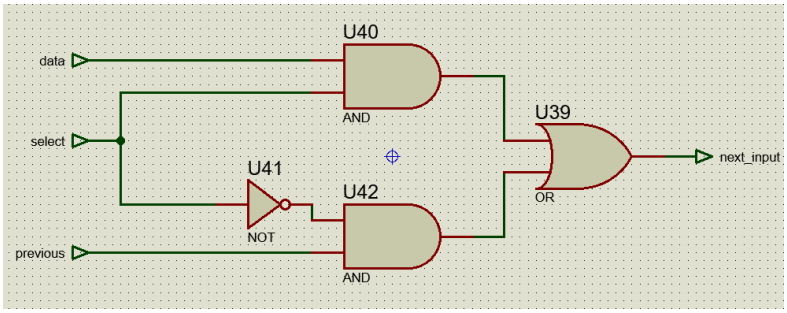
همانطور که در تصویر دیده می‌شود هر ماژول علاوه بر ورودی‌های d_0 تا d_7، ورودی $enable$ نیز دارد که با فعال شدن این سیگنال، ورودی‌های d_0 تا d_7 در رجیسترهای این ماژول بارگذاری می‌شود. خروجی آن‌ها نیز همواره فعال و مقدار ذخیره شده در رجیسترهای آن‌ها را نشان می‌دهد.

برای طراحی داخلی هر یک از آن‌ها نیز از هشت فلیپ‌فلاپ نوع D استفاده شده که طراحی آن را در شکل ۸ می‌توان مشاهده کرد.



شکل ۸: طراحی داخلی ثبات‌ها

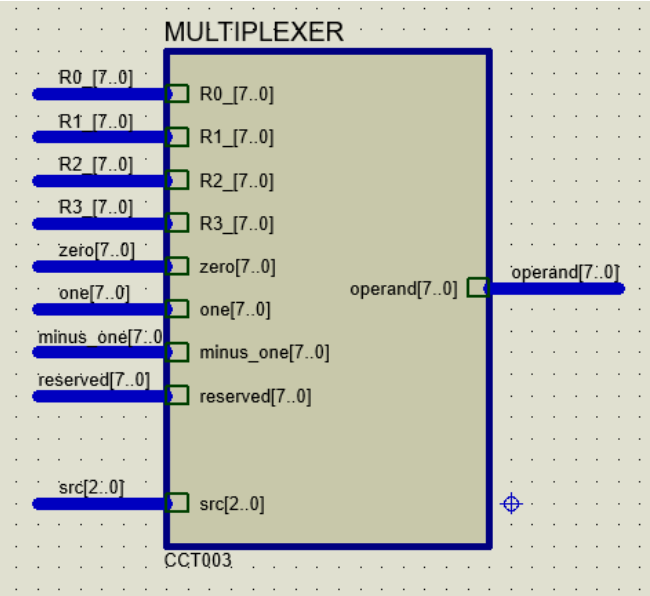
هم‌چنین از یک مالتی‌پلکسر برای مشخص کردن ورودی هر فلیپ‌فلاپ استفاده شده که تا زمانی که سیگنال $enable$ فعال نباشد، با هر کلاک رجیسترها همان مقدار قبلی خود را در خود نگه دارند. طراحی داخلی هر یک از این مالتی‌پلکسرهای ۲ به۱ را در شکل ۹ می‌توان دید.



شکل ۹: طراحی داخلی مالتی‌پلکسر ۲ به۱

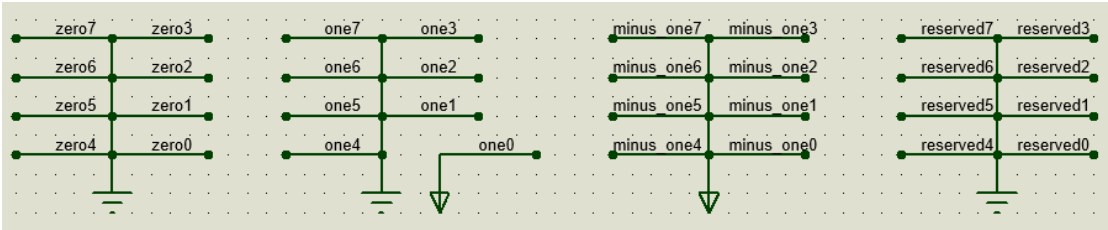
۳.۲ ماژول MULTIPLEXER

از این ماژول برای تعیین یکی از عملوندهای عملیت‌ها استفاده می‌شود. همانطور که قبلا اشاره شد، یکی از عملوندهای این واحد محاسبه، ثابت بوده و از ثبات R_0 می‌آید و دیگری با استفاده از آدرسی که در دستورالعمل آمده است مشخص می‌شود. ورودی‌ها و خروجی‌های این ماژول را در شکل ۱۰ می‌توان مشاهده کرد.



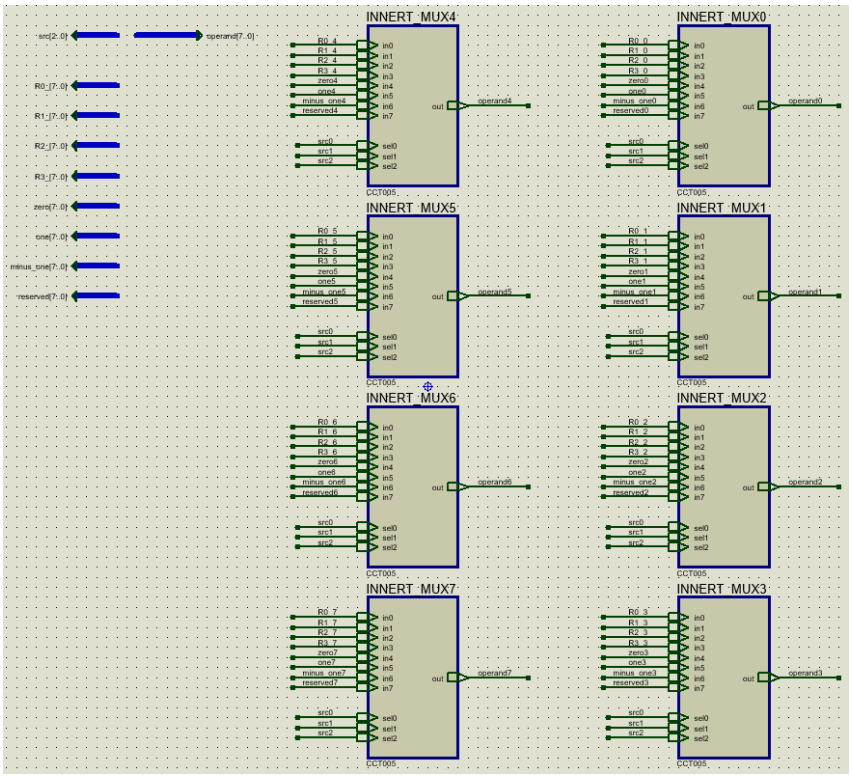
شکل ۱۰: ورودی و خروجی‌های MULTIPLEXER

این ماژول یک مالتی‌پلکسر ۸به۱ است که در واقع هر ورودی آن خود هشت بیتی بوده و خروجی آن نیز هشت بیتی است. سیگنال انتخاب‌کننده‌ی آن نیز همانطور که در تصویر مشخص است از سه بیت آخر دستورالعمل می‌آید که با باس src مشخص شده است. پیش از پرداختن به طراحی داخلی مدار باید توجه کنیم که این واحد توانایی جمع و تفریق با مقادیر ثابت 0، 1 و -1 را نیز داراست که به ترتیب اگر آدرس مبدا برابر 4، 5 و 6 باشد این عملیات صورت می‌گیرد. برای همین منظور سه باس $zero$، one و $minus\_one$ به ورودی‌های ۴ تا ۶ این ماژول داده شده‌اند. ساخت این باس‌ها نیز مطابق شکل ۱۱ صورت انجام شده است.



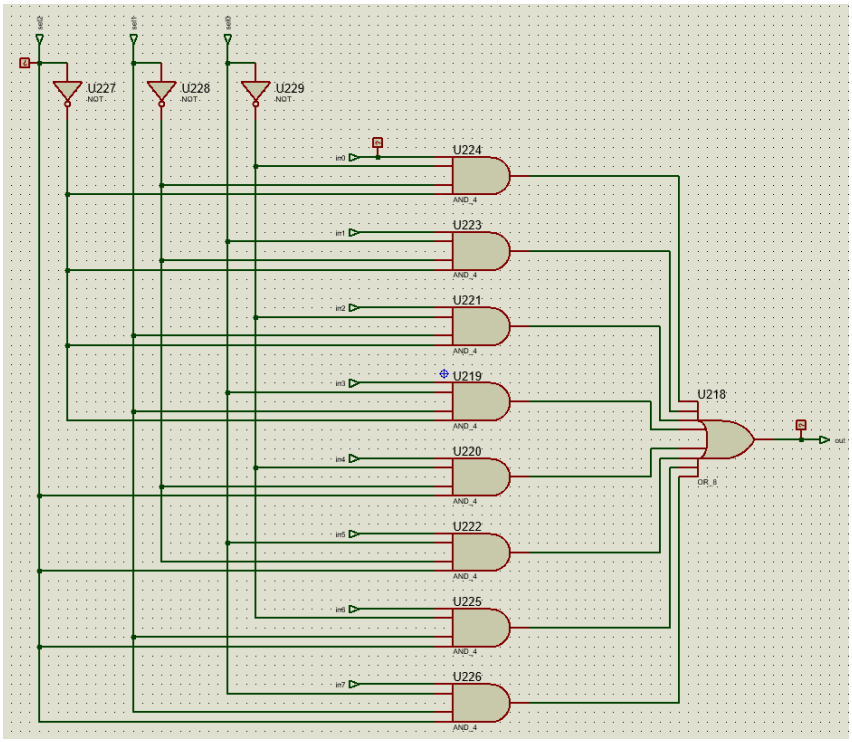
شکل ۱۱: عملوندهای ثابت

در طراحی داخلی ماژول نیز با توجه به سیگنال سلکتور، هر بیت ورودی‌های هشت بیتی به صورت جداگانه بررسی شده و انتخاب شده است. به عبارتی به ازای هر بیت سیگنال‌های ورودی، از یک مالتی‌پلکسر ۸به۱ استفاده شده که ورودی‌ها و خروجی آن یک بیتی است. طراحی داخلی این ماژل در شکل ۱۲ آمده است.



شکل ۱۲: طراحی داخلی ماژول MULTIPLEXER

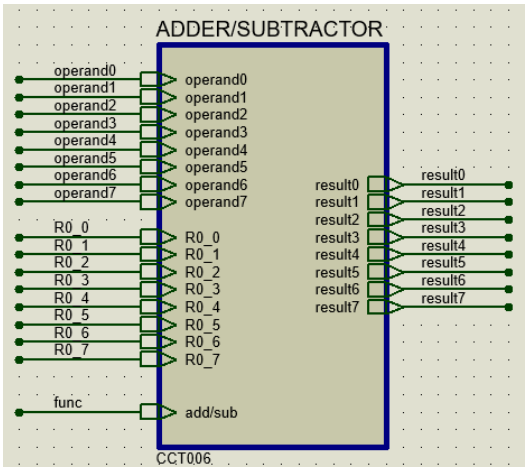
طراحی داخلی مالتی‌پلکسرهای ۸به۱ یک بیتی نیز در شکل ۱۳ آمده است.



شکل ۱۳: طراحی داخلی ماژول‌های INNERT_MUX استفاده شده در شکل ۱۲

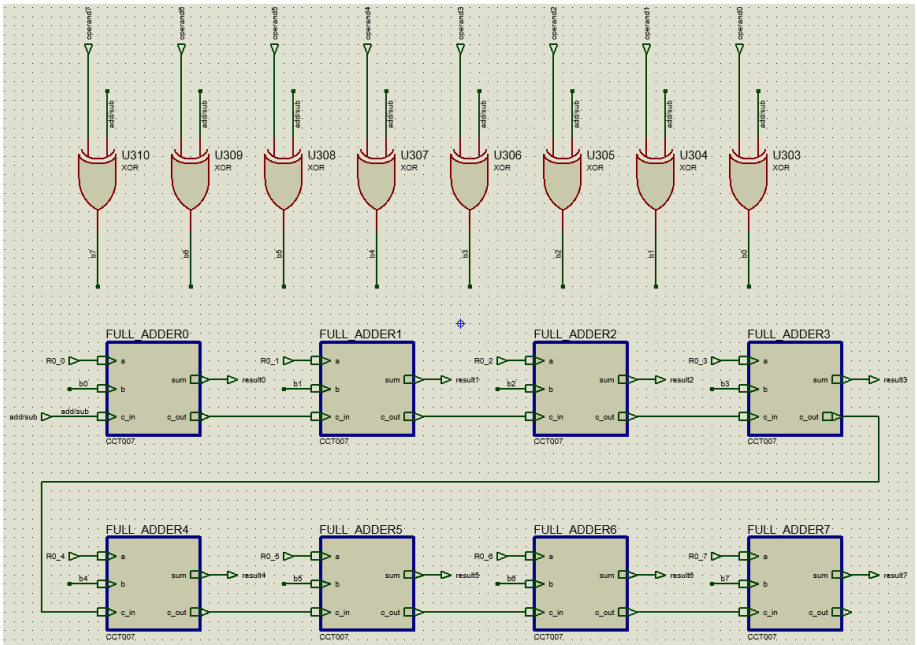
۴.۲ ماژول ADDER/SUBTRACOTR

این ماژول نیز وظیفه‌ی اصلی محاسبه را انجام می‌دهد. ورودی‌های آن هشت بیت رجیستر R_0 است که یکی از عملوندهای عملیات را تشکیل داده و هشت ورودی دیگر از خروجی ماژول MULTIPLEXER آمده‌اند که عملوند دوم عملیات را تشکیل می‌دهند. ورودی‌ها و خروجی‌های این مدار در شکل ۱۴ نشان داده شده است.



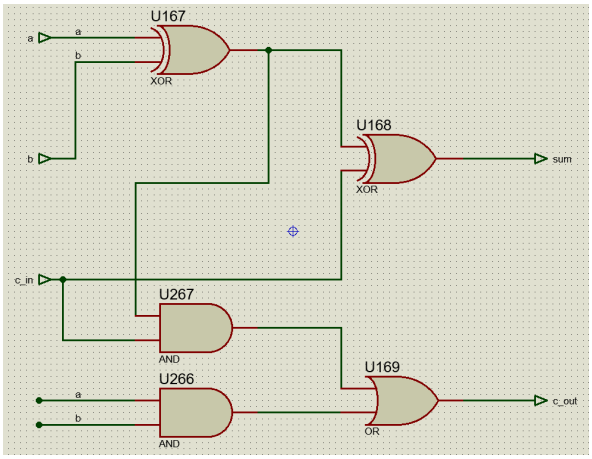
شکل ۱۴: ورودی و خروجی‌های ADDER/SUBTRACTOR

سیگنال add/sub نیز به بیت اول دستورالعمل متصل شده که مشخص می‌کند عملیات جمع یا تفریق است. برای طراحی داخلی این ماژول نیز از جمع‌کننده‌ی کامل (Full Adder) استفاده شده که در شکل ۱۵ نیز طراحی داخلی آن نشان داده شده است.



شکل ۱۵: طراحی داخلی ماژول ADDER/SUBTRACTOR

برای تفریق از روش جمع با متمم دو استفاده شده که برای متمم دو کردن عملوندنیز بیت‌های آن در صورت یک بودن سیگنال add/sub نات می‌شوند که این کار با XOR کردن این بیت‌ها با سیگنال add/sub صورت می‌گیرد و با اضافه شدن همین سیگنال به c_in فول ادر اول، عدد مورد نظر متمم دو می‌شود. در ادامه جمع به صورت سری با استفاده از فول ادر ها انجام می‌شود. طراحی داخلی فول ادرها نیز در شکل ۱۶ نمایش داده شده است.



شکل ۱۶: طراحی داخلی ماژول‌های FULL_ADDER

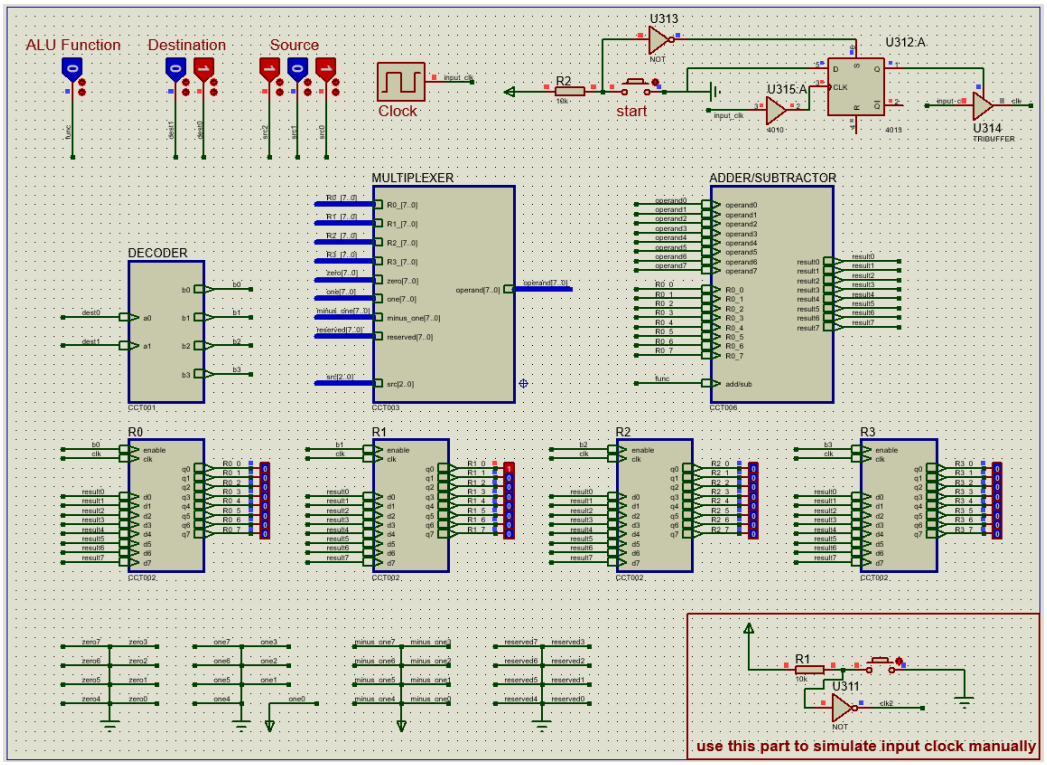
در ادامه به تست مدار روی دستورالعمل‌های مختلف می‌پردازیم.

۳ تست مدار

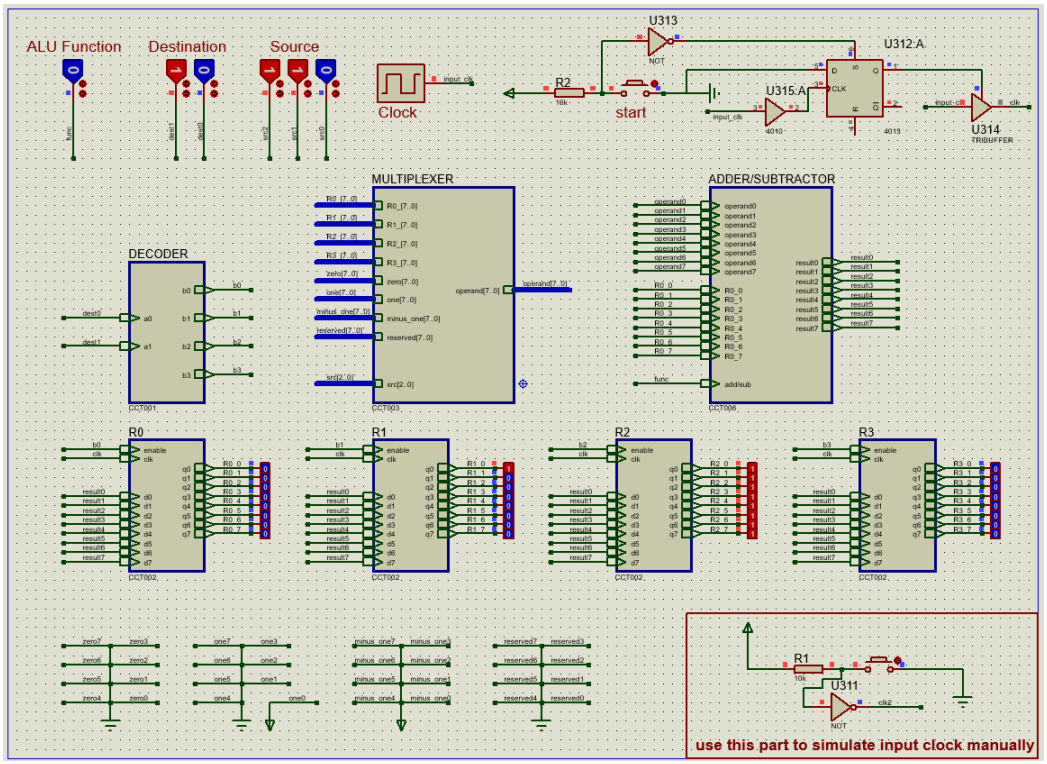
برای تست مدار، مجموعه دستورالعمل‌های زیر به صورت متوالی به مدار داده شده و شکل‌هایی که در ادامه می‌آیند وضعیت مدار را بعد از پردازش هر دستورالعمل نشان می‌دهند. هم‌چنین مقداری که هر ثبات بعد از اجرای هر دستورالعمل خواهد داشت در ادامه آمده است تا با مقداری که از مدار خروجی می‌گیریم مقایسه شود.

$$
\begin{aligned}
001101 \quad & (ADD,\ R1\ ,1) \quad \rightarrow \{R0=0,\ \ R1=1,\ \ R2=0,\ \ R3=0\} \\
010110 \quad & (ADD,\ R2\ ,-1) \quad \rightarrow \{R0=0,\ \ R1=1,\ \ R2=-1,\ \ R3=0\} \\
100001 \quad & (SUB,\ R0,\ R1) \quad \rightarrow \{R0=-1,\ R1=1,\ \ R2=-1,\ \ R3=0\} \\
011010 \quad & (ADD,\ R3,\ R2) \quad \rightarrow \{R0=-1,\ R1=1,\ \ R2=-1,\ \ R3=-2\} \\
001000 \quad & (ADD,\ R1,\ 0) \quad \rightarrow \{R0=-1,\ R1=0,\ \ R2=-1,\ \ R3=-2\} \\
100110 \quad & (SUB,\ R0,-1) \quad \rightarrow \{R0=0,\ \ R1=0,\ \ R2=-1,\ \ R3=-2\} \\
110011 \quad & (SUB,\ R2,\ R3) \quad \rightarrow \{R0=0,\ \ R1=0,\ \ R2=2,\ \ R3=-2\}
\end{aligned}
$$

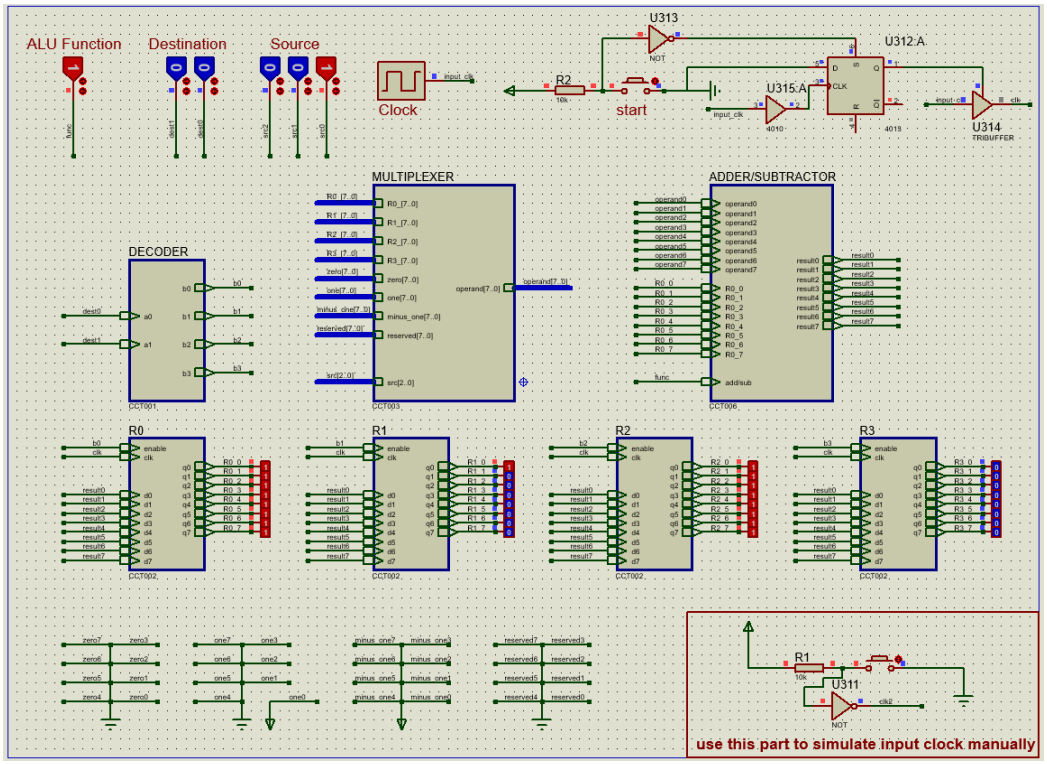
و خروجی‌های مدار به صورت زیر خواهد بود:



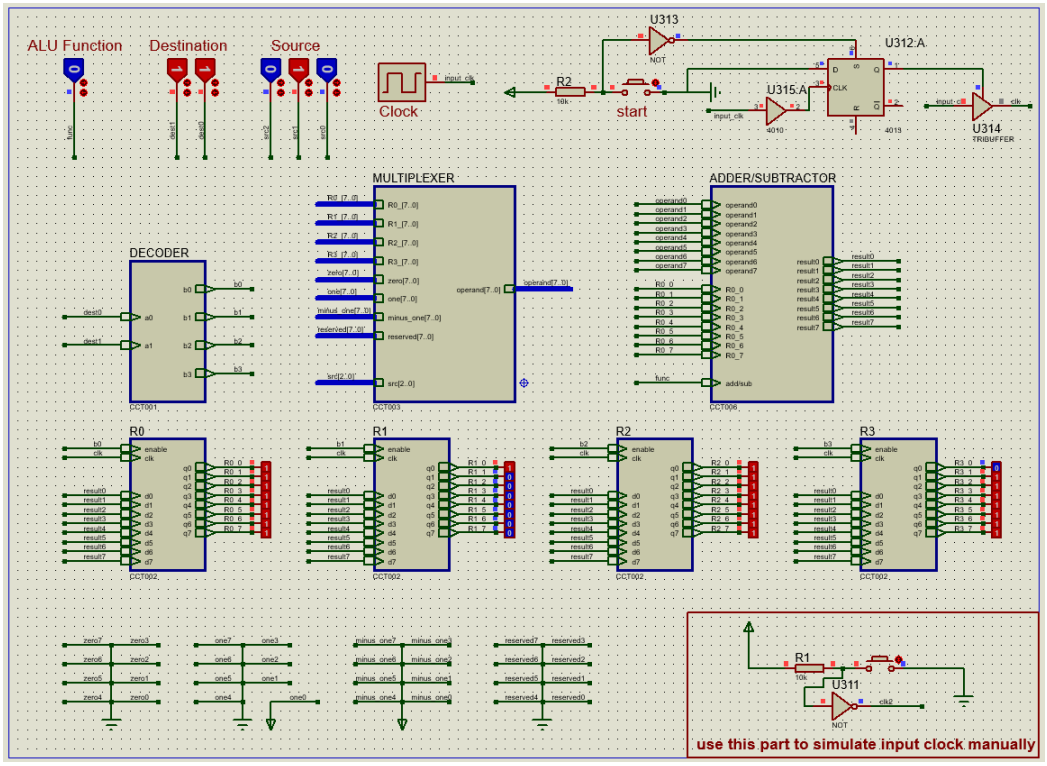
شکل ۱۷: وضعیت مدار پس از پردازش 001101



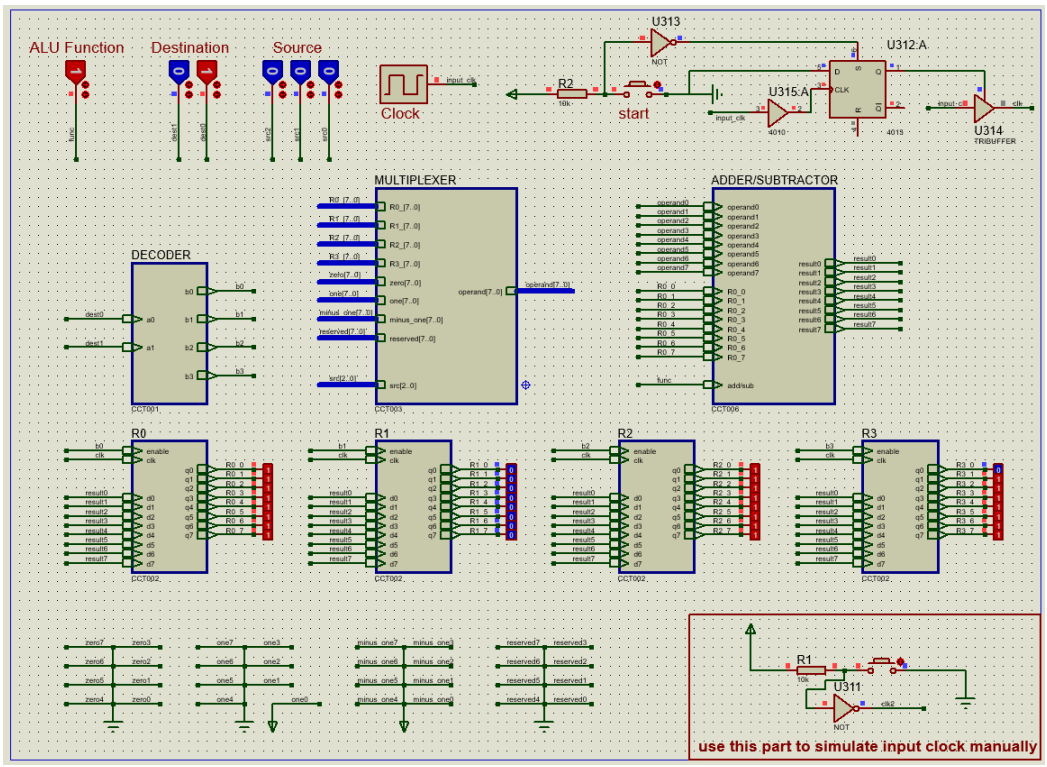
شکل ۱۸: وضعیت مدار پس از پردازش 010110



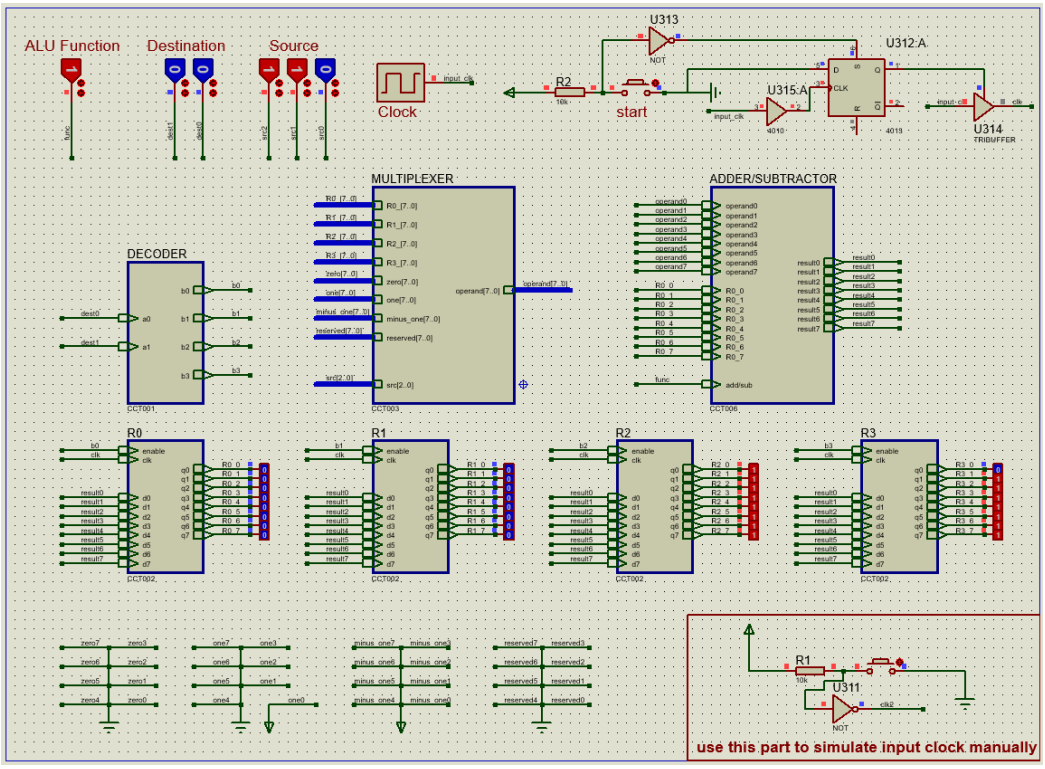
شکل ۱۹: وضعیت مدار پس از پردازش 100001



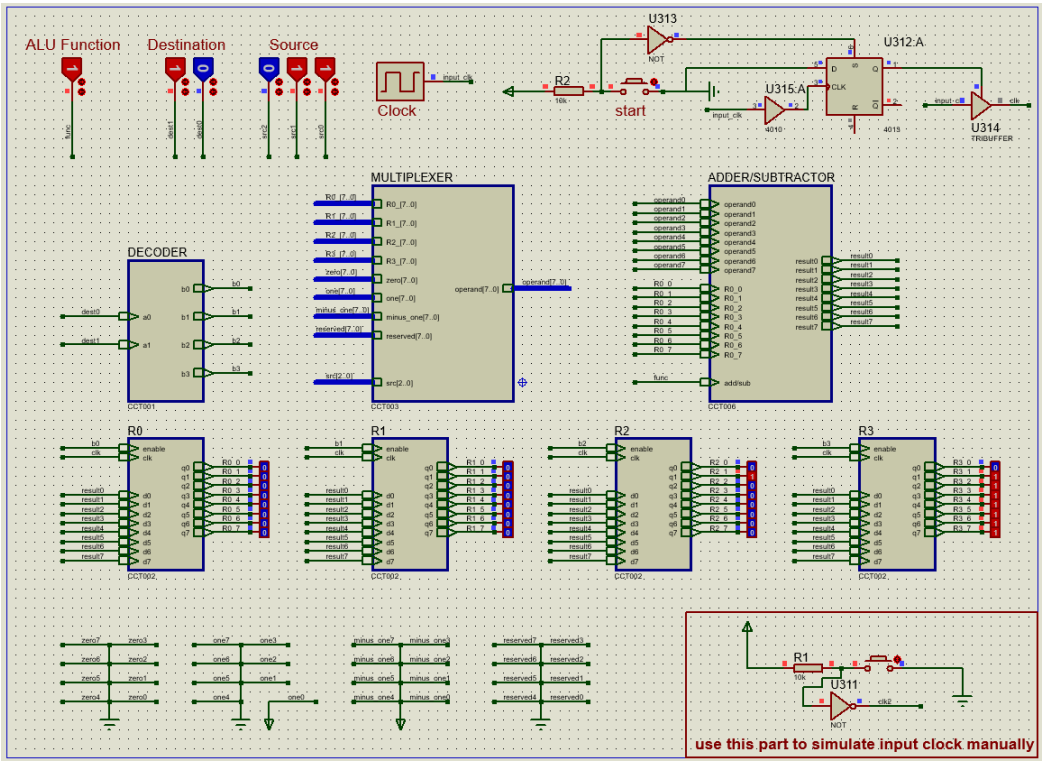
شکل ۲۰: وضعیت مدار پس از پردازش 011010



شکل ۲۱: وضعیت مدار پس از پردازش 001000



شکل ۲۲: وضعیت مدار پس از پردازش 100110



شکل ۲۳: وضعیت مدار پس از پردازش 110011