

دانشگاه صنعتی امیرکبیر
دانشکده مهندسی کامپیوتر

پروژه نوروزی درس معماری کامپیوتر

نام دانشجو : ملیکا عبداللہی - حمیدرضا رمضانی

شماره دانشجویی : 9323030-9431804

در این پروژه یک پردازنده متشکل از دو بخش کلی کنترل‌کننده و جریان داده پیاده‌سازی شده است. بخش جریان داده شامل زیر بخش‌های واحد آدرس (AddressUnit.vhd)، واحد محاسبات (ArithmeticUnit.vhd)، بافرهای سه حالت (BUF16.vhd، BUF8.vhd)، پرچم (Flags.vhd)، ثبات دستور (IR.vhd)، حافظه (memory.vhd)، مالتی پلکسر (Mux.vhd)، ثبات‌های همه منظوره (RegisterFile.vhd) و ثبات نشانگر (WP.vhd) است که در ادامه به تشریح بیشتر هر کدام پرداخته شده است.

1. واحد آدرس

واحد آدرس شامل دو زیر بخش شمارنده برنامه (ProgramCounter.vhd) و AddressLogic.vhd است و مطابق کد موجود در صورت پروژه پیاده شده است.

2. واحد محاسبات

این واحد وظیفه انجام عملیات محاسباتی را بر عهده دارد. از طریق دو ورودی 16 بیتی به نام A و B داده‌هایی که باید عملیات بر روی آنها انجام شود به این واحد وارد می‌شوند. 10 ورودی کنترلی B15to0، ShlB، ... که توسط واحد کنترل‌کننده مقداردهی می‌شوند، سبب انجام عملیات گوناگون نظیر انتقال، جمع، تفریق، ... و شیفت به چپ بر روی داده‌های ورودی می‌شوند. همچنین دو بیت Cin و Zin نیز به عنوان ورودی به این بلوک اعمال شده و دو بیت خروجی دیگر Cout و Zout که به ترتیب نشان دهنده وجود رقم نقلی و برابری دو عدد ورودی است در خروجی تولید می‌شود.

3. بافرهای سه حالت

این بافرها به گونه‌ای طراحی شده‌اند که در صورت یک بودن ورودی Enable و ورودی دیگر خود را به خروجی منتقل می‌کنند و در غیر این صورت خروجی آنها صفر خواهد بود. در BUF16 ورودی و خروجی 16 بیتی و در BUF8، 8 بیتی است. از این بافرها برای ارسال و دریافت داده‌های بلوک‌های گوناگون بر روی باس مشترک نظیر Databus استفاده می‌شود.

4. پرچم

یک ثبات دو بیتی است که Cout و Zout تولید شده توسط واحد محاسبات را در خود ذخیره می‌کند و همچنین برای محاسبات بعدی از آنها به عنوان ورودی واحد محاسبات استفاده می‌شود. این واحد 5 ورودی کنترلی دارد که این ورودی‌ها امکان صفر و یک کردن C و Z و بارگذاری ورودی‌ها در پرچم را به صورت مستقل فراهم می‌آورند.

5. ثبات دستور

این ثبات در صورت فعال بودن ورودی load محتویات ورودی را به خروجی منتقل می‌کند و در غیر این صورت مقدار قبلی را در خود نگه می‌دارد. از این ثبات برای ذخیره دستور واکشی شده از حافظه برای دیکد کردن استفاده می‌شود.

6. حافظه

دستورات مورد نظر برای اجرا در این بلوک ذخیره می‌شود که از کد موجود در صورت پروژه برای این قسمت استفاده شده است.

7. مالتی پلکسر

این واحد با توجه به ورودی select خود یکی از دو ورودی دیگر را به خروجی منتقل می‌کند.

8. ثبات همه منظوره

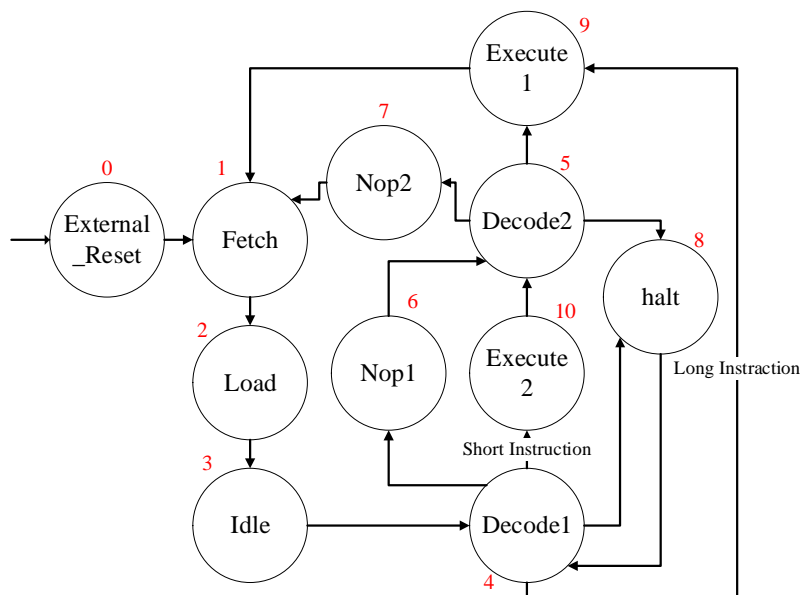
شامل 64 ثبات 16 بیتی است. یک ورودی 6 بیتی به نام WP و یک ورودی 4 بیتی به نام Shadowout دارد. دو بیت کم ارزش Shadowout با ورودی WP جمع شده و آدرس مبدا را مشخص می‌کنند. همچنین حاصل جمع ورودی WP با دو بیت پرارزش ورودی Shadowout آدرس مقصد را تولید می‌کند. محتویات رجیستر مقصد به خروجی Left و محتویات رجیستر مبدا به خروجی Right منتقل می‌شود. در صورت یک بودن ورودی Load، ورودی Data به رجیستر مقصد انتقال می‌یابد. ضمناً در صورت فعال بودن ورودی RFLwrite محتویات 8 بیت کم ارزش موجود در ورودی data به 8 بیت کم ارزش رجیستر مقصد منتقل شده و 8 بیت پر ارزش ثبات مقصد مقدار 'Z' خواهد داشت. در صورت فعال بودن RFHwrite نیز نظیر همین عمل منتهی برای بیت‌های پر ارزش ثبات مقصد اتفاق می‌افتد.

9. ثبات نشانگر

یک ثبات 6 بیتی است که قابلیت صفر شدن با استفاده از سیگنال WPreset در آن فراهم شده است. همچنین در صورت یک بودن سیگنال WPadd محتویات آن با ورودی جمع شده و به خروجی انتقال می‌یابد.

واحد کنترل کننده

این بخش که وظیفه مقداردهی سیگنال‌های کنترلی موجود در مدار را بر عهده دارد، توسط یک FSM شامل 11 وضعیت گوناگون پیاده شده است که در شکل زیر قابل مشاهده است. اعداد قرمز رنگ شماره‌هایی است که به هر حالت انتساب داده شده است.



هنگامی که سیگنال External_Reset فعال شود، پردازنده وارد حالت External_Reset شده و سیگنال WPrese، ZReset و CReset یک می‌شود تا پرچم‌ها و ثبات نشانگر به صفر مقداردهی شوند. همچنین سیگنال ResetPC و EnablePC فعال می‌شود تا ضمن صفر شدن خروجی واحد آدرس، مقدار صفر در ثبات شمارنده برنامه نیز بارگذاری شود.

WP <= "000000";

Cout <= '0';

Cin <= '0';

Address <= "0000000000000000";

PC <= "0000000000000000";

سپس کنترل اجرا به حالت واکنشی دستور (Fetch) منتقل می‌شود. در این وضعیت کلیه سیگنال‌هایی که در حالت قبلی فعال شده بودند، غیر فعال شده و سیگنال ReadMem فعال می‌شود. با فعال شدن این سیگنال محتویات خانه‌ای از حافظه که آدرس آن توسط سیگنال Address که خروجی واحد آدرس است؛ روی باس داده قرار می‌گیرد. کنترل کننده نیز به حالت بارگذاری منتقل می‌شود.

Databus <= Memory(Address);

در حالت بارگذاری (Load) سیگنال IRload فعال می‌شود. بدین طریق مقدار Databus که در ورودی ثبات دستور وجود دارد، در این ثبات دستور بارگذاری می‌شود. همچنین سیگنال ReadMem نیز غیر فعال می‌شود.

IRout <= Databus;

با انتقال از حالت بارگذاری به حالت بیکار (idle) سیگنال IRload غیرفعال می‌شود. از این حالت استفاده شده است تا زمان کافی برای تغییر محتوای ثبات دستور برای پردازنده به وجود بیاید.

تا به حال محتوای یک خانه از حافظه واکنشی شده و در ثبات دستور ذخیره شده است. حال با توجه به مقدار موجود در ثبات دستور باید عملیات گوناگونی انجام شود. بدین منظور کنترل کننده به حالت Decode1 منتقل می‌شود. در این معماری دو ساختار دستوری کوتاه و بلند وجود دارد. به همین دلیل دو مسیر متفاوت برای کنترل کننده در این حالات وجود دارد. چنانچه چهار بیت پر ارزش ثبات دستور [IRout[15:12] برابر "1111" باشد یا [IRout[15:10] برابر "000010" باشد و یا [IRout[15:8] برابر "00000111" باشد، ساختار دستور از نوع بلند بوده و کنترل برنامه به حالت Execute1 منتقل می‌شود. در بقیه حالات دستور از نوع کوتاه است. دستورات jpa و jpr همواره محتویات شمارنده برنامه را تغییر می‌دهد. همچنین دستور brz در صورت یک بودن Zout و دستور brc در صورتی که Cout یک باشد، محتویات شمارنده برنامه را تغییر می‌دهد. در سایر حالات شمارنده برنامه باید یک عدد افزایش یابد تا در واکنشی بعدی پردازنده به خانه بعدی حافظه دسترسی پیدا کند. به همین دلیل در پیاده سازی از سیگنالی به نام incr استفاده شده است که در صورت یک بودن آن در حالات بعدی به شمارنده برنامه یک عدد اضافه می‌شود. در صورتی که دستور واکنشی شده jpa یا jpr نباشد و یا شرط brc و brz صحیح نباشد، این سیگنال در حالت Decode1 فعال می‌شود.

اگر دستور از نوع کوتاه باشد، آنگاه ثبات دستور شامل دو دستور است که به صورت متوالی باید اجرا شوند. ابتدا 8 بیت پر ارزش ثبات دستور به عنوان دستور اول مورد بررسی قرار می‌گیرد.

اگر IRout[15:8] برابر "00000000" باشد، دستور no operation است. در این حالت تنها سیگنال incr فعال می‌شود و کنترل به حالت Nop1 منتقل می‌شود. در این حالت با فعال شدن سیگنال PCplus1 و EnablePC مقدار شمارنده برنامه یک واحد افزایش می‌یابد و سیگنال incr غیرفعال می‌شود. همچنین کنترل برنامه به حالت Decode2 منتقل می‌شود تا دیکد و اجرای دستور دوم آغاز شود.

اگر IRout[15:8] برابر "00000001" باشد، دستور halt بوده و کنترل برنامه به وضعیت halt منتقل می‌شود. در این حالت باید واکنشی دستور متوقف شود، به همین دلیل مقدار شمارنده برنامه تغییر نکرده و کنترل برنامه از وضعیت halt مجدداً به وضعیت Decode1 منتقل می‌شود. بدین ترتیب تا زمانی که سیگنال External_Reset غیر فعال است، مقدار ثبات دستور تغییر نکرده و همواره آخرین دستور واکنشی شده که شامل halt است، مجدداً اجرا خواهد شد.

به ازای بقیه مقادیر ممکن برای IRout[15:8] سیگنال incr فعال می‌گردد. همچنین برحسب نوع دستور دیکد شده سیگنال‌های گوناگونی فعال می‌شود و کنترل برنامه به وضعیت Execute2 منتقل می‌شود.

حالت Execute2 زمان کافی برای اجرای دستور دیکد شده را فراهم می‌آورد. همچنین در این حالت کلیه سیگنال‌هایی که ممکن است در حالت Decode1 فعال شده باشند، غیر فعال می‌شوند. چنانچه در این حالت سیگنال incr فعال باشد، غیر فعال شده و به شمارنده برنامه یک واحد اضافه خواهد شد. انتقال کنترل کننده به این حالت، بدان معناست که دستور واکنشی شده از نوع دستور کوتاه بوده و همچنین halt و no operation نیز نبوده است. بنابراین چون تنها دستور اول از ثبات دستور اجرا شده است، باید کنترل کننده به وضعیتی منتقل شود که امکان اجرای دستور دوم را که در IRout[7:0] واقع است، فراهم آورد. به همین دلیل کنترل کننده به حالت Decode2 منتقل می‌شود.

در این حالت سیگنال PCplus1 و EnablePC که در مرحله قبلی فعال شده‌اند، غیر فعال می‌شوند. اگر IRout[7:0] برابر "00000000" باشد، دستور از نوع no operation بوده و کنترل کننده به وضعیت Nop2 منتقل می‌شود. در وضعیت Nop2 هیچ سیگنال فعال نشده و در نتیجه هیچ عملیاتی انجام نمی‌شود و تنها کنترل به حالت واکنشی دستور منتقل می‌شود. در صورتی که IRout[7:0] برابر "00000001" باشد، دستور halt بوده و کنترل کننده به حالت halt منتقل می‌شود. چنانچه دستور از نوع halt و no operation نباشد و با توجه به اینکه انتقال به وضعیت Decode2 مستلزم آن است که دستور از نوع کوتاه باشد. بنابراین تنها با توجه به چهار بیت پر ارزش دستور یعنی IRout[7:4] می‌توان نوع دستور را مشخص و با توجه به نوع دستور سیگنال‌های مورد نظر را فعال کرد. در این حالت انتقال برنامه به وضعیت Execute1 منتقل می‌شود.

به دو طریق مختلف می‌توان به وضعیت Execute1 انتقال یافت.

1. دستور از نوع بلند باشد که در این حالت سیگنال incr فعال بوده و باید شمارنده برنامه افزایش یافته و این سیگنال غیر فعال شود.
2. دستور از نوع کوتاه باشد و دومین دستور از نوع کوتاه اجرا شود.

در این حالت کلیه سیگنال‌های فعال شده در مراحل قبل غیر فعال شده و کنترل مجدداً به وضعیت واکنشی دستور منتقل می‌شود و دستور بعدی واکنشی می‌گردد. لازم به ذکر است که در کلیه حالات چنانچه سیگنال External_reset فعال گردد، کنترل کننده به حالت External_Reset منتقل می‌شود.

در فایل SAYEH.vhd ماژول اصلی پردازنده پیاده شده است که کلیه زیرماژول‌های طراحی شده را فراخوانی و به هم متصل می‌کند. برای تست این پردازنده در فایل memory.vhd memory خانه‌های 0، 1، 2، 14 و 15 به صورت زیر مقداردهی شده است.

Mem[0] <= "0100-00-00-0100-01-00";

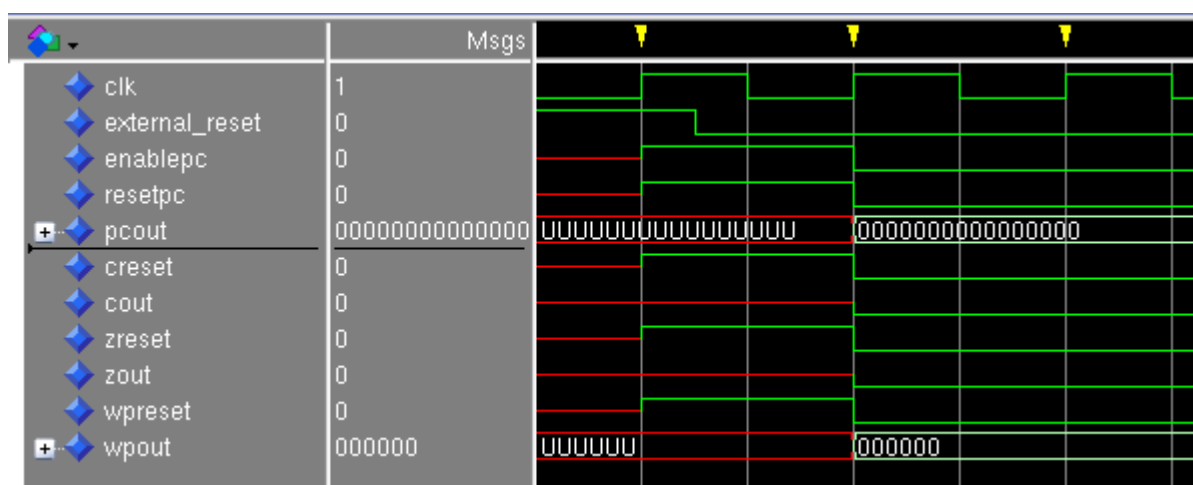
Mem[1] <= "1011-01-00-0001-11-00";

Mem[2] <= "0000-10-01-00001100";

Mem[14] <= "0000-10-10-00010111";

Mem[15] <= "0000-00-01-0000-00-00";

فایل SAYEH_TB.vhd تست پردازنده را انجام می‌دهد. در ابتدا سیگنال External_Reset فعال می‌شود که سبب صفر شدن شمارنده برنامه، پرچم‌ها و ثبات نشانگر می‌گردد. متغیر index نشان دهنده وضعیت کنترل کننده است که چون سیگنال External_Reset فعال است، مقدار آن صفر شده است.



شکل بالا نحوه انجام این عمل را در شبیه سازی‌های انجام شده نشان می‌دهد. همانطور که انتظار می‌رفت، pcout و wpout همگی صفر شده‌اند.

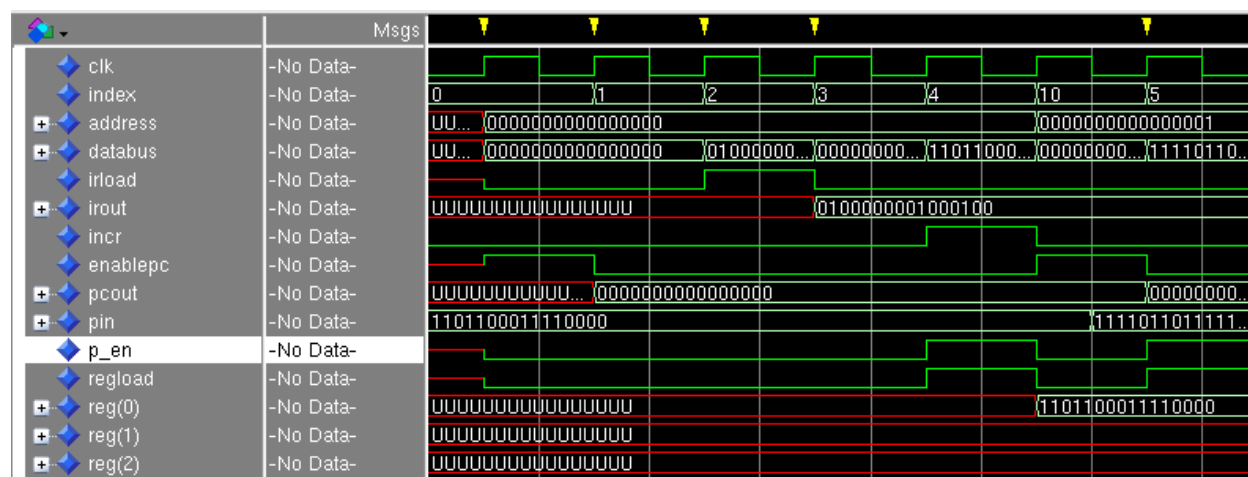
در مرحله بعدی کنترل کننده وارد وضعیت واکنشی (index = 1) می‌شود. کلیه سیگنال‌های فعال شده در مرحله قبلی اعم از سیگنال ResetPC و EnablePC غیر فعال می‌شوند. چون هیچ کدام از 5 سیگنال کنترلی مربوط به بخش AddressLogic فعال نیستند، بنابراین بلوک AddressLogic خروجی PC را که در مرحله قبل صفر شده است، به خروجی Address منتقل می‌کند. با فعال شدن سیگنال ReadMem محتوای خانه صفر حافظه روی Databus قرار می‌گیرد.

با انتقال به وضعیت بارگذاری (index = 2) سیگنال‌های فعال شده در وضعیت قبلی غیر فعال شده و سیگنال IRload فعال می‌شود تا مقدار موجود بر روی Databus در ثبات دستور بارگذاری شود.

در وضعیت بیکار (index = 3) سیگنال IRload غیرفعال شده و خروجی ثبات دستور برابر "0100-00-00-0100-01-00" یا همان محتوای خانه صفر حافظه می‌گردد. این خانه محتوی دو دستور کوتاه inp است که دو عدد 16 بیتی را از پورت ورودی دریافت کرده و در ثبات‌های R0 و R1 ذخیره می‌کند.

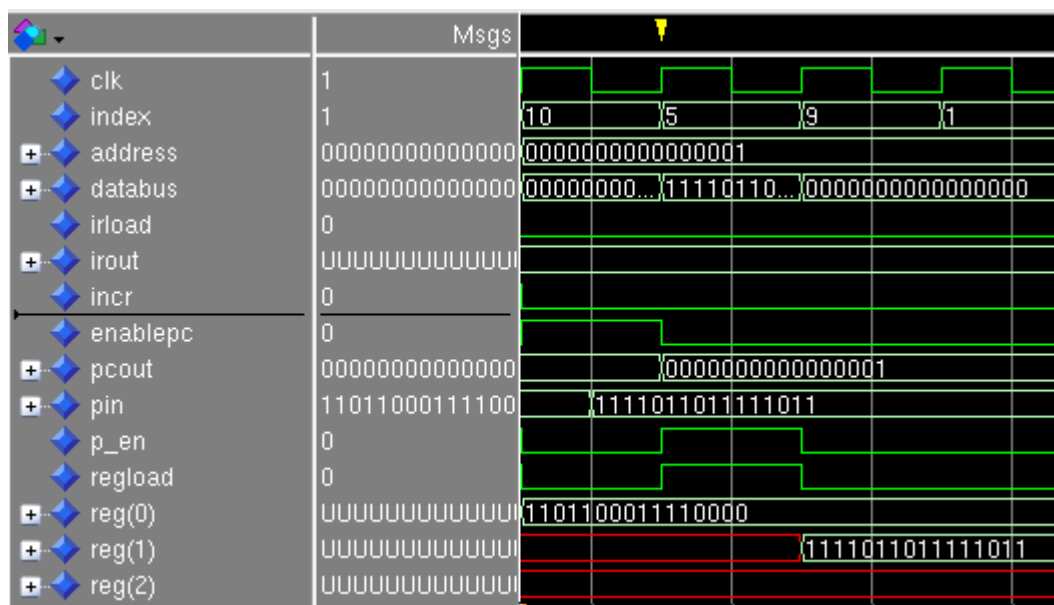
در وضعیت Decode1 (index = 4) ابتدا هشت بیت پرارزش ثبات دستور مورد بررسی قرار می‌گیرد. با توجه به اینکه بیت‌های IRout[15:12] برابر "0100" است، بنابراین دستور inp بوده و سیگنال p_en یک می‌شود تا داده موجود در پورت ورودی روی Databus قرار گیرد. همچنین چون دو بیت IRout[11:10] برابر "00" است، ثبات R0 به عنوان مقصد تعیین می‌گردد. با جمع کردن محتوای ثبات نشانگر (WP = "000000") با شماره ثبات مقصد ("00") آدرس ثبات R0 مشخص می‌گردد. باید سیگنال regload یک شود تا داده موجود بر روی Databus در ثبات مقصد نوشته شود. چون دستور از نوعی است که شمارنده برنامه را تغییر نمی‌دهد، مقدار سیگنال incr نیز یک می‌شود.

با انتقال به وضعیت Execute2 (index = 10) سیگنال‌های فعال شده در مرحله قبلی غیر فعال می‌گردد و مقدار موجود در پورت ورودی یعنی "1101100011110000" در ثبات R0 قرار می‌گیرد. همچنین چون سیگنال incr فعال است، سیگنال EnablePC و PCplus1 فعال می‌شود. کنترل اجرا به وضعیت decode2 منتقل می‌شود تا دستور بعدی انجام شود.

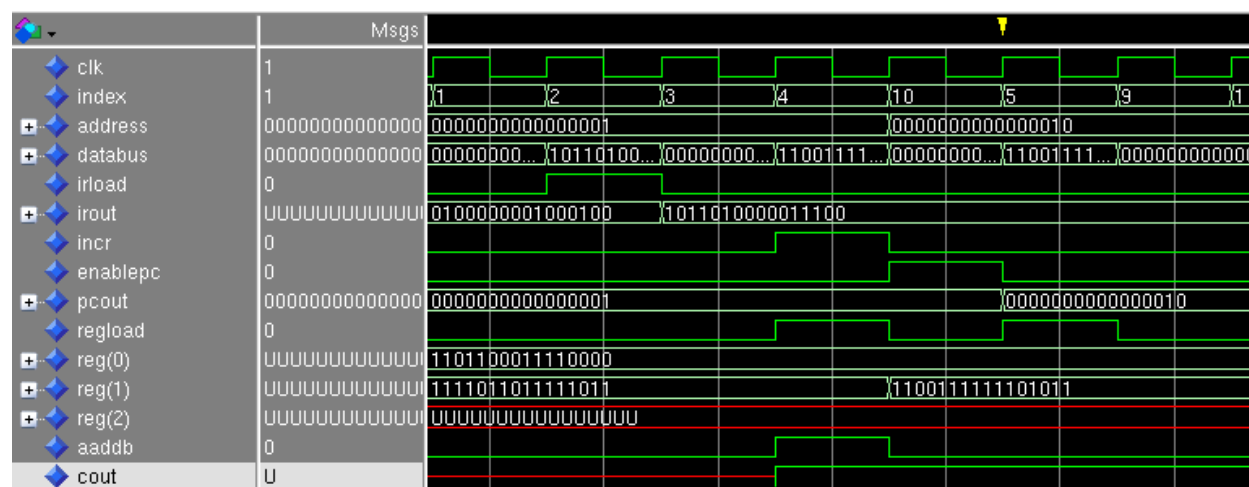


شکل بالا کلیه این مراحل و مقدار متغیرهای مختلف را در هر کلاک نشان می‌دهد. مقدار سیگنال index وضعیت جاری کنترل-کننده را در هر کلاک نشان می‌دهد. همانطور که مشخص است مقدار ثبات R0 و پورت ورودی یکسان است.

در وضعیت Decode2 (index = 5) مقدار شمارنده برنامه به یک افزایش یافته و بیت‌های IRout[7:0] مورد بررسی قرار می‌گیرد. این دستور نیز inp است و روال کار مطابق قبل است و پس از انتقال کنترل اجرا به وضعیت Execute1 (index = 10) مقدار پورت ورودی یعنی "111101101111011" در ثبات R1 ذخیره می‌گردد. سپس مجدداً کنترل اجرا به وضعیت واکنشی دستور (index = 1) منتقل می‌شود. شکل زیر این عمل را نشان می‌دهد.

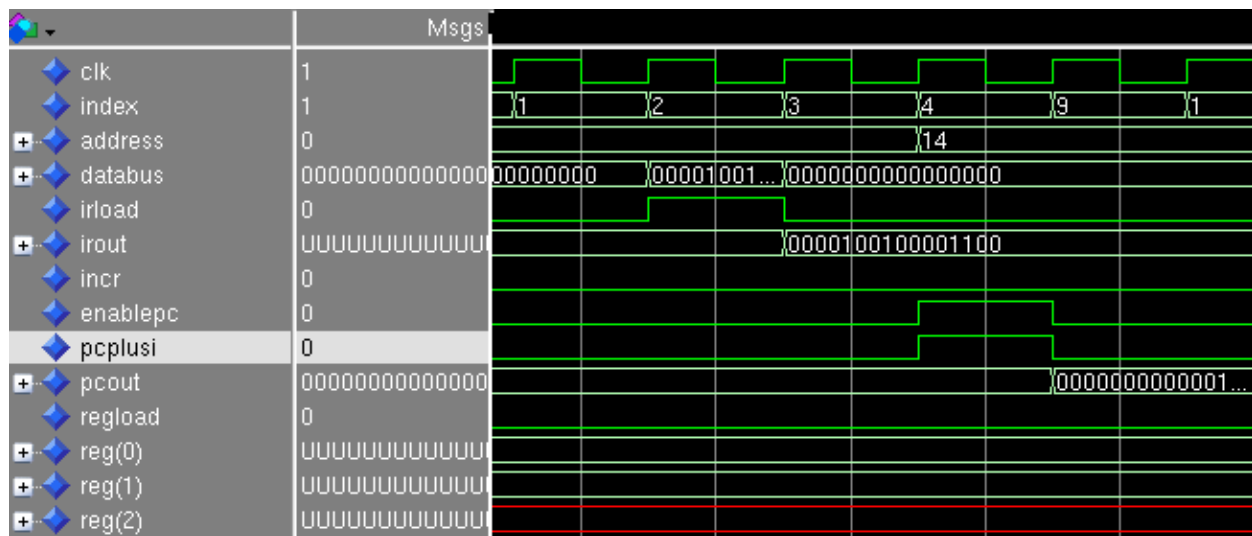


اینبار چون PC برابر یک است، محتویات خانه یک حافظه در ثبات دستور قرار می‌گیرد. در این خانه دو دستور کوتاه وجود دارد. دستور اول محتویات دو رجیستر R0 و R1 را با هم جمع کرده و در ثبات R1 ذخیره می‌کند. همچنین مقدار پرچم Cout نیز متناسب با عملیات جمع انجام شده تغییر می‌کند. در شکل زیر مقدار Cout به یک تغییر کرده است.

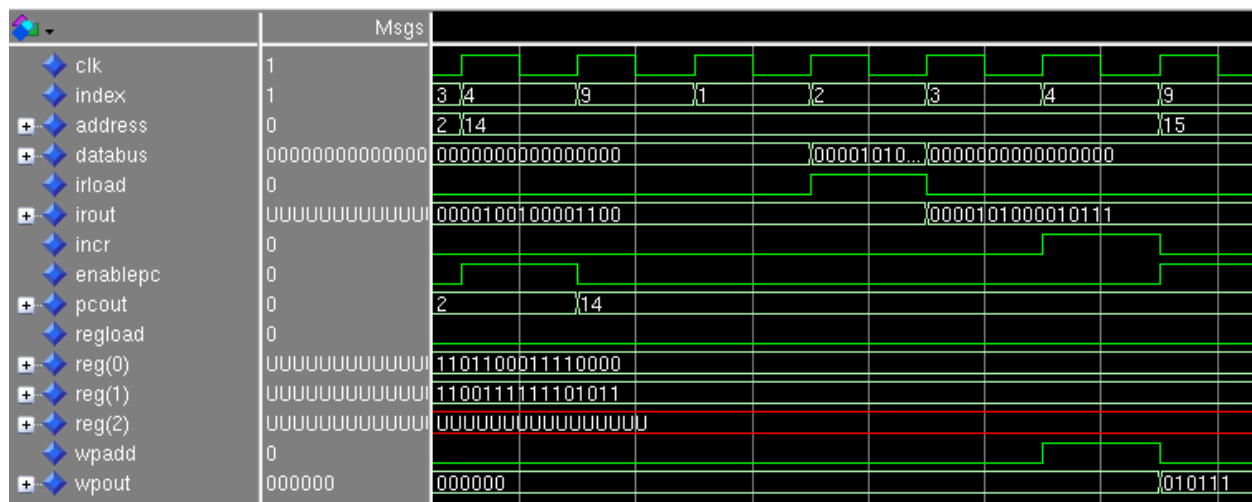


همچنین مقدار شمارنده برنامه به 2 افزایش می‌یابد. سپس دستور دوم که mvr است باید اجرا شود. این دستور سبب انتقال محتویات ثبات R0 به R3 می‌شود.

دستور موجود در خانه 2 حافظه brc است و چون $Cout = 1$ است. انشعاب انجام می‌شود. این انشعاب باعث می‌شود که محتویات شمارنده برنامه یعنی 2 با مقدار موجود در 8 بیت کم ارزش دستور یعنی 12 جمع شود. بنابراین در واکنشی بعدی به خانه 14 حافظه دسترسی می‌یابد. همانطور که شکل زیر نشان می‌دهد، مقدار شمارنده برنامه به 14 تغییر می‌یابد.



دستور موجود در خانه 14، awp است که سبب می‌شود، محتویات ثابت نشانگر یعنی صفر با مقدار موجود در 8 بیت کم ارزش دستور یعنی 23 جمع و در آن ذخیره شود. ضمناً شمارنده برنامه یک واحد افزایش می‌یابد. شکل زیر نحوه اجرای این عمل را نشان می‌دهد



بدین ترتیب دستور موجود در خانه 15 واکنشی می‌شود. چون این دستور halt است، شمارنده برنامه همان مقدار 15 باقی مانده و هیچ عملیاتی توسط برنامه انجام نمی‌شود. مطابق شکل زیر همواره کنترل کننده بین وضعیت 4 و 8 جا به جا می‌شود.

