

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

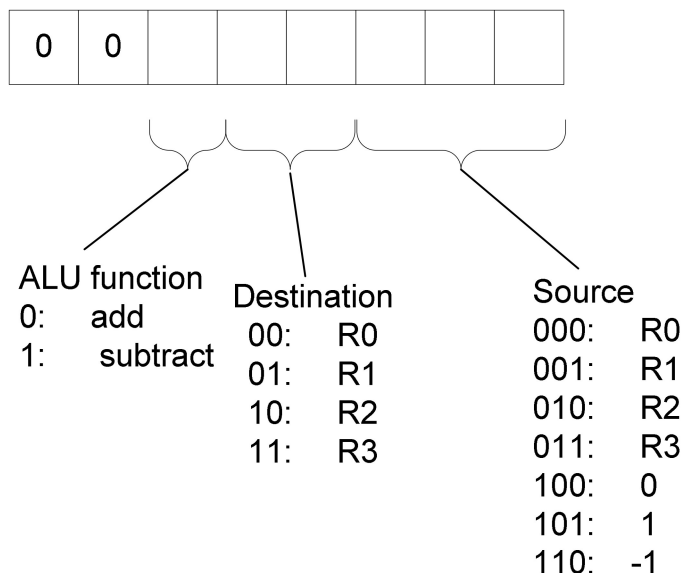
برای حافظه داده ها از یک RAM با گنجایش ۳۲ بیت استفاده می کنیم. شکل زیر معماری کامپیوتر مورد نظر را نشان می دهد. (به اینگونه معماری که در آن حافظه برنامه از حافظه داده جداست ، معماری Harvard می گویند) دستورات این ماشین (به انضمام دستورات محاسباتی قابل اجرا در آزمایش ششم) به سه گروه دستورات محاسباتی -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

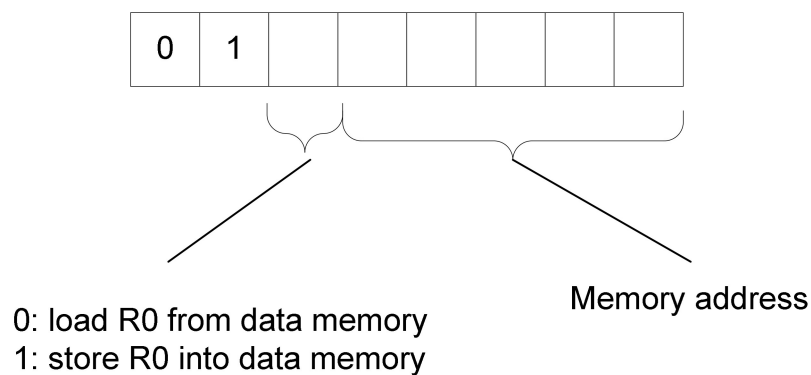
دستورات محاسباتی - انتقالی

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



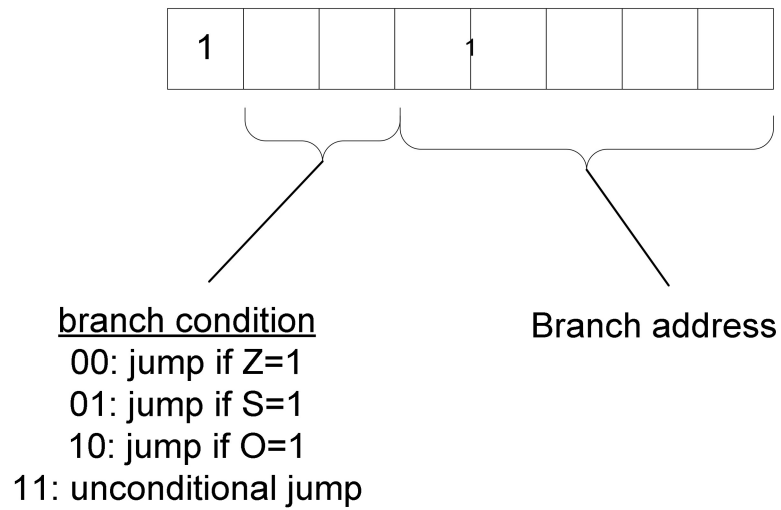
دستورات دسترسی به حافظه داده

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:

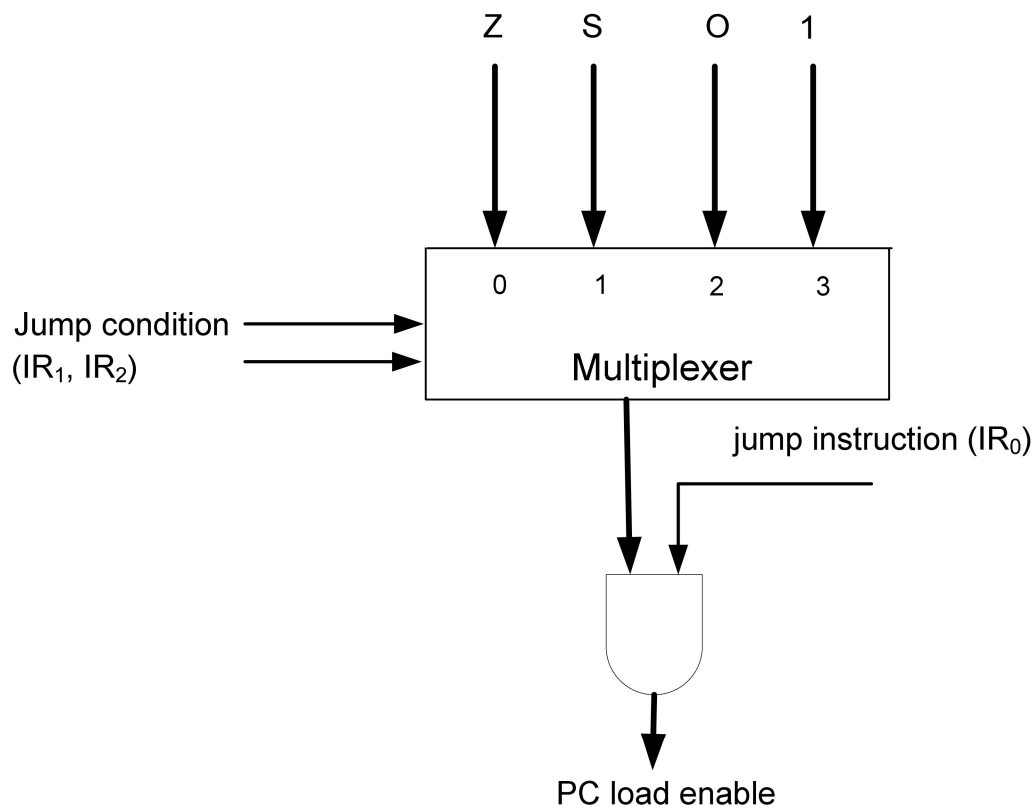


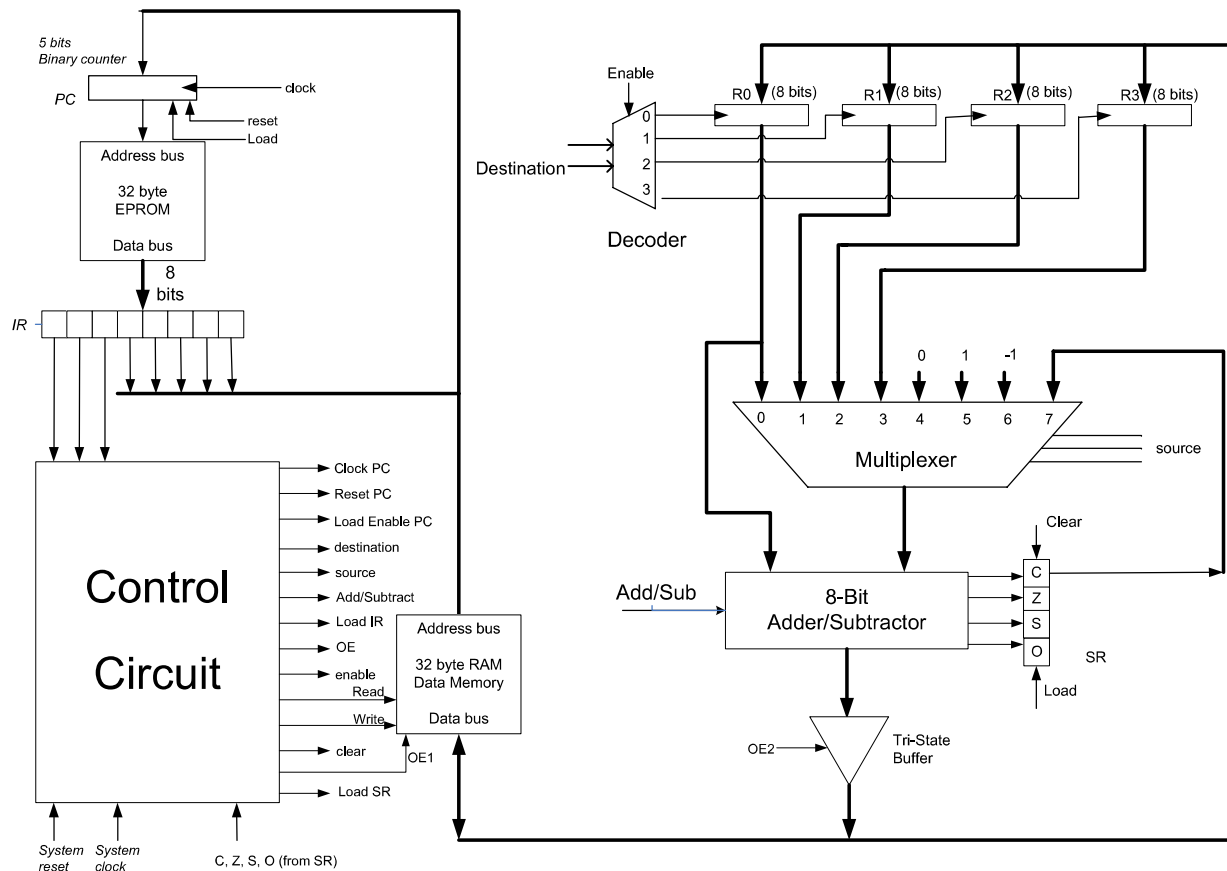
دستورات پرش (شرطی و غیر شرطی)

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) مجموعه 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.

۲-۳-۸ نتایج مورد انتظار

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