

بسمه تعالی



گزارش کار هفتم آزمایشگاه معماری کامپیوتر

کنترل توسط برنامه ذخیره شده در حافظه

استاد: دکتر سربازی

دستیار آموزشی: سرکار خانم غیبی

نویسندگان

مریم شیران ۴۰۰۱۰۹۴۴۶

فرزام کوهی رونقی ۴۰۱۱۰۶۴۰۳

ثنا بابایان ونستان ۴۰۱۱۰۵۶۸۹

دانشگاه صنعتی شریف

تابستان ۱۴۰۳

فهرست مطالب

۱ ۱
۱-۱ ۱
۱-۲ ۱
۱-۳ ۴
۱-۳-۱ ۴
۱-۳-۲ ۴
۱-۴ ۸

۱ آزمایش هفتم: کنترل توسط برنامه ذخیره شده در حافظه

۱-۱ مقدمه و هدف

هدف این آزمایش، آشنایی با چگونگی واکنش‌های پردازنده‌ها به دستورات مختلف است. در این آزمایش، مجموعه فرمان‌های مورد نیاز برای کنترل مدار آزمایش پنجم در حافظه‌ای که برنامه در آن ذخیره شده است (حافظه EPROM) قرار داده می‌شود. این فرمان‌ها به ترتیب توسط شمارنده برنامه (PC) آدرس‌دهی شده و سپس از حافظه خوانده و اجرا می‌شوند. برای انجام این کار، مدارهای مورد نیاز باید به مدار آزمایش ششم اضافه شوند. شکل زیر دیاگرام بلوک سیستم را نمایش می‌دهد.

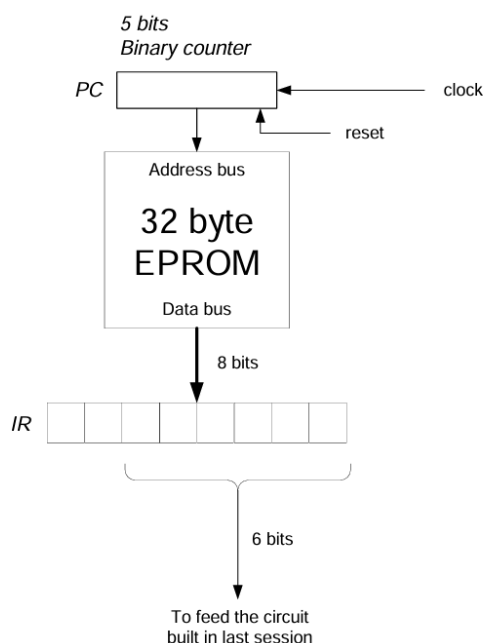


Figure ۱ شمای کلی مدار

ما قصد داریم که حافظه را به گونه ای برنامه ریزی کنیم که ۶ جمله اول سری فیبوناچی تولید شوند.

۲-۱ شرح دستگاه ها و وسایل مورد استفاده

برای شبیه سازی از نرم افزار Proteus استفاده میشود. در خود این نرم افزار از قطعات زیر استفاده میکنیم.

مدارهای مجتمع (ICها):

۱. ۷۴LS83 (U24, U25)

- عملکرد: این‌ها جمع‌کننده‌های کامل باینری ۴ بیتی هستند. آن‌ها اعداد باینری را جمع می‌کنند و خروجی آن‌ها شامل حاصل جمع و خروجی حمل است. در مدار، برای عملیات حسابی استفاده می‌شوند.

۲. ۷۴LS153 (U6, U7, U8, U9, U10, U11, U12, U13)

- عملکرد: این‌ها انتخاب‌کننده‌های داده دوگانه ۴ خط به ۱ خط هستند. آن‌ها یکی از چهار خط ورودی را بر اساس ورودی‌های کنترلی انتخاب می‌کنند و به خروجی ارسال می‌کنند. این ICها برای انتخاب داده استفاده می‌شوند.

۳. ۷۴LS198 (U1, U2, U3, U4)

- عملکرد: این‌ها شمارنده‌های باینری ۴ بیتی با قابلیت شمارش صعودی و نزولی هستند که قابلیت پاک کردن (clear) دارند. آن‌ها می‌توانند بر اساس ورودی‌های کنترلی به صورت صعودی یا نزولی شمارش کنند. در مدار، این ICها برای نگه داشتن شمارش باینری یا اجرای یک مکانیسم شمارش استفاده می‌شوند.

۴. ۴۰۴۳ (U5A)

- عملکرد: این یک فلیپ‌فلاپ NOR چهارگانه است. از آن برای ذخیره داده استفاده می‌شود؛ خروجی‌ها حالت منطقی آخر را تا زمانی که ورودی‌ها آن را تغییر دهند، حفظ می‌کنند.

۵. ۷۴LS85 (U22)

- عملکرد: این یک مقایسه‌گر مقدار ۴ بیتی است که دو عدد باینری ۴ بیتی را مقایسه می‌کند و نشان می‌دهد که آیا یکی از آن‌ها بزرگتر، کوچکتر یا برابر با دیگری است.

۶. ۷۴LS47 (U21)

- عملکرد: این یک مبدل BCD به نمایشگر ۷ قسمتی (7-segment) است. این IC، ورودی کد باینری-دهدهی (BCD) را به خروجی‌های مربوطه برای نمایشگر ۷ قسمتی تبدیل می‌کند و امکان نمایش عددی بر روی LEDهای ۷ قسمتی را فراهم می‌کند.

گیت‌های منطقی:

۱. گیت‌های AND، OR، XOR (U23, U20)

- عملکرد: گیت‌های منطقی پایه برای انجام عملیات منطقی استفاده می‌شوند. گیت AND تنها زمانی خروجی بالا می‌دهد که تمام ورودی‌ها بالا باشند، گیت OR در صورتی که هر ورودی بالا باشد، خروجی ۱ می‌دهد و گیت XOR در صورتی که تعداد ورودی‌های ۱ فرد باشد، خروجی ۱ می‌دهد.

نمایشگرها:

۱. نمایشگرهای ۷ قسمتی (7 segment)

- عملکرد: این‌ها نمایشگرهای LED هستند که برای نمایش خروجی‌های عددی استفاده می‌شوند. مدار از نمایشگرهای ۷ قسمتی برای نمایش خروجی عملیات باینری یا شمارنده‌ها استفاده می‌کند.

کلیدها، مقاومت‌ها، و اتصالات

۱. کلیدها (مثل clk)

- عملکرد: این‌ها برای کنترل دستی سیگنال‌هایی مانند پالس‌های ساعت (clk) و ریست در مدار استفاده می‌شوند.

۲. مقاومت (R1)

- عملکرد: مقاومت‌ها جریان را در مدار محدود می‌کنند تا از محافظت از اجزا یا ایجاد افت ولتاژ استفاده کنند.

۳-۱ شرح آزمایش

۱-۳-۱ مدار کلی

در ابتدا با استفاده از تراشه 4040 که یک شمارنده هست، مقدار pc را هر بار یک واحد زیاد می‌کنیم. (دقت شود که از ۵ بیت این تراشه استفاده شده زیرا حافظه دستورات ما ۳۲ خانه دارد). سپس، آدرس ساخته شده توسط این شمارنده را به تراشه ۲۷۳۲ یا همان EPROM می‌دهیم که در نتیجه خروجی آن، دستور متناظر در آدرس داده شده است. دو بیت سمت چپ آن را کاری نداریم و دستورات ما ۶ بیتی هستند. با دو تراشه ۷۴۸۳ که جمع‌کننده ۴ بیتی هستند، حاصل هر دستور را محاسبه می‌کنیم.

ورودی اول این تراشه‌ها، رجیستر صفرم هستند. ورودی دوم آنها از طریق ۸ تا مالتی پلکسر با تراشه ۷۴۱۵۱ بدست می‌آید که سیگنال کنترلی آنها، ۳ بیت سمت راست دستور است. برای اینکه تعیین کنیم در کدام رجیستر باید عملیات write صورت گیرد، از یک دیکودر ۲ به ۴ استفاده کردیم که سیگنال ورودی آن، دو بیت وسط دستور (destination) هستند. در نهایت ۴ رجیستر با تراشه ۷۴۱۹۸ به عنوان رجیسترهای صفرم تا سوم داریم. از ۴ جفت seven segment برای نمایش مقادیر رجیسترها استفاده کردیم.

۱-۳-۲ حافظه دستورات

برای این قسمت، تنها تغییرات جزئی در مدار قسمت قبلی مورد نیاز است. ما فقط یک EPROM و یک شمارنده اضافه می‌کنیم، سپس خروجی EPROM را به ورودی مدار موجود متصل می‌کنیم.

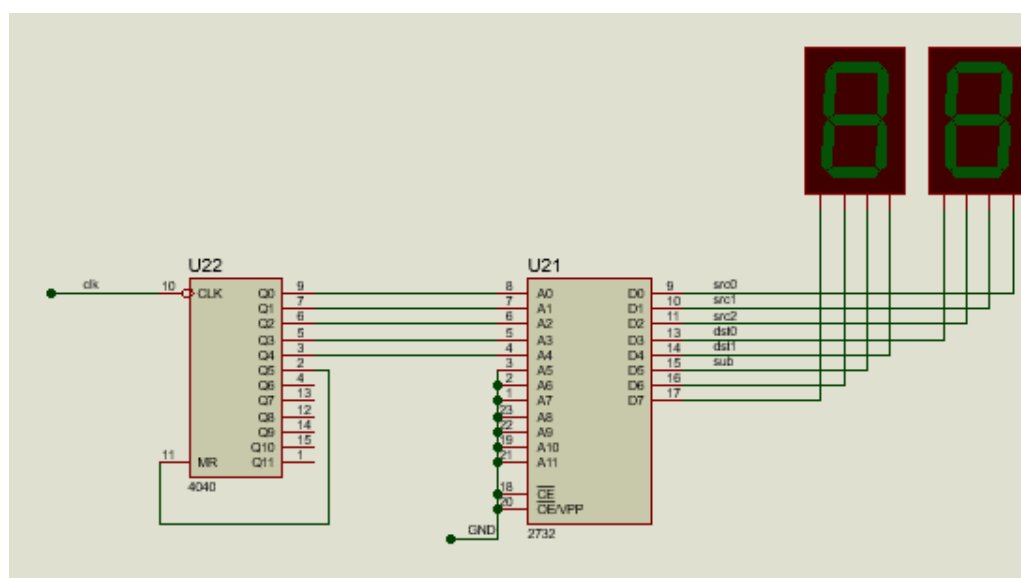


Figure ۲ حافظه دستورات

در ادامه دستورالعمل‌های مورد نیاز برای تولید ۶ جمله اول دنباله فیبوناچی را به دست میآوریم.

Memory Address	Binary Code	Hex Code	Assembly Instruction	Comment
00000	00 1 00 000	20	SUB R0, R0	R0 ← 0, جمله اول در R0
00001	00 0 01 101	0D	ADD R1, 1	R1 ← 1, جمله دوم در R1
00010	00 0 00 001	01	ADD R0, R1	R0 ← 1, جمله سوم در R0
00011	00 0 01 001	09	ADD R1, R0	R1 ← 2, جمله چهارم در R1
00100	00 0 00 001	01	ADD R0, R1	R0 ← 3, جمله پنجم در R0
00101	00 0 01 001	09	ADD R1, R0	R1 ← 5, جمله ششم در R1
00110	00 0 00 001	01	ADD R0, R1	R0 ← 8, جمله هفتم در R0
00111	00 0 01 001	09	ADD R1, R0	R1 ← 13, جمله هشتم در R1
01000	00 0 00 001	01	ADD R0, R1	R0 ← 21, جمله نهم در R0
01001	00 0 01 001	09	ADD R1, R0	R1 ← 34, جمله دهم در R1

Figure ۳ جدول دستورات

دستگاه برنامه‌ریزی (EPROM) که ما استفاده می‌کنیم نیاز به یک فایل هگز دارد تا آدرس‌های مناسب را پر کند. این فایل باید در فرمت هگز اینتل نوشته شود که در زیر به تفصیل شرح داده شده است.

فرمت Intel HEX :

فرمت Intel HEX یک فرمت فایل است که برای نمایش داده‌های باینری در یک فایل متنی استفاده می‌شود. این فرمت معمولاً برای برنامه‌ریزی میکروکنترلرها، EEPROMها و سایر دستگاه‌های قابل برنامه‌ریزی استفاده می‌شود. این فرمت به‌خصوص در سیستم‌های نهفته (embedded systems) برای ذخیره‌سازی فریمور یا داده‌هایی که باید در حافظه بارگذاری شوند، کاربرد دارد.

ساختار یک فایل Intel HEX شامل چندین خط از متن ASCII است که هر خط نشان‌دهنده‌ی یک رکورد داده می‌باشد. هر رکورد با یک علامت دو نقطه (:) شروع می‌شود و شامل فیلدهای زیر است:

کد شروع: یک علامت دو نقطه (:) که آغاز رکورد را مشخص می‌کند.

تعداد بایت‌ها: دو رقم هگزادسیمال که تعداد بایت‌های داده در رکورد را مشخص می‌کند.

آدرس: چهار رقم هگزادسیمال که آدرس حافظه شروع داده‌ها را نشان می‌دهد.

نوع رکورد: دو رقم هگزادسیمال که نوع رکورد را مشخص می‌کند. انواع رایج شامل: ۰۰: رکورد داده ۰۱:

رکورد پایان فایل ۰۲: رکورد آدرس سگمنت توسعه‌یافته ۰۴: رکورد آدرس خطی توسعه‌یافته

داده: یک فیلد با طول متغیر که شامل بایت‌های داده واقعی به صورت هگزادسیمال است.

مجموع کنترلی (Checksum): دو رقم هگزادسیمال که مجموع کنترلی رکورد را نشان می‌دهد.

ما محتویات زیر را در فایل هگز مربوط به EPROM نوشته ایم:

:0A000000200D0109010901090109A1

:00000001FF

در زیر به صورت خط به خط کد بالا را توضیح می‌دهیم.

خط اول:

:0A000000200D0109010901090109A1

این خط یک رکورد داده است و از چند بخش تشکیل شده است:

۱. کد شروع: (0A)

هر خط از فرمت Intel HEX با کاراکتر دو نقطه (:) شروع می‌شود.

۲. تعداد بایت‌ها: (0A)

دو کاراکتر بعدی تعداد بایت‌های داده در رکورد را نشان می‌دهد.

○ 0A در مبنای شانزده برابر با ۱۰ در مبنای ده است، بنابراین این رکورد شامل ۱۰ بایت داده است.

۳. آدرس: (0000)

چهار کاراکتر بعدی آدرس ۱۶ بیتی را نشان می‌دهد که داده باید در آن بارگذاری شود.

○ 0000 به این معنی است که داده‌ها از آدرس 0x0000 در حافظه بارگذاری می‌شوند.

۴. نوع رکورد: (00)

دو کاراکتر بعدی نوع رکورد را نشان می‌دهد.

○ 00 به معنای یک رکورد داده است که حاوی داده‌هایی برای بارگذاری در حافظه است.

۵. داده‌ها: (200D0109010901090109)

20 کاراکتر بعدی (۱۰ بایت) نشان دهنده داده‌هایی است که باید در حافظه بارگذاری شوند.

○ این بایت‌ها عبارتند از: 20 0D 01 09 01 09 01 09 01 09 01 09

۶. بررسی مجموع: (A1)

- دو کاراکتر آخر مجموع بررسی را نشان می‌دهد که برای تشخیص خطا استفاده می‌شود.
- مجموع بررسی به گونه‌ای محاسبه می‌شود که جمع تمام بایت‌های رکورد، از جمله مجموع بررسی، به عنوان یک مقدار ۸ بیتی (بدون توجه به سرریز) برابر با ۰ باشد.

خط دوم:

:00000001FF

این خط یک رکورد پایان فایل (EOF) است که نشان دهنده پایان فایل Intel HEX می‌باشد.

۱. کد شروع: (00)

همانند قبل.

۲. تعداد بایت‌ها: (00)

این مقدار نشان می‌دهد که هیچ بایت داده‌ای در این رکورد وجود ندارد.

۳. آدرس: (0000)

معمولاً برای رکورد EOF نادیده گرفته می‌شود.

۴. نوع رکورد: (01)

01 نشان دهنده یک رکورد پایان فایل (EOF) است.

۵. بررسی مجموع: (FF)

این مقدار مجموع بررسی برای رکورد EOF است که همیشه FF می‌باشد.

در پایان شکل نهایی مدار به قسم زیر می‌باشد:

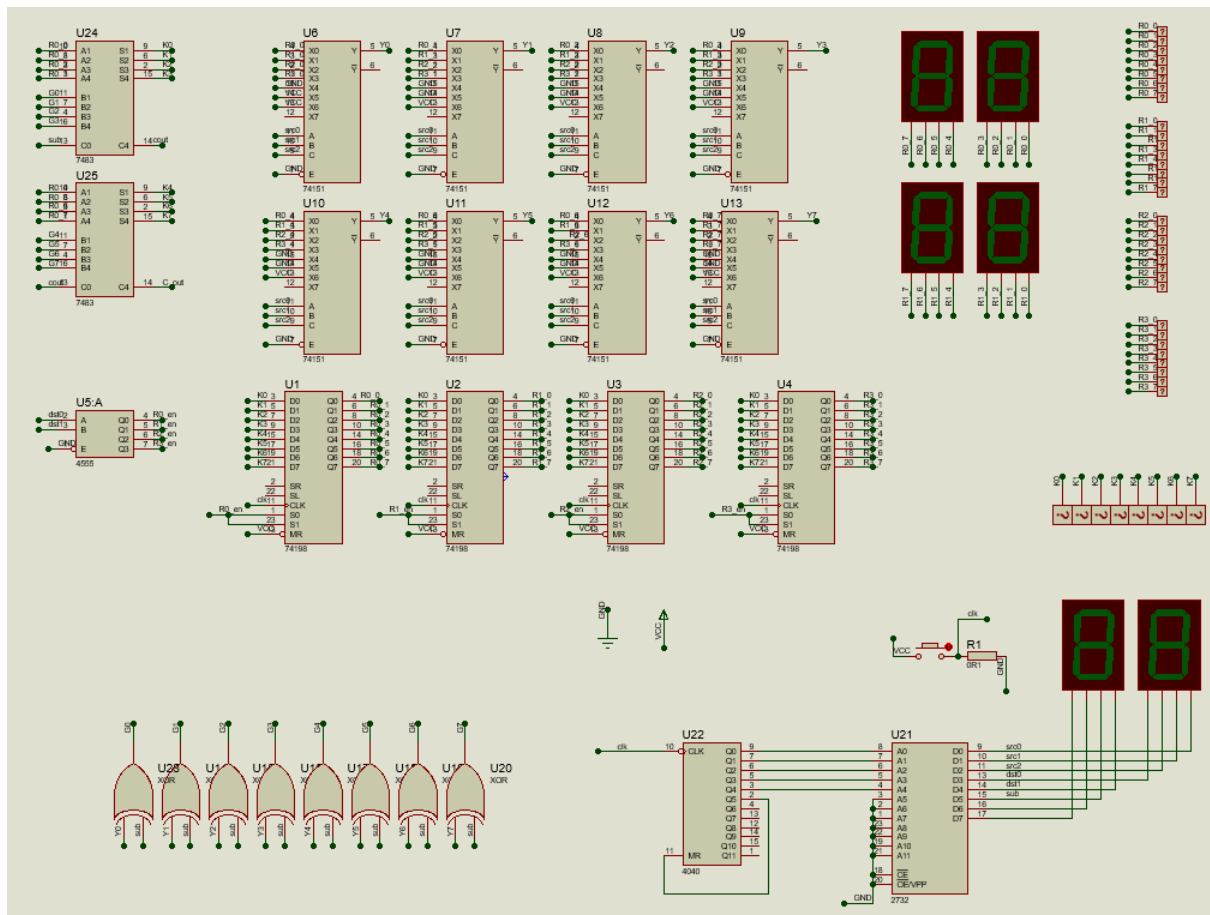


Figure ۴ شکل کلی مدار

۴-۱ منابع و مراجع

- وبسایت گیکز فور گیکز
- وبسایت یوتیوب
- وبسایت ویکی پدیا