آزمایش پنجم معماری کامپیوتر

دانشکده مهندسی کامپیوتر، دانشگاه صنعتی شریف

سایه جارالهی، ۹۸۱۰۱۳۳۹

آرین احدی نیا، ۹۸۱۰۳۸۷۸

امیررضا سلیمان‌بیگی، ۹۸۱۰۹۸۳۷

استاد درس: جناب آقای دکتر سربازی آزاد

دستیار آموزشی: سرکار خانم غیبی

تابستان ۱۴۰۱

فهرست عناوین

مقدمه

در این آزمایش به پیاده‌سازی یک واحد محاسبه و منطق[۱] می‌پردازیم. واحد محاسبه و منطق نوعا یک مدار ترکیبی است اما در این آزمایش ثبات‌های واسط منتهی به این مدار ترکیبی را که برای دریافت ورودی و نوشتن خروجی استفاده می‌شوند و ثبات پرچم‌ها[۲] را قرار می‌دهیم.

ماژول‌های آماده مورد استفاده

۴۰۰۸: جمع‌کننده چهاربیتی با محاسبه بیت نقلی موازی

دیتاشیت این ماژول در این لینک در دسترس است.

۴۵۵۵: رمزگشا[۳] ۲ به ۴

دیتاشیت این ماژول در این لینک در دسترس است.

۷۴۱۵۱: تسهیم‌کننده[۴] ۸ به ۱ تک بیتی

دیتاشیت این ماژول در این لینک در دسترس است.

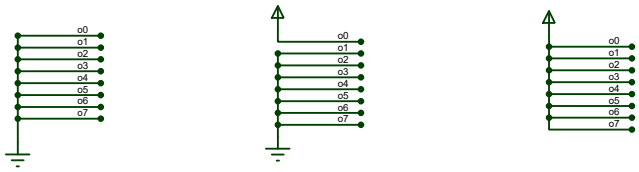
۷۴۱۹۸: شیفت‌رجیستر[۵] هشت بیتی

دیتاشیت این ماژول در این لینک در دسترس است.

نحوه انجام آزمایش

ماژول صفر، یک و منفی یک

این سه ماژول همواره مقادیر ثابت صفر، یک و منفی یک به صورت هشت بیتی هستند. توجه کنید که مقدار منفی یک به صورت مکمل دوم[۶] آن نمایش داده می‌شود.



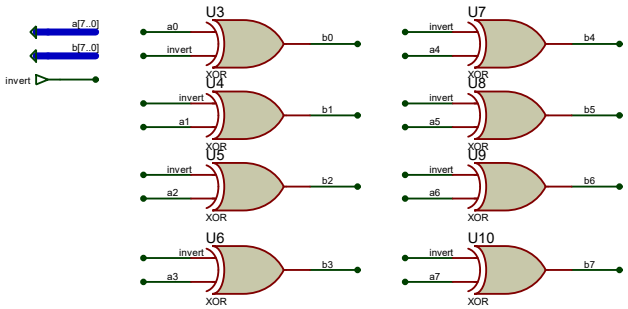
شکل ۱ ماژول‌های منفی یک، یک و صفر (به ترتیب از راست به چپ) که همواره مقدار ثابت هشت بیتی را برمیگردانند.

---

۱ Arithmetic and Logic Unit
۲ Flags Register
۳ Decoder
۴ Multiplexer
۵ Shift Register
۶ Two’s Complement

ماژول وارون‌کننده

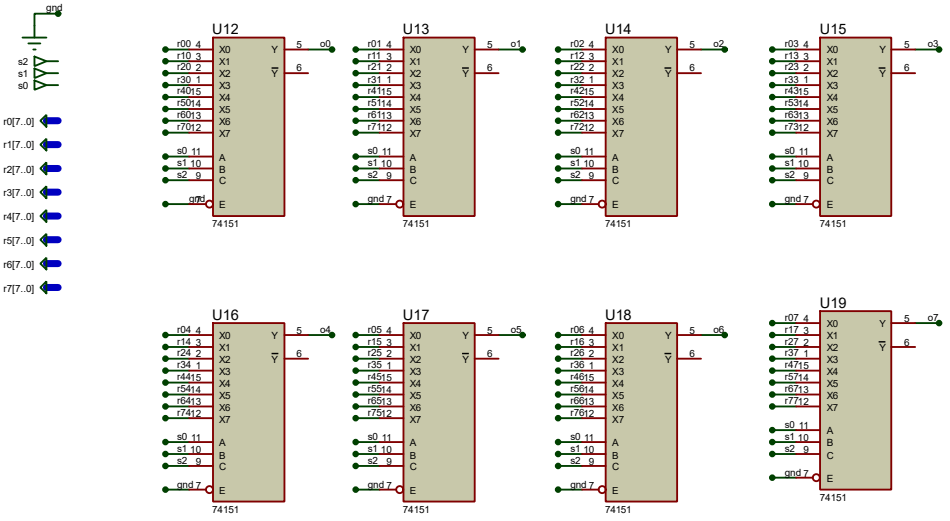
این ماژول یک ورودی هشت بیتی به انضمام یک بیت کنترلی `invert` دریافت می‌کند. در صورتی که ورودی کنترلی فعال باشد، وارون ورودی و در غیر این صورت خود ورودی برگردانده می‌شود. برای پیاده‌سازی این مدار از دروازه XOR استفاده شده است. به تعبیری XOR را می‌توان دروازه NOT با ورودی کنترلی فعال/غیرفعال در نظر گرفت که اگر یکی از ورودی‌های آن فعال باشد خروجی آن وارون ورودی دیگر و در غیر این صورت خود ورودی دیگر خواهد بود. بنابرین با استفاده از هشت گیت XOR می‌توان عملیات وارونه‌سازی را پیاده‌سازی کرد.



شکل ۲ ماژول وارون‌کننده، این ماژول با استفاده از هشت دروازه XOR در صورت فعال بودن ورودی `invert` مکمل اول ورودی را برمی‌گرداند.

تسهیم‌کننده هشت بیتی با عرض بیتی هشت

این ماژول یک تسهیم‌کننده هشت بیتی با عرض بیتی هشت است. برای پیاده‌سازی این ماژول از هشت عدد تسهیم‌کننده یک بیتی ۷۴۱۵۱ استفاده کرده‌ایم که تسهیم‌کننده iام آنها وظیفه تسهیم بیت iام را برعهده خواهد داشت. این تسهیم‌کننده به انضمام هشت خروجی هشت بیتی خود، سه ورودی انتخابی نیز دارد و در خروجی هشت بیت انتخاب شده را خروجی می‌دهد.



شکل ۳ تسهیم‌کننده هشت بیتی با عرض هشت، با استفاده از هشت تسهیم‌کننده یک بیتی تسهیم هشت بیت انجام می‌شود. r_0 الی r_7 ورودی‌های هشت‌بیتی مدار و o خروجی هشت بیتی مدار است. ورودی سه بیتی s نقش ورودی انتخابی را بر عهده دارد.

مدار محاسبه‌کننده

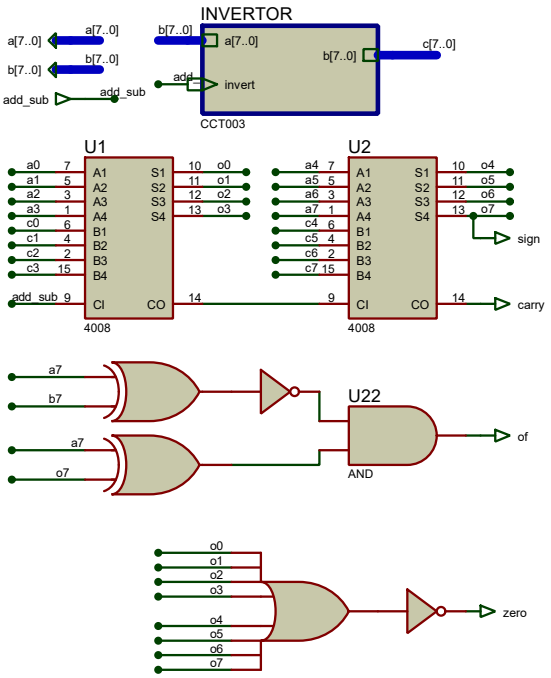
با استفاده از مدار وارون‌کننده که پیشتر طراحی کردیم، مدار محاسبه‌کننده را می‌سازیم. این مدار یک ورودی کنترلی دارد که نشان‌دهنده جمع یا تفریق بودن عملیات است. جمع و تفریق به وسیله دو جمع‌کننده ۴۰۰۸ که به صورت آبشاری به

هم متصل شده‌اند پیاده‌سازی شده است. در صورتی که عملیات تفریق باشد، با استفاده از ماژول Invertor مکمل اول آن محاسبه می‌شود و آن را به مدار جمع‌کننده می‌دهیم. در این حالت ورودی نقلی جمع‌کننده را برابر یک می‌دهیم و به این صورت عدد دوم را مکمل دوم می‌کنیم. در صورتی که عملیات جمع باشد، با ورودی دوم بدون تغییر و با ورودی نقلی صفر جمع را انجام می‌دهیم.

همچنین در این بخش پرچم‌های zero، sign، carry و overflow را محاسبه می‌کنیم. پرچم zero در صورتی فعال می‌شود که همه هشت بیت برابر با صفر باشند فعال می‌شود. این پرچم با استفاده از یک گیت NOR قابل محاسبه است. پرچم carry نیز به سادگی از خروجی جمع‌کننده بدست می‌آید. پرچم sign نیز برابر بیت پرارزش خروجی خواهد بود. برای محاسبه پرچم overflow نیز می‌دانیم که در صورتی که دو عدد هم علامت باشند، علامت نهایی باید برابر علامت اعداد باشد، در صورتی که دو عدد غیرهم‌علامت باشند، قطعا سرریز اتفاق نمی‌افتد. بنابرین با استفاده از رابطه زیر که با یک مدار ترکیبی قابل پیاده‌سازی است، می‌توانیم وقوع یا عدم وقوع سرریز را بررسی کنیم. فرض بفرمایید a و b ورودی‌های n بیتی مدار هستند و s برابر حاصل جمع این دو است.

$$overflow = and\big(xnor(a_{n-1}, b_{n-1}), xor(a_{n-1}, s_{n-1})\big)$$

بنابرین در نهایت مدار به صورت زیر پیاده‌سازی می‌شود.



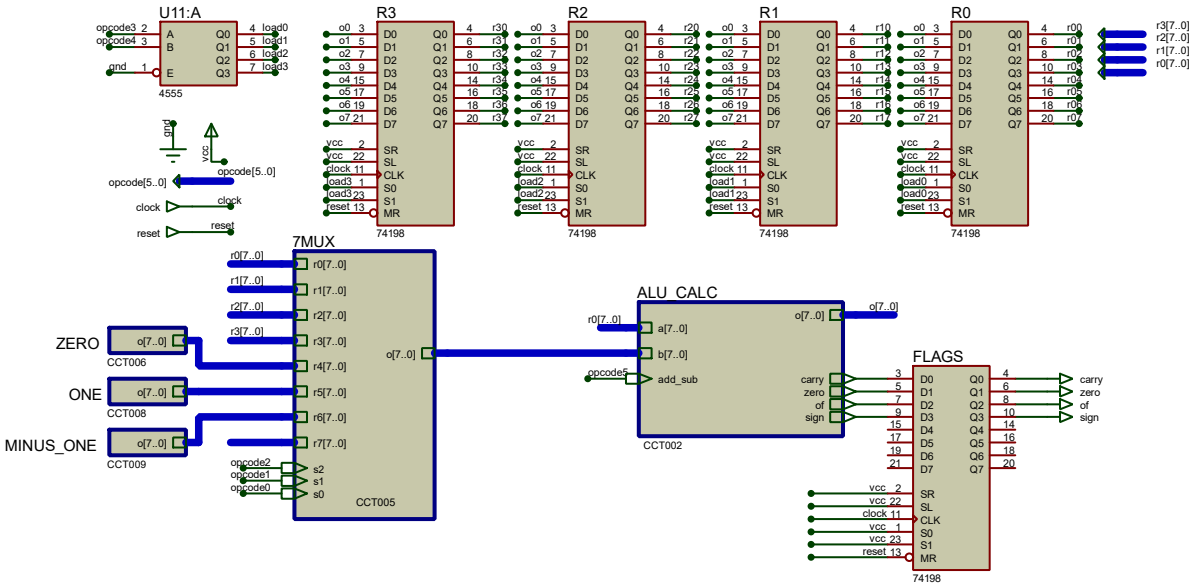
شکل ٤ مدار ترکیبی ALU با قابلیت محاسبه پرچم‌های zero، sign، overflow و carry

مدار ALU

در نهایت با سرهم بندی ماژول‌های فوق، ماژول نهایی ALU بدست می‌آید. با استفاده از تسهیم‌کننده هشت‌بیتی هشت به یک با توجه به سه بیت آخر کد دستور[۷] ورودی دوم محاسبه‌کننده را انتخاب می‌کنیم. همچنین بیت اول کد دستور را که مشخص‌کننده عملیات است را به محاسبه‌کننده می‌دهیم. بیت‌های دوم و سوم کد دستور را که مشخص‌کننده مقصد است را به یک رمزگشا می‌دهیم. ورودی Load ثبات مقصد با خروجی این رمزگشا فعال می‌شود. خروجی مدار محاسبه‌کننده نیز به

[۷] Opcode

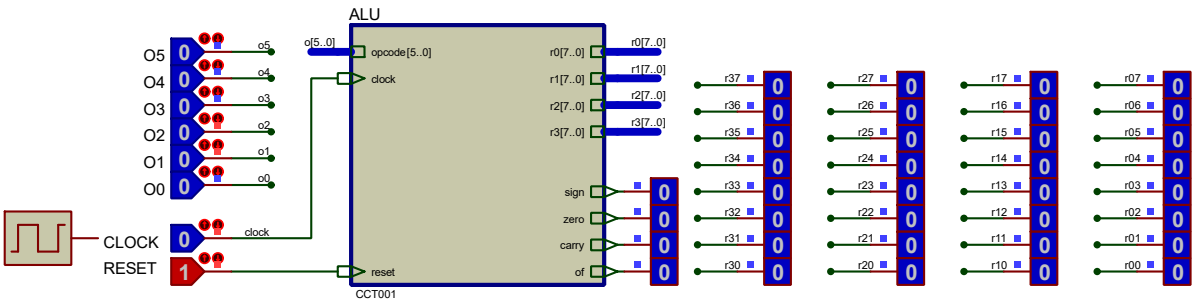
هر چهار ثبات متصل شده که در صورتی که ورودی Load آنها فعال شود، حاصل محاسبه در آنها ریخته شود. با استفاده از یک رجیستر پرچم‌ها نیز ثبت می‌شوند. توجه بفرمایید که از یک رجیستر ۸ بیتی استفاده شده است چرا که ممکن است در آینده پرچم‌های دیگری نیز اضافه شوند. همچنین ورودی reset نیز به صورت ناهمگام مدار را به حالت اولیه بازنشانی می‌کند.



شکل ۵ ALU به همراه ثبات‌های مورد نظر، این ALU با دریافت OPCODE محاسبه را انجام داده و نتایج را در ثبات‌های خود بازنویسی می‌کند. پرچم‌های محاسبات زیر در رجیستر FLAGS ثبت می‌شود.

ماژول نهایی

در نهایت مدار را برای استفاده در بخش‌های دیگر لفاف‌بندی[۸] می‌کنیم تا به شکل زیر در آید. توجه بفرمایید که به منظور ساده‌تر شدن فرآیند تست کلاک به صورت ورودی تعریف شده است.

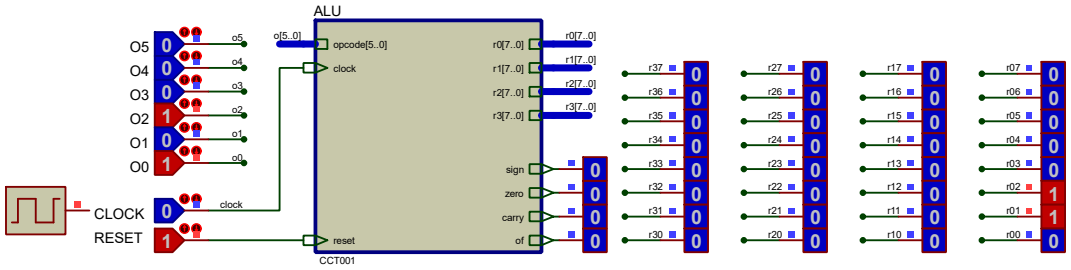


شکل ۶ مدار لفاف‌بندی شده به همراه کنترل ورودی و خروجی

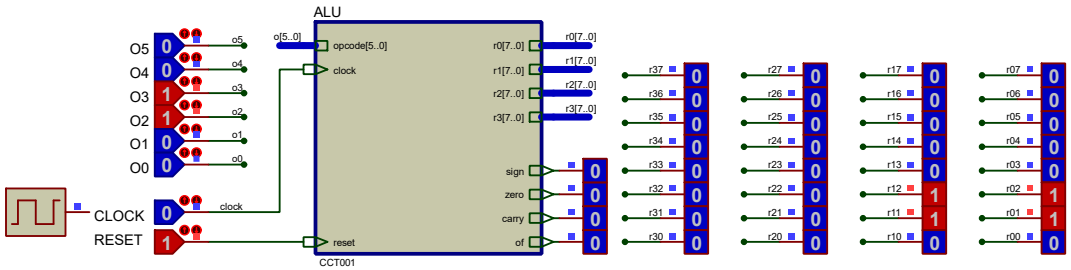
خروجی‌های مدار

حال با یک دنباله از عملیات‌های مختلف مدار را مورد آزمون قرار می‌دهیم.

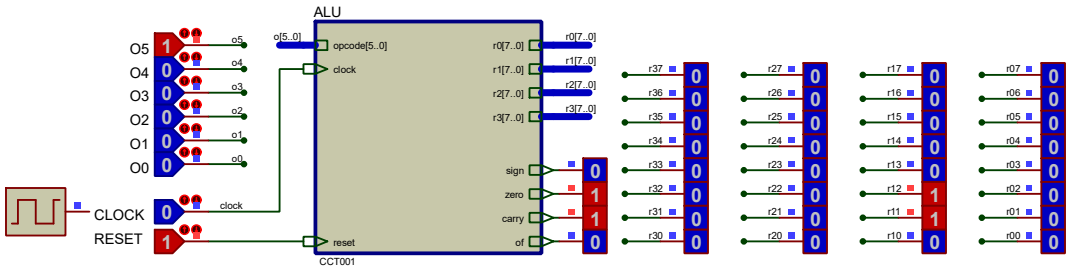
Wrapping ^



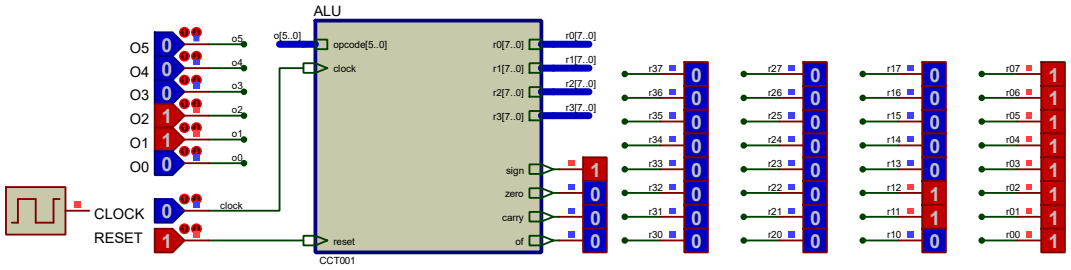
شکل ۷ عملیات: جمع، ثبات مقصد: ثبات صفرم، ثبات مبدا: مقدار ثابت یک. پس از شش کلاک مقدار شش در ثبات صفرم ذخیره می‌گردد.



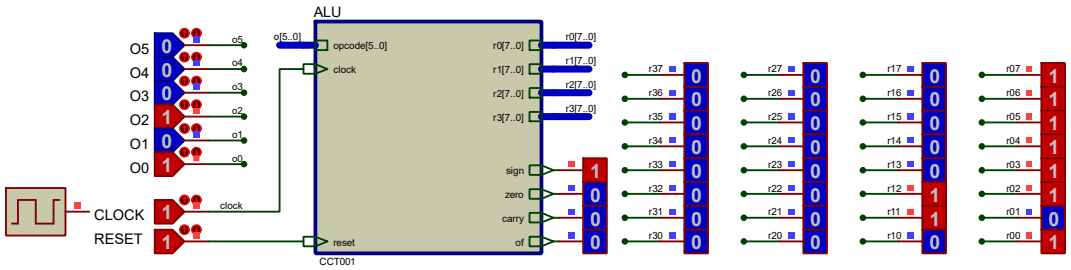
شکل ۸ عملیات: جمع، ثبات مقصد: ثبات یکم، ثبات مبدا: مقدار ثابت صفر. پس از یک کلاک مقدار ثبات صفرم در مقدار ثبات اول قرار می‌گیرد.



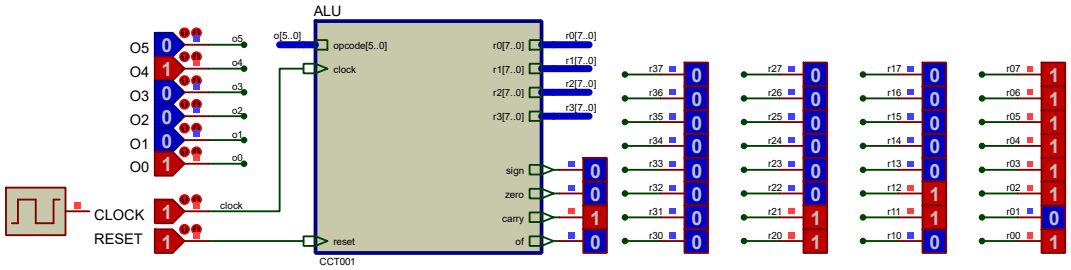
شکل ۹ عملیات: تفریق، ثبات مقصد، ثبات صفرم، ثبات مبدا: ثبات صفرم. حاصل تفریق ثبات صفرم از ثبات صفرم برابر صفر است. بنابرین پس از یک کلاک پرچم zero فعال می‌گردد. همچنین از آنجایی که این تفریق به صورت مکمل دوم انجام می‌شود، پرچم carry نیز فعال می‌شود. مقدار صفر در ثبات صفرم قرار می‌گیرد.



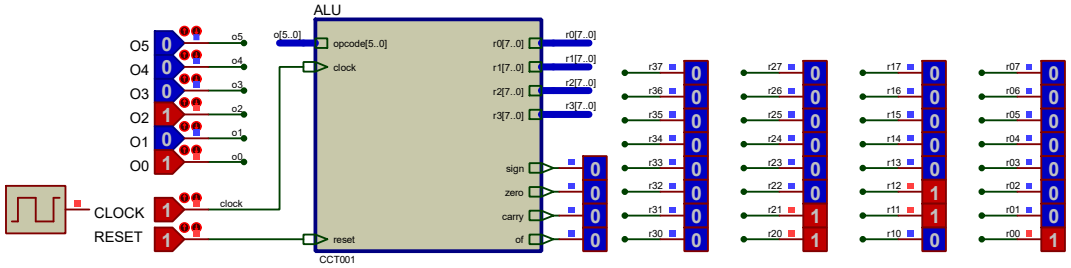
شکل ۱۰ عملیات: جمع، ثبات مقصد: ثبات صفرم، ثبات مبدا: مقدار ثابت منفی یک. پس از یک کلاک مقدار منفی یک در ثبات صفرم قرار می‌گیرد. پرچم علامت نیز به دلیل منفی بودن حاصل محاسبات فعال می‌گردد.



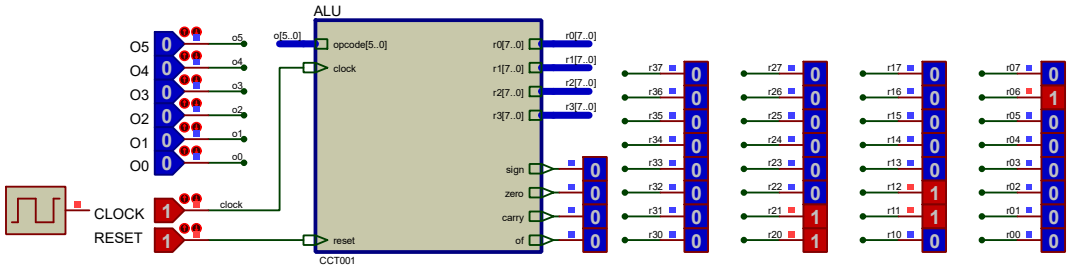
شکل ۱۱ عملیات: جمع، ثبات مقصد: ثبات صفرم، ثبات مبدا: مقدار ثابت منفی یک. پس از دو کلاک دیگر مقدار 3− در ثبات صفرم قرار می‌گیرد.



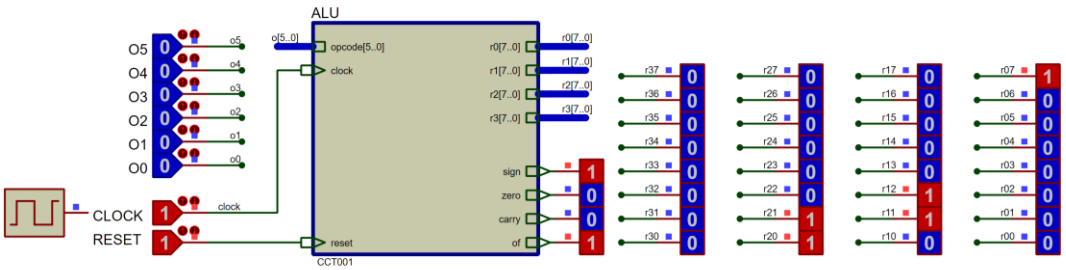
شکل ۱۲ عملیات: جمع، ثبات مقصد: ثبات دوم، ثبات مبدا: ثبات اول. حاصل جمع شش و منفی سه برابر سه می‌شود که در ثبات دوم قرار گرفته است. همچنین پرچم carry نیز مطابق انتظار فعال شده است.



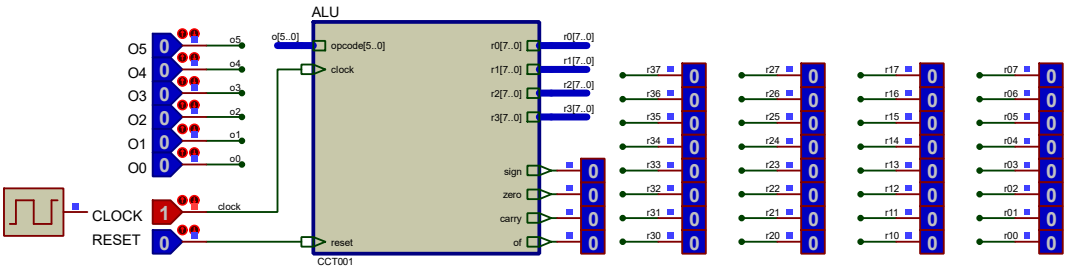
شکل ۱۳ عملیات: جمع، ثبات مقصد: ثبات صفرم، ثبات مبدا: مقدار ثابت یک. پس از چهار کلاک و واحد به واحد اضافه شدن ثبات صفرم، مقدار یک در آن قرار می‌گیرد.



شکل ۱۴ عملیات: جمع، ثبات مقصد: ثبات صفرم، ثبات مبدا: ثبات صفرم. به این صورت در هر مرحله مقدار ذخیره شده در ثبات صفرم دو برابر می‌شود. پس از شش کلاک مقدار ۶۴ در ثبات صفرم قرار می‌گیرد.



شکل ۱۵ با یک بار دیگر تکرار عملیات مرحله قبل سرریز رخ می‌دهد و پرچم `overflow` فعال می‌شود. توجه بفرمایید که حداکثر مقدار در نمایش مکمل دوم در ۸ بیت برابر ۱۲۷ است و مقدار ۱۲۸ باعث سرریز می‌شود.



شکل ۱۶ با غیر فعال شدن `reset` مدار به حالت اولیه بازنشانی می‌شود. توجه بفرمایید که ورودی `reset`، فعال پایین یا همان `active low` است.