

به نام آن که جان را فکرت آموخت



دانشگاه صنعتی شریف
دانشکده‌ی مهندسی کامپیوتر

دستور کار

آزمایشگاه معماری کامپیوتر

تهیه و تدوین:

دکتر حمید سربازی آزاد

دکتر حسین اسدی

مهندس محمدحسن خبازیان

مهندس امیرعلی حبیبی

پاییز ۱۳۸۸

فهرست مندرجات

عنوان	صفحه
۱ معرفی	۳
۱-۱ هدف	۳
۲-۱ پیش‌نیازهای نظری و عملی	۳
۳-۱ تجهیزات و نرم‌افزارهای لازم	۳
۴-۱ منابع علمی مورد نیاز و چگونگی انجام آزمایشها	۳
۵-۱ دستور تهیه گزارش کار	۴
۶-۱ مقررات آزمایشگاه و نحوه ارزیابی	۴
۲ آزمایشها	۶
۱-۲ مقدمه	۶
۲-۲ آزمایش اول: جمع کننده دهنده	۷
۱-۲-۲ هدف	۷
۲-۲-۲ شرح آزمایش	۷
۳-۲-۲ نتایج مورد انتظار	۷
۳-۲ آزمایش دوم: ضرب کننده ممیز ثابت	۸
۱-۳-۲ هدف	۸
۲-۳-۲ شرح آزمایش	۸
۳-۳-۲ نتایج مورد انتظار	۹
۴-۲ آزمایش سوم: جمع / تفریق کننده ممیز شناور	۱۰
۱-۴-۲ هدف	۱۰
۲-۴-۲ شرح آزمایش	۱۱
۳-۴-۲ نتایج مورد انتظار	۱۱

۵-۲	آزمایش چهارم: مبدل دهنده‌ی به دودوئی	۱۲
۱-۵-۲	هدف	۱۲
۲-۵-۲	شرح آزمایش	۱۲
۳-۵-۲	نتایج مورد انتظار	۱۳
۶-۲	آزمایش پنجم: واحد محاسبه با امکان انتخاب ثبات مبداء و مقصد	۱۴
۱-۶-۲	هدف	۱۴
۲-۶-۲	شرح آزمایش	۱۴
۷-۲	آزمایش ششم: کنترل توسط برنامه ذخیره شده در حافظه	۱۶
۱-۷-۲	هدف	۱۶
۲-۷-۲	شرح آزمایش	۱۶
۳-۷-۲	نتایج مورد انتظار	۱۷
۸-۲	آزمایش هفتم: استفاده از حافظه داده و دستورات پرش	۱۸
۱-۸-۲	هدف	۱۸
۲-۸-۲	شرح آزمایش	۱۸
۳-۸-۲	نتایج مورد انتظار	۲۱
۹-۲	آزمایش هشتم: واحد کنترل ریزبرنامه سازی شده	۲۲
۱-۹-۲	هدف	۲۲
۲-۹-۲	شرح آزمایش	۲۲
۲۴	منابع	

۱ معرفی

۱-۱ هدف

هدف از آزمایش‌های این جزوه تجربه عملی طراحی و پیاده‌سازی برخی از مفاهیم و روشهای مطالعه شده در درس معماری کامپیوتر می‌باشد. جزوه شامل چهار بخش می‌باشد. بخش اول به آشنایی با ابزار CAD در طراحی و آزمایش درستی عملکرد مدارات منطقی اختصاص دارد. در بخش دوم به طراحی و پیاده‌سازی معماری‌های محاسباتی، و در بخش سوم به طراحی و پیاده‌سازی معماری یک کامپیوتر ساده می‌پردازیم. بخش چهارم نیز به طراحی و پیاده‌سازی همان پردازنده است (ساخته شده در بخش سوم) اما با واحد کنترل ریزبرنامه‌پذیر اختصاص خواهد داشت. دانشجویان عزیز در گروه‌های سه نفری به انجام آزمایش‌های هر بخش می‌پردازند.

۲-۱ پیش‌نیازهای نظری و عملی

درس معماری کامپیوتر و آزمایشگاه مدار منطقی پیش‌نیاز این آزمایشگاه می‌باشند.

۳-۱ تجهیزات و نرم‌افزارهای لازم

در طول این آزمایشگاه، تجهیزاتی از قبیل شبیه‌ساز Quartus و مدارات برنامه‌پذیر مورد استفاده قرار می‌گیرد. دانشجویان لازم است که قبل از شروع آزمایش ۳ با ابزار شبیه‌ساز Quartus و قبل از آزمایش ۷ با نحوه کار با مدارات برنامه‌پذیر آشنا گردند.

۴-۱ منابع علمی مورد نیاز و چگونگی انجام آزمایشها

تعداد جلسات لازم جهت هر آزمایش یک تا دو جلسه (بسته به آزمایش) می‌باشد. دانشجویان خود مدار مورد نظر در هر آزمایش را طراحی و پس از انتخاب تراشه‌های لازم پیاده‌سازی کرده و درستی عملکرد آنرا بررسی می‌نمایند. برای هر آزمایش، طراحی مدار و انتخاب تراشه‌ها و قطعات لازم جهت پیاده‌سازی قبلاً توسط دانشجویان انجام می‌شود و روز آزمایش جهت بستن مدار و اشکال‌زدائی و نهایتاً جواب گرفتن از مدار خواهد بود. هر یک از مدارهای آزمایش‌ها در صورت لزوم باید امکان ورود اطلاعات توسط یک سری کلید (Dip-switch) را داشته و هم چنین خروجی را روی نمایشگرهایی از قبیل LED یا 7Segment نمایش دهند.

توصیه می‌شود قبل از بستن مدار نهایی، طرح مورد نظر را در یک محیط شبیه‌ساز آزمایش کرده و از درستی عملکرد آن اطمینان حاصل کنید و سپس مدار نهایی را با استفاده از تراشه‌های مناسب روی تخته مدار در آزمایشگاه پیاده‌سازی کنید. بدین ترتیب امکان ایجاد تغییرات در مدار و بررسی درستی عملکرد آن بصورت مرحله به مرحله سریعتر شده و همچنین تلفات و خرابی تراشه نیز در آزمایشگاه کاهش پیدا می‌کند. بعلاوه این روش گرفتن نصف امتیاز انجام آزمایش را تضمین می‌کند.

۱-۵ دستور تهیه گزارش کار

هر گروه موظف است برای هر آزمایش انجام شده گزارشی کامل تدوین کرده و در اولین جلسه آزمایش بعدی به مربی آزمایشگاه تحویل دهد. هر گزارش باید حداقل شامل این موارد باشد: بعنوان مقدمه مختصری راجع به مقدمات آزمایش مورد نظر آورده شده و بحث و استدلال لازم در انتخاب روش طراحی و پیاده‌سازی ذکر شود. سپس بلوک دیاگرام طرح پیشنهادی (اولیه) می‌آید و پس از آن بین دیاگرام طرح پیشنهادی (اولیه) و در انتها نیز بین دیاگرام طرح نهایی (اگر تغییر کرده باشد) به همراه بیان علت تغییرات ایجاد شده قرار می‌گیرد. گزارش‌ها باید طبق اصول ارائه مطالب علمی و فنی در تدوین گزارشهای دانشجویی تدوین شده باشند. استفاده از کتاب زیر میتواند در این زمینه مفید باشد :

عنوان : شیوه ارائه مطالب

مؤلف : سید محمد تقی روحانی رانکوهی

منتشر کننده : انتشارات جلوه

نوبت چاپ : ششم

سال نشر : ۱۳۸۰

۱-۶ مقررات آزمایشگاه و نحوه ارزیابی

۹۰٪ از نمره هر دانشجو را انجام دقیق و مرتب آزمایشها و تهیه گزارش (توسط گروه) داراست و ۱۰٪ مابقی به حضور مرتب و به موقع دانشجویان در جلسات آزمایشگاه اختصاص دارد. هر غیبت غیر موجه موجب کسر ۲ نمره از نمره کل آزمایشگاه می‌شود. ۶۰٪ از ۹۰٪ نمره هر آزمایش را انجام کامل آزمایش و جواب گرفتن دارد و ۳۰٪ مابقی را تهیه گزارش کامل از آزمایش داراست. اگر گروهی موفق به انجام کامل پیاده سازی مدار و تست کامل و موفقیت آمیز آن نشود، انجام آزمایش در محیط شبیه ساز ۴۰٪ از ۶۰٪ امتیاز انجام آزمایش را کسب خواهد کرد.

جدول ۱: جدول زمان بندی جلسات آزمایشگاه

بخش اول آشنایی با ابزار CAD	آزمایش ۱: طراحی و تست مدار جمع دو عدد دهمی دو رقمی به کمک شبیه ساز	جلسه اول
بخش دوم معماری مدارهای محاسباتی	آزمایش ۲: طراحی و پیاده سازی ضرب کننده ممیز ثابت چهاربیتی و تحویل گزارش آزمایش ۱	جلسه دوم
	آزمایش ۳: طراحی و پیاده سازی جمع / تفریق کننده ممیز شناور و تحویل گزارش آزمایش ۲	جلسه سوم
	آزمایش ۴: طراحی و پیاده سازی مبدل دهمی به دودویی و تحویل گزارش آزمایش ۳	جلسه چهارم
		جلسه پنجم
بخش سوم معماری یک کامپیوتر ساده	آزمایش ۵: طراحی و پیاده سازی واحد محاسبه با امکان انتخاب مبداء و مقصد و تحویل گزارش آزمایش ۴	جلسه ششم
	آزمایش ۶: طراحی و پیاده سازی واحد محاسبه با امکان کنترل توسط برنامه و تحویل گزارش آزمایش ۵	جلسه هفتم
	آزمایش ۷: طراحی و پیاده سازی کامل کامپیوتر با حافظه داده و دستورات پرش و تحویل گزارش آزمایش ۶	جلسه هشتم
		جلسه نهم
بخش چهارم معماری ریز برنامه پذیر	آزمایش ۸: طراحی و پیاده سازی مدار کنترل ریزبرنامه پذیر و تحویل گزارش آزمایش ۷ و ۸	جلسه دهم
		جلسه یازدهم

۲ آزمایش‌ها

۱-۲ مقدمه

دانشجویان لازم است که در ابتدای هر آزمایش، هدف آزمایش، شرح آزمایش و نتایج مورد انتظار را مطالعه کرده و با نحوه انجام آزمایش قبل از حضور در آزمایشگاه آشنا باشند. همچنین برای هر آزمایش، طراحی مدار و انتخاب تراشه‌ها و قطعات لازم جهت پیاده‌سازی قبلاً توسط دانشجویان بایستی انجام پذیرد. در صورتی که مطالب لازم در مورد آزمایش را به لحاظ تئوری فراموش کرده‌اید، لازم است از مراجع مربوط و پیش از جلسه آزمایشگاه، مطالعه کرده و با آگاهی کامل در کلاس حضور بهم رسانید.

۲-۲ آزمایش اول: جمع کننده ددهی

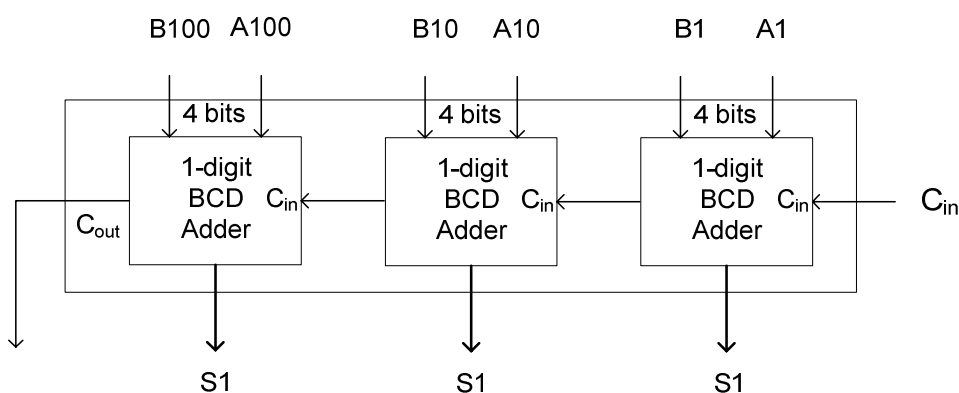
۲-۲-۱ هدف

هدف از این آزمایش آشنایی با نحوه کار یک جمع کننده ددهی می باشد. در این آزمایش دو عدد ۳ رقمی در مبنای ده به مدار داده شده و نتیجه مورد انتظار در خروجی مشاهده می شود.

۲-۲-۲ شرح آزمایش

در تمامی آزمایشهایی که در این آزمایشگاه انجام می شوند استفاده از شبیه سازی قبل از پیاده سازی نهایی مدار قویاً توصیه می شود. این جلسه از آزمایشگاه به آشنایی با یک شبیه ساز اختصاص دارد. بدین منظور لازم است که یک مدار جمع کننده ددهی سه رقمی را طراحی کرده و به کمک شبیه ساز درستی عملکرد آنرا بررسی کنید. اغلب برای سادگی کار و همچنین راحتی اشکال زدایی لازم است که طراحی را بصورت سلسله مراتبی انجام دهیم. بدین منظور می توانید:

- ۱- ابتدا یک بلوک تمام جمع کننده تک بیتی طرح کنید.
 - ۲- سپس با استفاده از بلوک طراحی شده، یک بلوک جمع کننده ددهی یک رقمی طراحی کنید.
 - ۳- نهایتاً با استفاده از بلوک طرح شده در قسمت ۲، یک جمع کننده سه رقمی طراحی کنید.
- پس از طراحی مدار درستی عملکرد آنرا با ورودی های مختلف آزمایش کنید.



شکل ۱: بلوک دیاگرام جمع کننده ددهی ۳ رقمی

۲-۲-۳ نتایج مورد انتظار

در این آزمایش، چندین سری عدد سه رقمی در مبنای ده به ورودی مدار داده شده و انتظار می رود که جمع این اعداد در خروجی مشاهده شود.

۳-۲ آزمایش دوم: ضرب کننده ممیز ثابت

۳-۲-۱ هدف

در این جلسه یک مدار ضرب کننده دو دویی چهاربیتی را طراحی و پیاده سازی می‌کنیم. مشخصات مدار مورد نظر به قرار زیر است:

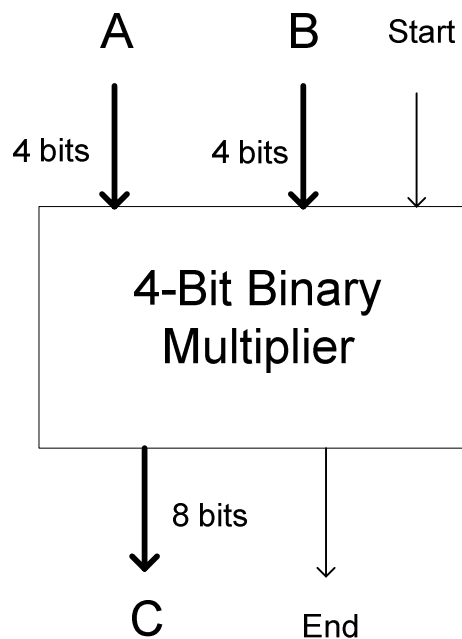
A : مضروب (ورودی)

B : مضروب فیه (ورودی)

C : حاصل ضرب (خروجی)

Start : شروع ضرب (ورودی)

End : پایان ضرب (خروجی)



شکل ۲: ضرب کننده ممیز ثابت

۳-۲-۲ شرح آزمایش

با فعال شدن سیگنال Start ضرب کننده شروع به کار کرده و حاصل ضرب دو عدد ورودی چهار بیتی A و B را محاسبه می‌کند (به روش shift & add) و پس از اتمام عملیات حاصل ضرب ۸ بیتی را روی خطوط C قرار می‌دهد و با فعال کردن سیگنال End پایان عملیات را خبر می‌دهد.

هنگام طراحی مدار به کمک شبیه‌ساز سعی کنید از تراشه‌های TTL موجود در کتابخانه شبیه‌ساز و آزمایشگاه استفاده کنید. بدین ترتیب هنگام پیاده‌سازی عملی نیازی به تغییر مدار برای استفاده از تراشه‌های موجود نیست.

۲-۳-۳ نتایج مورد انتظار

در این آزمایش ضرب دو عدد دودویی با فعال شدن سیگنال Start محاسبه می‌شود. انتظار می‌رود نتیجه درست بعد از چند سیکل ساعت بسته در زمان فعال شدن سیگنال End در خروجی دیده شود.

۴-۲ آزمایش سوم: جمع / تفریق کننده ممیز شناور

۴-۲-۱ هدف

در این آزمایش (طی دو جلسه) مدار یک جمع / تفریق کننده ممیز شناور را طراحی کرده و با ابزار Quartus شبیه‌سازی می‌نماییم. پس از اطمینان از صحت عملکرد در شبیه‌ساز، بر روی برد پیاده‌سازی می‌کنیم. مدار اولیه برای شبیه‌سازی را مطابق استاندارد IEEE-754 ۳۲ بیتی طراحی نمایید. برای سهولت پیاده‌سازی بر روی برد تعداد بیت‌ها را از ۳۲ به ۱۲ کاهش دهید. مشخصات مدار مورد نظر به قرار زیر است:

عملوند اول (ورودی): A

عملوند دوم (ورودی): B

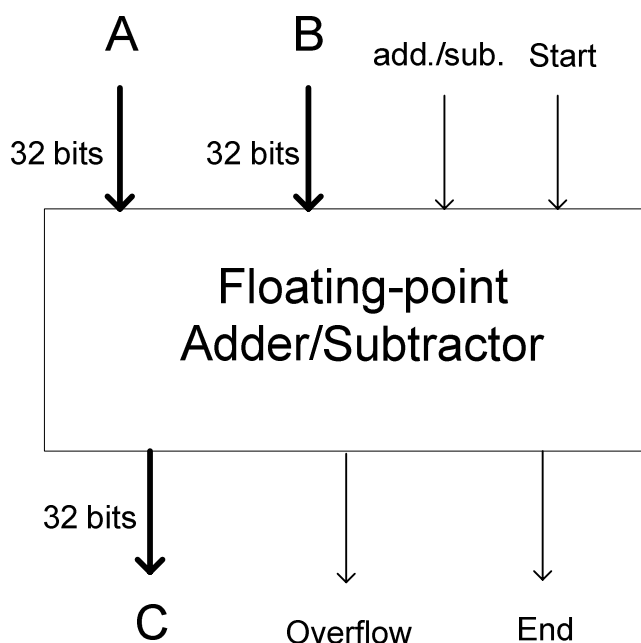
حاصل جمع / تفریق (خروجی): C

شروع عملیات (ورودی): Start

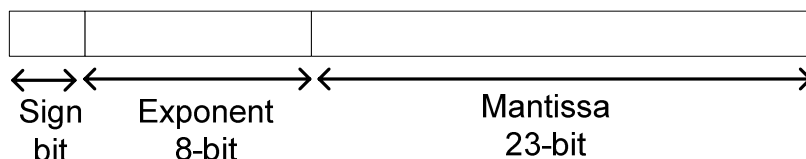
پایان عملیات (خروجی): End

سرریزی (خروجی): Overflow

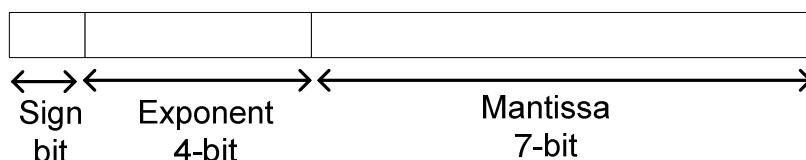
مشخص کننده جمع یا تفریق (ورودی): add/sub



شکل ۳: جمع / تفریق کننده ممیز شناور



شکل ۴- (الف) فرمت اعداد ممیز شناور استاندارد IEEE-754 برای طراحی در شبیه‌ساز



شکل ۴- (ب): فرمت اعداد ممیز شناور برای پیاده‌سازی بر روی برد

۲-۴-۲ شرح آزمایش

ابتدا مدار جمع و تفریق کننده دو عدد ممیز شناور (شکل ۳) را با فرمت استاندارد IEEE-754 مطابق شکل ۴- (الف) طراحی کرده و با ابزار Quartus شبیه‌سازی نمایید. پس از اطمینان از صحت عملکرد، طراحی انجام شده را بر روی برد پیاده‌سازی نمایید. برای سهولت پیاده‌سازی تعداد بیت‌های مدار طراحی شده را مطابق شکل ۴- (ب) از ۳۲ بیت به ۱۲ بیت کاهش دهید. با فعال شدن سیگنال Start مدار شروع به کار کرده و اگر سیگنال add/sub برابر صفر باشد، مقدار $A + B$ و اگر این سیگنال برابر یک باشد، مقدار $A - B$ را محاسبه کرده و روی خطوط C قرار می‌دهد و سیگنال End را به منزله اتمام عملیات فعال می‌کند. ورودی‌های A و B نرمالیزه بوده و خروجی C نیز باید نرمالیزه باشد. در صورت بروز سرریزی سیگنال Overflow فعال می‌شود. استفاده از شمارنده با قابلیت شمارش رو به بالا و پایین برای نگهداری نما در طراحی می‌تواند حجم مدار را کاهش دهد.

۲-۴-۳ نتایج مورد انتظار

در این آزمایش جمع یا تفریق دو عدد دودویی ممیز شناور با فعال شدن سیگنال Start محاسبه می‌شود. انتظار می‌رود نتیجه‌ی درست بعد از چند سیکل ساعت بسته به تفاوت دو نما با فعال شدن سیگنال End در خروجی دیده شود.

۲-۵ آزمایش چهارم: مبدل دهمی به دودویی

۲-۵-۱ هدف

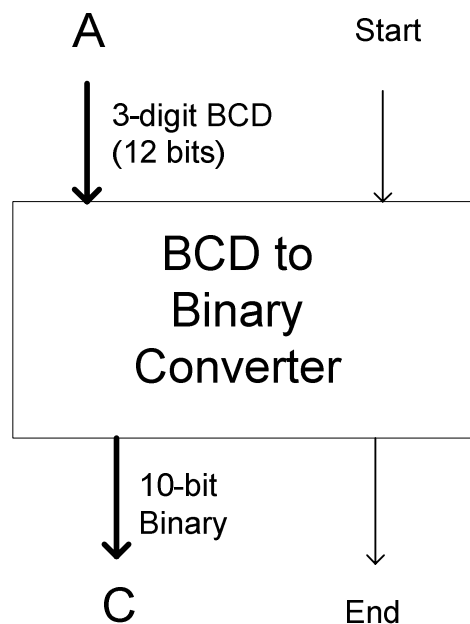
در این جلسه مدار یک مبدل دهمی به دودویی را طراحی کرده و با ابزار Quartus شبیه‌سازی می‌نماییم. پس از اطمینان از صحت عملکرد در شبیه‌ساز، مدار را بر روی برد پیاده‌سازی می‌کنیم. مشخصات مدار مورد نظر به قرار زیر است:

عدد دهمی (ورودی) : A

معادل دودویی (خروجی) : B

شروع عملیات (ورودی) : Start

پایان تبدیل (خروجی) : End



شکل ۵: مبدل دهمی به دودویی

۲-۵-۲ شرح آزمایش

با فعال شدن سیگنال Start مدار شروع به کار کرده و ورودی دهمی را که یک عدد سه رقمی (برای سادگی، در پیاده‌سازی روی برد اعداد دو رقمی در نظر بگیرید) است به معادل دودویی آن تبدیل کرده

- و حاصل را روی خطوط خروجی می‌گذارد و سیگنال End را به منزله اعلام اتمام عملیات فعال می‌کند. الگوریتم تبدیل یک عدد دهدهی r رقمی به دودویی معادل به صورت زیر است:
- الف - عدد دهدهی ورودی را یک بیت به راست شیفت دهید .
- ب - اگر با ارزشترین بیت رقم نام یک باشد از آن رقم ۳ تا کم کنید ($1 \leq i < r$).
- ج- مراحل الف و ب را آنقدر تکرار کنید تا تمام ارقام دهدهی صفر شوند (حداکثر ۱۰ بار تکرار لازم است).
- در پایان بیت‌هایی که بوسیله شیفت بر راست بیرون می‌آیند ، عدد دودویی معادل عدد دهدهی ورودی را تشکیل می‌دهند.
- در مثال زیر عدد دهدهی 110 طبق این الگوریتم به دودویی معادل تبدیل شده است:

عمل	خروجی	رقم ۱	رقم ۲	رقم ۳
شیفت به راست	0	0000	0001	0001
ازرقمهای 1 و 2 سه تا کم کن	0	1000	1000	0000
شیفت به راست	0	0101	0101	0000
از رقم 1 سه تا کم کن	10	1010	0010	0000
شیفت به راست	10	0111	0010	0000
شیفت به راست	110	0011	0001	0000
از رقم 1 سه تا کم کن	1110	1001	0000	0000
شیفت به راست	1110	0110	0000	0000
شیفت به راست	01110	0011	0000	0000
شیفت به راست	101110	0001	0000	0000
پایان عملیات	1101110	0000	0000	0000

۲-۵-۳ نتایج مورد انتظار

در این آزمایش انتظار می‌رود عدد دودویی معادل سه رقمی دهدهی ورودی با فعال شدن سیگنال Start محاسبه شود و پایان کار با سیگنال End مشخص شود.

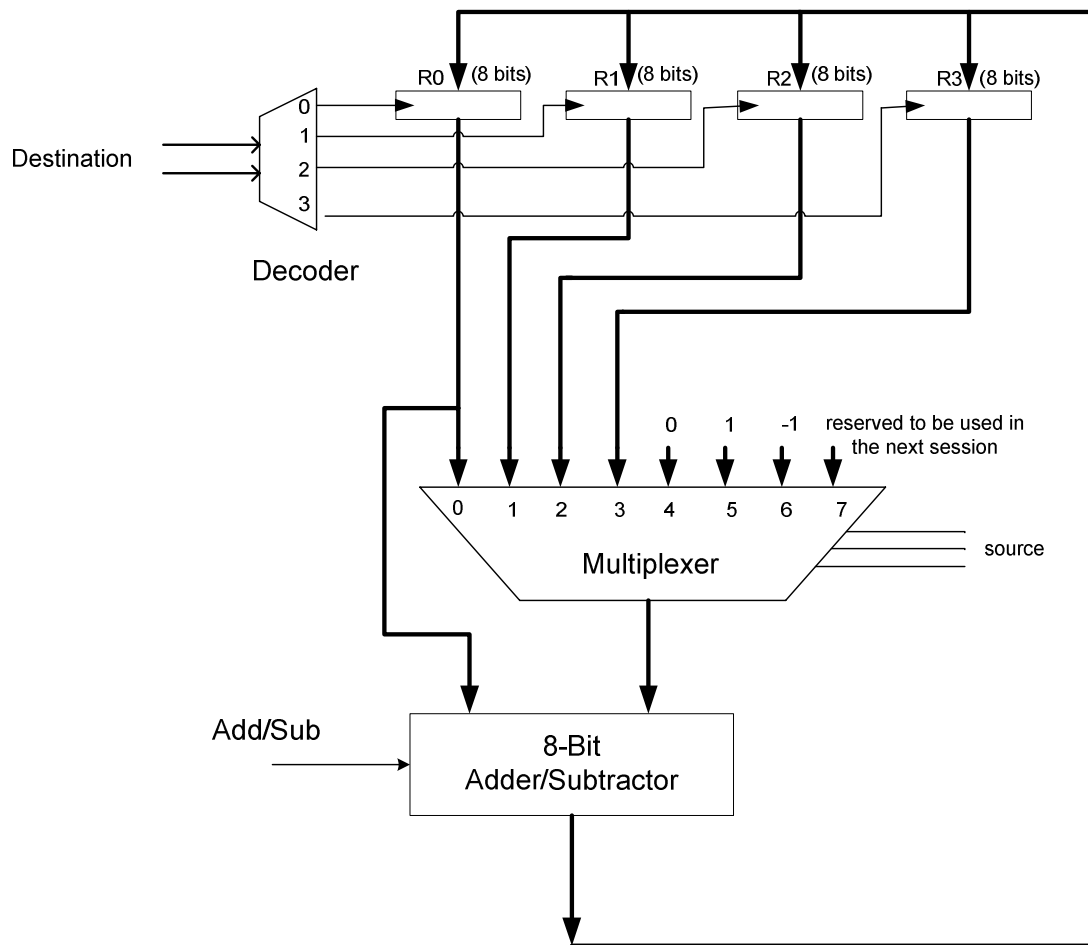
۲-۶ آزمایش پنجم: واحد محاسبه با امکان انتخاب ثبات مبداء و مقصد

۲-۶-۱ هدف

طی آزمایش‌های پنجم، ششم و هفتم یک کامپیوتر ساده را بطور کامل طراحی و پیاده‌سازی کرده و برنامه‌ای را به زبان ماشین نوشته روی آن اجرا می‌کنیم.

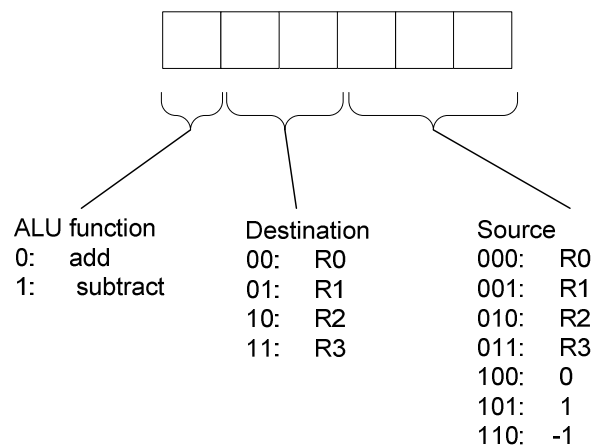
۲-۶-۲ شرح آزمایش

در این آزمایش، واحد محاسبات و مجموعه ثباتهای عمومی ماشین را طراحی و پیاده‌سازی می‌کنیم. معماری مورد نظر در شکل ۶ نشان داده شده است. این معماری امکان انجام جمع و تفریق با امکان انتخاب ثبات‌های مبداء و ثبات نگهدارنده نتیجه (مقصد) را دارد. چهار ثبات عمومی $R0$ ، $R1$ ، $R2$ و $R3$ هشت بیتی هستند. همانطور که در شکل پیداست، یکی از عملوندهای ALU به صورت ثابت محتوای ثبات $R0$ و دیگری می‌تواند محتوای یکی از ثباتهای $R0$ تا $R3$ و یا مقادیر ثابت 0 و 1 و -1 باشد. حاصل تولید شده توسط ALU (جمع / تفریق) به یکی از ثباتهای مقصد $R0$ تا $R3$ منتقل می‌شود.



شکل ۶: معماری واحد محاسبات

این معماری را طوری پیاده‌سازی کنید که قابلیت انجام فرمانهای شش بیتی زیر را داشته باشد:



شکل ۷: قالب فرمانهای شش بیتی

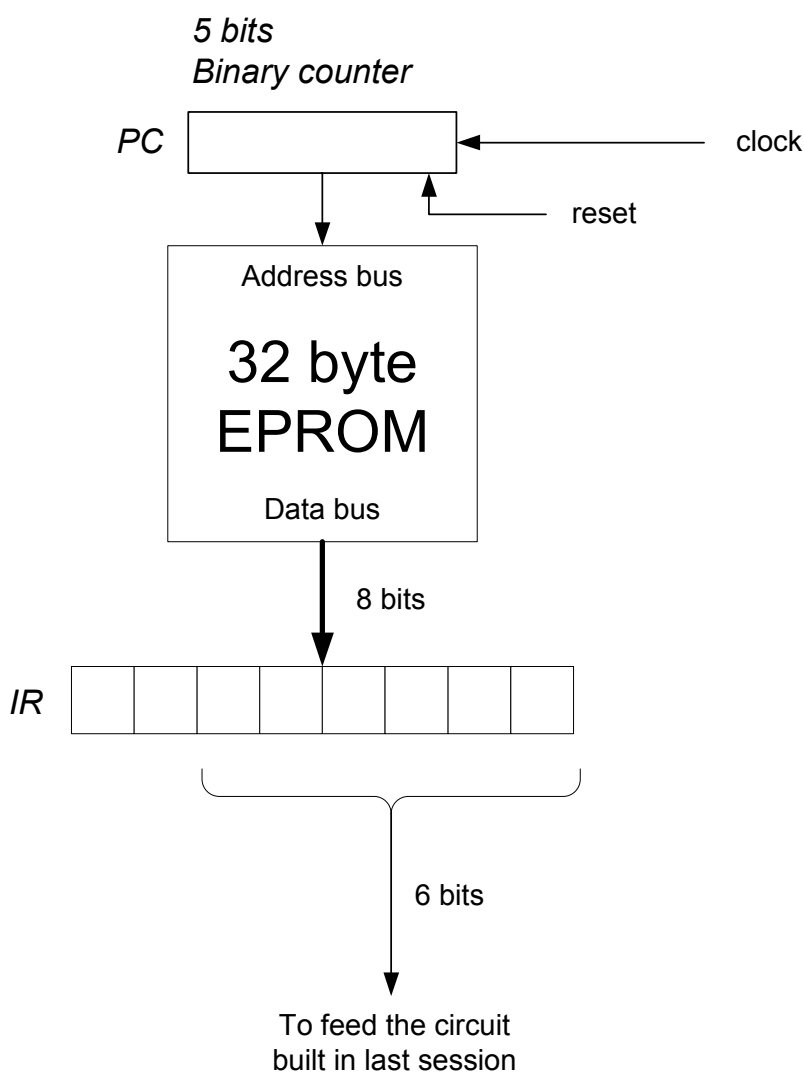
۷-۲ آزمایش ششم: کنترل توسط برنامه ذخیره شده در حافظه

۱-۷-۲ هدف

هدف از این آزمایش آشنایی با نحوه واکنشی دستورات در پردازنده‌ها می‌باشد.

۲-۷-۲ شرح آزمایش

در این آزمایش فرمانهای لازم جهت کنترل مدار آزمایش پنجم را از برنامه ذخیره شده در یک حافظه EPROM می‌گیریم. فرمانها به ترتیب توسط یک شمارنده (PC) آدرس شده، پس از واکنشی از حافظه دستورات اجرا می‌گردند. بدین منظور لازم است که به مدار آزمایش پنجم مدارات لازم اضافه گردد. شکل ۸ بلوک دیاگرام سیستم را نشان می‌دهد.



شکل ۸: بلوک دیاگرام سیستم

پس از اضافه کردن قسمت‌های لازم به مدار آزمایش پنجم، برنامه زیر را کد کرده و در حافظه EPROM ذخیره کنید و سپس توسط معماری پیاده‌سازی شده اجرا کنید.

برنامه تولید شش جمله از سری فیبوناچی

در سری فیبوناچی دو جمله اول 0 و 1 هستند و مقدار هر جمله دیگر حاصل جمع مقادیر دو جمله ما قبل آن است. سری اعداد فیبوناچی مطابق تابع زیر تولید می‌شود:

$$F(n) := \begin{cases} 0 & \text{if } n = 0; \\ 1 & \text{if } n = 1; \\ F(n-1) + F(n-2) & \text{if } n > 1. \end{cases}$$

قطعه برنامه زیر ده جمله اول این سری را در ثباتهای R0 و R1 تولید می‌کند.

Address	Code	Instruction	Comment	
00000		Sub R0.R0	Clear R0	جمله اول در R0
		Add R1, 1	$R1 \leftarrow 1$	جمله دوم در R1
		Add R0.R1	$R0 \leftarrow 1$	جمله سوم در R0
		Add R1.R0	$R1 \leftarrow 2$	جمله چهارم در R1
		Add R0.R1	$R0 \leftarrow 3$	جمله پنجم در R0
		Add R1.R0	$R1 \leftarrow 5$	جمله ششم در R1
		Add R0.R1	$R0 \leftarrow 8$	جمله هفتم در R0
		Add R1.R0	$R1 \leftarrow 13$	جمله هشتم در R1
		Add R0.R1	$R0 \leftarrow 21$	جمله نهم در R0
		Add R1.R0	$R1 \leftarrow 34$	جمله دهم در R1

۲-۳ نتایج مورد انتظار

در این آزمایش انتظار می‌رود که سیگنال‌های کنترلی در آزمایش قبل را از یک حافظه با روند ترتیبی واکنشی و اجرا نماید.

۸-۲ آزمایش هفتم: استفاده از حافظه داده و دستورات پرش

۸-۱-۲ هدف

در آزمایش ششم امکان استفاده از حافظه داده جهت ذخیره داده های بینابینی را نداشتیم. همچنین کمبود دستورات پرش و عدم امکان وجود حلقه در برنامه، به خوبی حس می شد. در این آزمایش مدار آزمایش ششم را تکمیل کرده و آنرا به یک کامپیوتر ساده با امکان دستیابی به حافظه داده جهت خواندن و ذخیره داده و همچنین امکان استفاده از دستورات پرش شرطی و غیر شرطی تبدیل می کنیم.

۸-۱-۲ شرح آزمایش

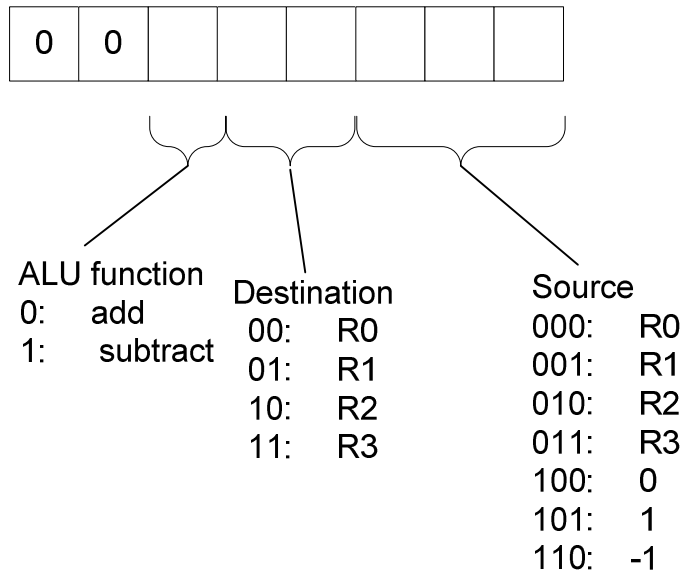
برای حافظه داده ها از یک RAM با گنجایش ۳۲ بیت استفاده می کنیم. شکل زیر معماری کامپیوتر مورد نظر را نشان می دهد. (به اینگونه معماری که در آن حافظه برنامه از حافظه داده جداست ، معماری Harvard می گویند) دستورات این ماشین (به انضمام دستورات محاسباتی قابل اجرا در آزمایش ششم) به سه گروه دستورات محاسباتی - انتقالی ، دستورات دسترسی به حافظه داده، و دستورات پرش شرطی و غیر شرطی تقسیم می شوند.

مراحل شبیه سازی

آزمایش ۷ و ۸ ابتدا توسط ابزار شبیه ساز Quartus پیاده سازی گردیده و نتایج مورد انتظار بر روی شبیه ساز مشاهده شود. پس از اطمینان از صحت شبیه سازی، این مدارها بر روی بوردهای آزمایشگاهی برنامه پذیر، پیاده سازی گردیده و نتایج مورد انتظار بر روی بوردهای آزمایشگاهی برنامه پذیر مشاهده گردند. در پایان، پس از پیاده سازی بر روی بوردهای برنامه پذیر، آزمایش ۷ و ۸ در بوردهای آزمایشگاه پیاده سازی فیزیکی گردند.

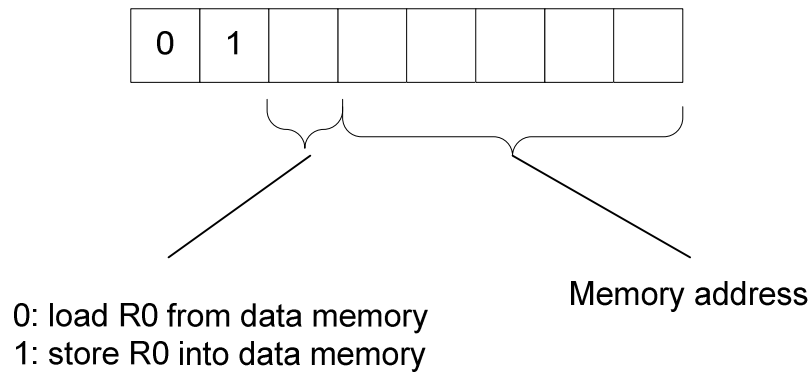
دستورات محاسباتی - انتقالی

این دستورات شامل عملیات انتقال داده بین ثبات ها و عملیات حسابی می شود . قالب این دستورات به صورت زیر است:



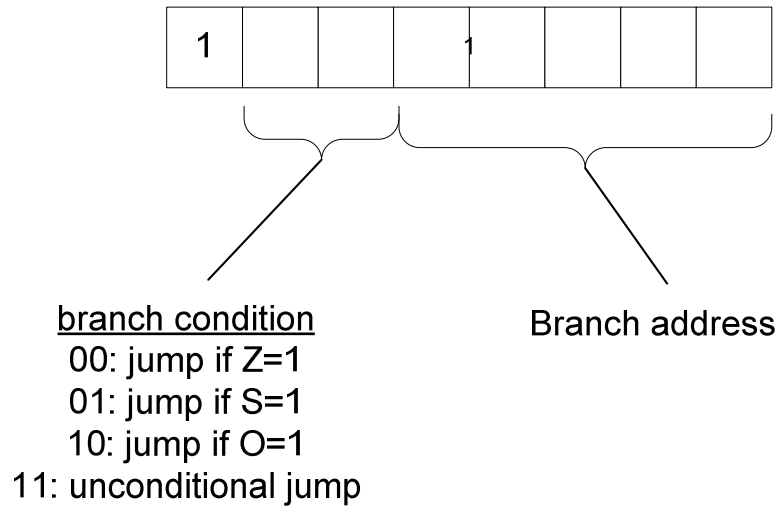
دستورات دسترسی به حافظه داده

این دسته از دستورات شامل دستور بار کردن محتویات خانه ای از حافظه در ثبات R0 و دستور ذخیره محتویات ثبات R0 در خانه ای از حافظه است. قالب این دستورات به صورت زیر است:

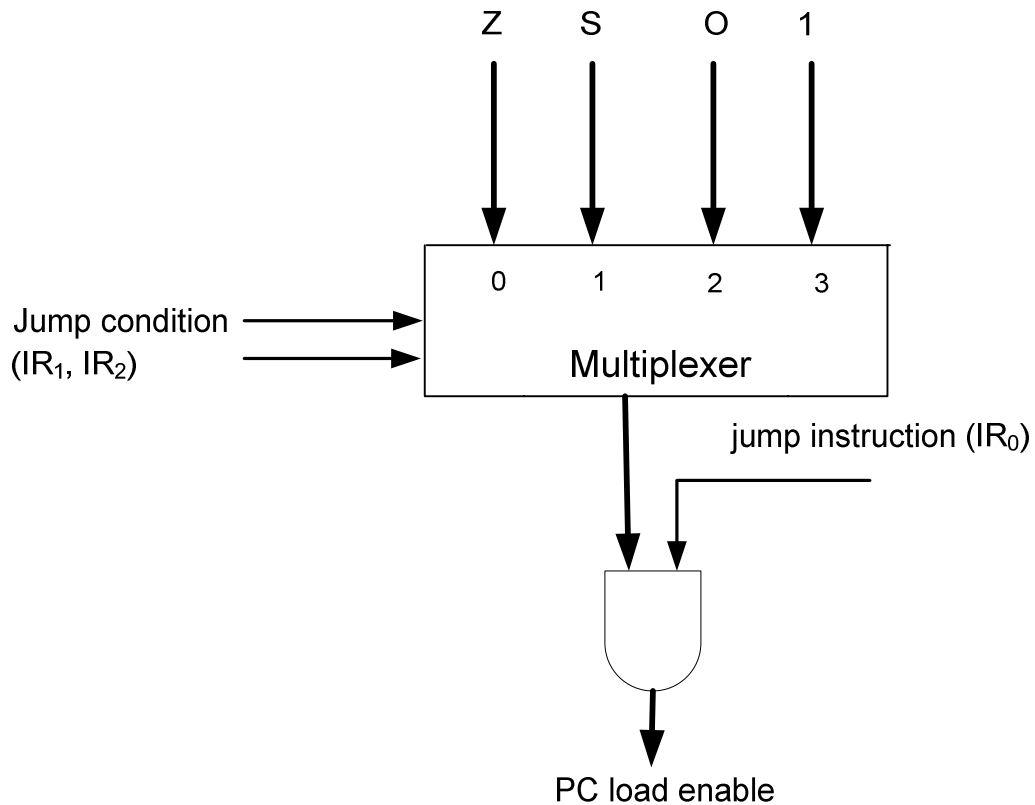


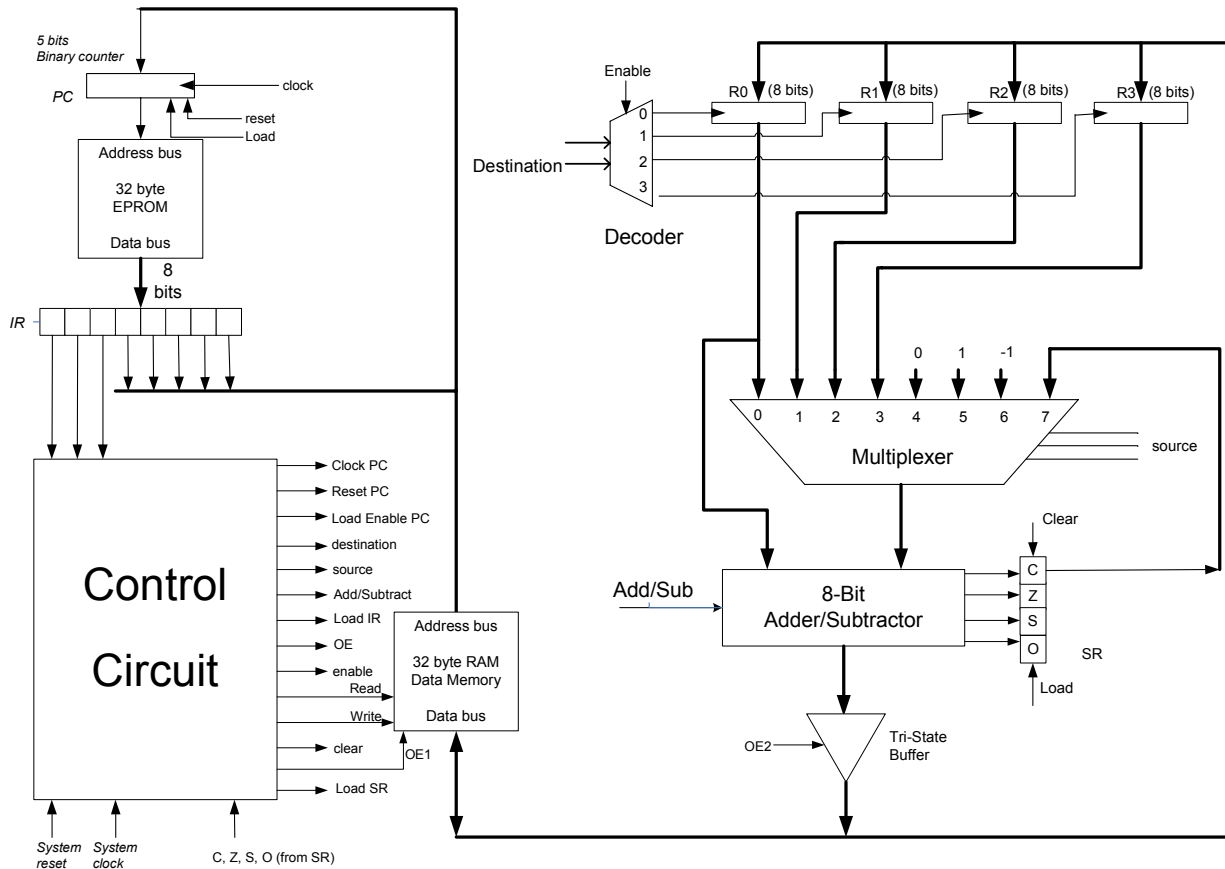
دستورات پرش (شرطی و غیر شرطی)

این دستورات شامل پرشهای شرطی و غیر شرطی به آدرس دلخواه در حافظه دستور می باشد. قالب این دستورات بصورت زیر است:



توجه کنید که سیگنال Load شمارنده برنامه (PC) می تواند توسط مدار زیر تولید شود:





شکل ۹: بلوک دیاگرام کلی کامپیوتر ساده

پس از پیاده‌سازی و حصول اطمینان از درستی عملکرد سیستم، برنامه‌ای به زبان ماشین بنویسید که: الف) مجموع ده جمله اول سری فیبوناچی را محاسبه کرده و در آدرس صفر حافظه داده‌ها ذخیره کند (با استفاده از حلقه).

ب) برنامه‌ای بنویسید که دو عدد ۶۴ بیتی ذخیره شده در آدرس صفر و ۸ حافظه داده را جمع کرده و حاصل ۶۴ بیتی را در آدرس ۱۶ حافظه داده ذخیره کند.

برنامه‌های بالا را کد کرده و در حافظه برنامه ذخیره کنید و سپس اجرا کنید. لازم است پردازنده پس از انجام این برنامه متوقف گردد و از پیشروی برای اجرای دستورات بعدی بازماند. بدین منظور می‌توان در انتهای برنامه پس از آخرین دستور یک دستور پرش غیرشرطی به آدرس خود دستور پرش داشت.

۲-۸-۳ نتایج مورد انتظار

در این آزمایش انتظار می‌رود که امکانات خواندن و نوشتن در حافظه و دستورات پرش به آزمایش قبلی اضافه شود.

۹-۲ آزمایش هشتم: واحد کنترل ریز برنامه سازی شده

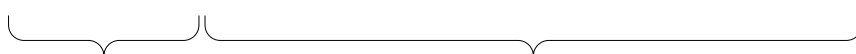
۹-۲-۱ هدف

در این آزمایش مدار کنترل کامپیوتر ساخته شده در آزمایشهای پنجم، ششم، و هفتم را بصورت ریز برنامه پذیر طراحی و پیاده سازی می کنیم.

۹-۲-۲ شرح آزمایش

قالب ریز دستورات بصورت زیر می باشد که شامل ۸ نوع ریز دستورات است:

0	1						
---	---	--	--	--	--	--	--



μ OpCode

Control signals / jump address

000: normal (no jump)

001: jump if IR0=1

010: jump if IR1=1

011: jump if IR2=1

100: jump if z=1

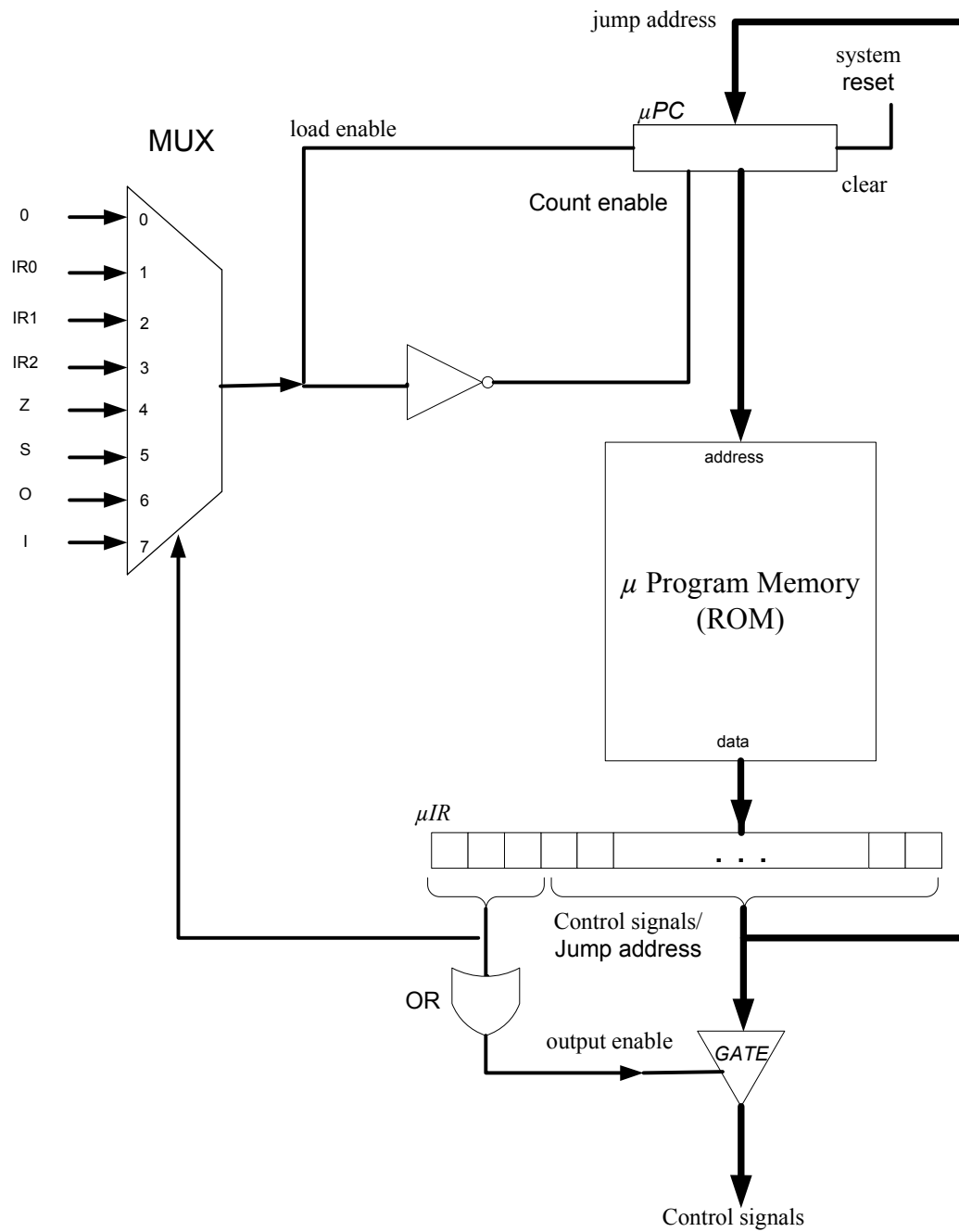
101: jump if S=1

110: jump if O=1

111: jump always

شکل ۱۰: قالب ریز دستورات

بلوک دیاگرام ریز معماری اجرای ریز دستورات در شکل ۱۱ آمده است. گنجایش ریز حافظه را ۲۵۶ کلمه فرض کنید. مدار کامل کامپیوتر ساده را که بخش کنترل آن با سیستم ریز برنامه پذیر بالا عمل می کند طراحی و پیاده سازی کنید. ابتدا ریز برنامه های مراحل واکنشی و اجرای دستورات مختلف ماشین را نوشته و در ریز حافظه ذخیره کنید. پس از حصول اطمینان از درستی عملکرد سیستم، برنامه الف آزمایش هفتم (جمع ۱۰ جمله اول از سری فیبوناچی) را روی ماشین اجرا کنید.



شکل ۱۱: بلوک دیاگرام ریزمعماری اجرا کننده ریز دستورات

منابع

- [1] “Computer Organization & Design, The Hardware / Software Interface”, D. Patterson and J. L. Hennessy, Morgan Kaufmann Publishing, 2005.
- [2] “Computer system architecture”, M. Mano, 3rd Edition, Prentice hall, 1992.