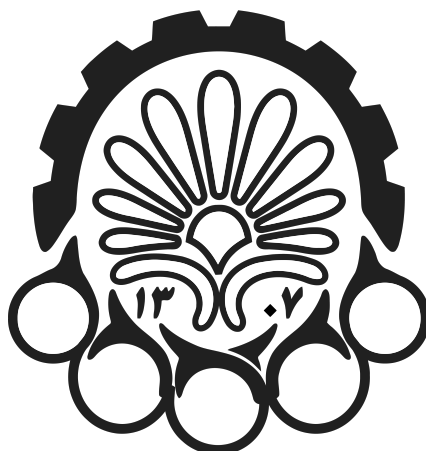


به نام خدا



دانشگاه صنعتی امیرکبیر
(پلی تکنیک تهران)
درس مدارهای منطقی

پروژه نهایی

دانشکده مهندسی کامپیوتر

دانشگاه صنعتی امیرکبیر

نیم سال دوم ۰۳ - ۰۲

استاد:

دکتر مرتضی صاحب الزمانی

مهلت ارسال و ارائه:

۱۷ تیر

مسئول پروژه:

مدرسین آزمایشگاه منطقی

طراحان پروژه:

رضا آدینه پور، محمد مهدی نعمتی

فهرست مطالب

۲	نکات قابل توجه
۳	مقدمه
۳	يعقوب برقي كه بود؟
۴	اهداف پروژه
۴	توضیحات کلی
۵	حالت مدیر
۷	حالت کاربر
۹	بخش های اصلی
۹	فاز اول
۹	راه اندازی مخزن GitHub
۱۱	طراحی FSM برای منطق پروژه
۱۱	فاز دوم
۱۲	بخش های امتیازی
۱۲	وضعیت دریافت ورودی
۱۳	وضعیت دریافت ورودی



نکات قابل توجه

- ددلاین تحویل فاز اول پروژه، در روز یکشنبه، ۱۳ خرداد در کلاس آزمایشگاه است.^۱ همچنین تحویل فاز دوم پروژه، یکشنبه ۱۷ تیر خواهد بود.
- انجام پروژه به صورت گروهی ست و گروه‌های شما همان گروه‌های آزمایشگاه است. تنها در موارد خاص، با هماهنگی با مدرسين آزمایشگاه می‌توانید گروه خود را تغییر دهید.
- دانشجویان ملزم به تحویل حضوری هر دو فاز پروژه هستند. در صورتی که گروهی در روز تحویل حاضر نشود نمره‌ای به ایشان تعلق نخواهد گرفت.
- در روز تحویل حضوری مشارکت تمام اعضای تیم در پروژه بررسی خواهد شد و در صورت عدم مشارکت بعضی از اعضا، نمره‌ای ایشان برای آن فاز از پروژه "صفر" لحاظ می‌گردد. مشارکت، با توجه به commit های افراد تیم در مخزن گیت‌هاب پروژه و سوالات پرسیده شده در روز تحویل و با صلاح دید مدرس مشخص می‌شود.
- ساختار مازولار کد ها حتما باید حفظ شود در صورت عدم رعایت، منفی نمره آن قسمت برای اعضای گروه ثبت خواهد شد.
- ددلاین فاز اول و دوم در آخرین زمان ممکن قرار داده شده است. لذا در هیچ یک از دو فاز مهلت تاخیر وجود ندارد.
- در صورت کشف تقلب از هریک از تیم‌ها، برای بار اول منفی نمره‌ای فاز برای آن تیم ثبت می‌شود و برای بار دوم، نمره‌ای منفی کل پروژه برای تیم لحاظ خواهد شد که معادل مردود شدن در درس است.

^۱ ساعت برگزاری کلاس توسط مدرس هر گروه اعلام می‌شود

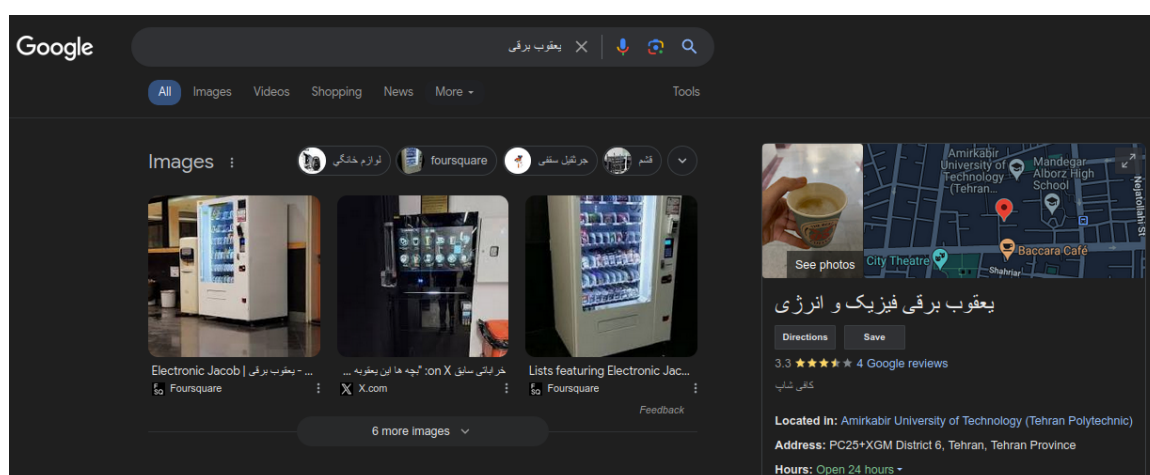


مقدمه

یعقوب^۲ برقی که بود؟

هرامیرکبیری میدونه!

که اسم جهانی یعقوب برقی^۳ مخصوص دستگاه‌های فروش خودکار است که اگر با همین نام اون رو در گوگل سرچ کنید متوجه جهانی بودن این اسم می‌شوید.



شکل ۱: این نام جهانیست!

احتمالا همگی یعقوب برقی دانشکده رو دیدین و از اون استفاده کردین. در این پروژه قصد داریم کمی بیشتر با یعقوب برقی آشنا بشیم و بخش کوچکی از سیستم کنترلی اون رو طراحی، شبیه‌سازی و پیاده‌سازی کنیم. بعد از انجام این پروژه، خریدهای شما از یعقوب برقی، فقط یک خرید ساده نیست!

^۲Electronic Jacob

^۳برگرفته از یعقوب آقا، دکه وسط صحن دانشگاه



اهداف پروژه

- هدف این پروژه، طراحی و شبیه‌سازی سیستم کنترلی دستگاه فروش خودکار است که در فاز اول آن صرفاً FSM های کلی پروژه طراحی و مخزن^۴ گیت‌هاب پروژه راه اندازی می‌شود.
- در فاز دوم، می‌بایست FSM های کشیده شده در فاز اول تکمیل شوند، طوری که تمام حالات پوشش داده شود و همچنین کد طراحی انجام شده به زبان Verilog نوشته شود و بر روی FPGA پیاده‌سازی شود.
- آشنایی با سیستم مدیریت نسخه Git و کار تیمی بر روی پروژه بر بستر یک مخزن Github، یکی دیگر از اهداف پروژه است. در این مورد توصیه می‌شود تغییرات خود را در دوره‌های کوتاه مدت commit کنید.

توضیحات کلی

پروژه در ۲ حالت^۵ کاری طراحی شده است.

- حالت مدیر^۶
- حالت کاربر^۷

بنابر این پروژه باید دارای ۳ ماژول اصلی `main.v` و `user.v`، `admin.v` باشد. زیر ماژول های هر کدام در ادامه توضیح داده خواهد شد.

Repository^۴
Mode^۵
Admin^۶
User^۷



حالت مدیر

سیستم پس از روشن شدن به صورت پیش فرض در حالت کاربر روشن می شود. با تغییر حالت یک بیت از Dip Switch های موجود بر روی بُرد وارد حالت مدیر می شویم. هر دستگاه پسوردی دارد که تنها مدیر دستگاه با وارد کردن آن می تواند وارد این حالت شود. پسورد دستگاه به صورت زیر در نظر گرفته شود:

Admin_pass: 02

مقدار پسورد باید جایی در حافظه رجیستر شود و در زمان عوض کردن حالت کاری سیستم، پسورد وارد شده توسط شخص با مقدار از قبل ذخیره شده مقایسه شود. شخص می تواند برای وارد کردن پسورد از Dip switch های بُرد استفاده کند. پس از وارد شدن به حالت مدیر، مدیر می تواند اجناس دستگاه را مطابق با شماره خانه های یعقوب برقی دانشکده (شکل «۲») شارژ کند



شکل ۲: یعقوب برقی دانشکده



همچنین برای انتخاب جایگاه مورد نظر از Dip switch ها استفاده کنید و برای مشخص کردن تعداد اجناس قرار داده شده در جایگاه انتخاب شده از ۲ Push button بُرد استفاده کنید. یکی از Push button ها بالا شمار و دیگری پایین شمار تنظیم شود. و در نهایت برای تایید تعداد جنس مورد نظر از یک بیت از Dip switch های باقی مانده استفاده شود. همچنین می بایست تعداد اجناس و شماره جایگاه انتخاب شده توسط مدیر، به صورت پویا^۸ بر روی Seven segment های بُرد در هنگام قرار دادن اجناس نمایش داده شود. برای راحتی فرض های زیر انجام شود:

- موجودی تمامی جایگاه ها در ابتدای کار صفر است
- هر جایگاه حداکثر ۹ ظرفیت دارد
- قیمت اجناس ردیف اول^۹ ۸۵۰۰ تومان
- ردیف دوم، ۶۰۰۰ تومان
- ردیف سوم ۴۵۰۰
- ردیف چهارم ۳۰۰۰ تومان
- و ردیف ۵ و ۶ هر کدام ۲۰۰۰ تومان

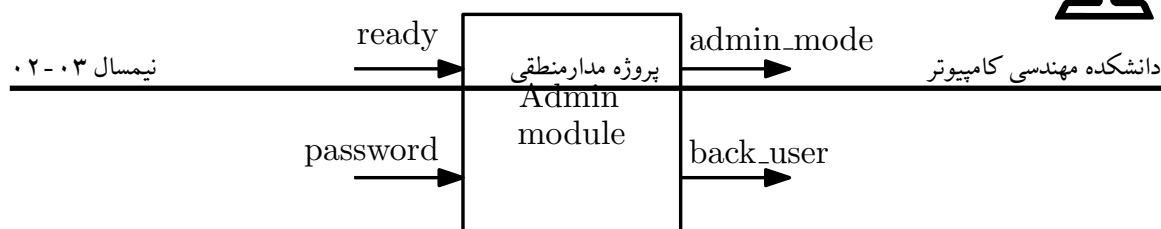
برای خروج از حالت مدیر می بایست بیت در نظر گرفته شده برای ورود به حالت مدیر را تغییر حالت دهید. در صورت فراموشی مدیر برای خروج از حالت مدیر، برای حفظ امنیت، دستگاه می بایست به صورت خودکار پس از ۱ دقیقه از حالت مدیر خارج شده و به حالت کاربر برگردد.

نکته مهم: قبل از خروج می بایست تمامی تغییرات صورت گرفته ذخیره شود.

ماژول طراحی شده در این قسمت باید به صورت زیر باشد:

(خروجی های admin_mode و back_user حالت کاری سیستم را نشان می دهند که باید به دو LED متصل شوند.)

^۸Dynamic^۹ترتیب ردیف ها از بالا بیان شده



شکل ۳: ماژول admin.v

نکته مهم: Push button های موجود بر روی بُرد، debounce نشده است. بنابراین نیاز این موضوع برای عملکرد صحیح سیستم در نظر گرفته شود^{۱۰}

حالت کاربر

در این حالت، کاربران می‌توانند شماره جنس مورد نظر خود را انتخاب کنند و در صورت موجود بودن کالا و داشتن موجودی مالی، آن را خریداری کنند. برای سادگی فرض می‌شود ۳ کاربر داریم که مقدار موجودی حساب هر کدام به صورت زیر است:

- کاربر ۱: ۷۰۰۰ تومان
- کاربر ۲: ۵۰۰۰ تومان
- کاربر ۳: ۳۰۰۰ تومان

انتخاب کاربرها با استفاده از Dip switch ها انجام می‌شود و فعال بودن هر کدام می‌بایست بر روی LED ها نمایش داده شود.

پس از انتخاب کاربر، موجودی آن باید بر روی Seven segment نمایش داده شود. پس از انتخاب کالای مورد نظر و تایید خرید (با استفاده از Push button) در صورت داشتن موجودی کافی در حساب کاربر و دستگاه، خرید با موفقیت انجام می‌شود و مانده حساب آپدیت شده و نمایش داده می‌شود.

موجودی هر کالا پس از انتخاب آن توسط کاربر می‌بایست بر روی LED ها نمایش داده شود.

^{۱۰} برای اطلاعات بیشتر می‌توانید اینجا را بخوانید

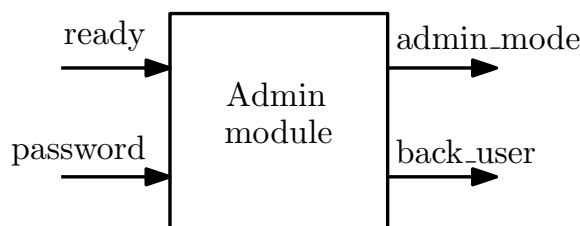


۱. هر کاربر می تواند چندین بار خرید انجام دهد.
۲. در صورت موجود نبودن کالا می بایست پیغام مناسب داده شود و فرایند خرید پایان یابد
۳. در صورت نداشتن موجودی حساب، کاربر اصلا نباید وارد فرایند خرید شود



۴. پس از اتمام خرید، موقعیت جایگاه دستگاه بر روی حالت قبلی می ماند اما تا تایید نکردن خرید نباید فرآیند خرید آغاز شود

ماژول طراحی شده در این قسمت باید به صورت زیر باشد:



شکل ۴: ماژول `user.v`

بخش های اصلی

فاز اول (۱۰ + ۱۰) نمره

در این فاز شما باید مقدمات پروژه را حاضر کنید. این مقدمات شامل ابزارهای مورد استفاده در پروژه و همچنین طراحی FSM های پروژه به صورت کلی است. مراحل این فاز عبارت اند از:

۱. راه اندازی مخزن **GitHub**

۲. طراحی FSM برای منطق پروژه

در بخش بعد، هر یک از این موارد شرح داده شده اند.

راه اندازی مخزن **GitHub** (۱۰) نمره

همانطور که می دانید برای پروژه لازم است با گروهتان بر روی یک مخزن (repository) گیت فعالیت کنید. برای ساختن این مخزن، کافیست وارد [این لینک](#) شوید.



ابتدا با لیستی مواجه می شوید که شماره دانشجویی تمام افراد در آن موجود است. شماره
دانشکده مهندسی کامپیوتر و بر روی آن کلیک کنید.
نیمسال ۰۲-۰۳



در صفحه‌ی بعد شما باید تیم خود را انتخاب کنید. چنانچه نفر اول گروه خود (سازنده‌ی مخزن) هستید، باید یک تیم بسازید. تنها شماره‌ی گروه پروژه خود را در قسمت نام تیم وارد کنید و تیم را بسازید. نفرات بعدی گروه شما، باید تیم‌شان را از لیست تیم‌های موجود انتخاب کنند و نیازی به ایجاد تیم ندارند.

پس از این مراحل مخزن شما آماده خواهد شد و لینک آن در اختیارتان قرار خواهد گرفت. پس از آماده شدن این مخزن، هر یک از اعضای پروژه باید نام و شماره دانشجویی خود را به فایل README.md اضافه کند.

طراحی FSM برای منطق پروژه (۱۰) نمره

همانطور که در بخش توضیحات کلی مطرح شد، پروژه دارای ۳ ماژول اصلی است. در این قسمت از شما می‌خواهیم ۳ FSM منطبق بر منطق ۳ ماژول `admin.v`، `user.v` و `main.v` رسم نمایید.

در این فاز نیازی به در نظر گرفتن جزئیات مسئله نمی‌باشد و صرفاً طراحی کلیات این ۳ ماژول کفایت می‌کند.

فاز دوم (۸۰) نمره

پس از طراحی FSM های کلی در فاز اول و راه‌اندازی مخزن گیت‌هاب، در این فاز از پروژه نیاز است که جزئیات نیز در نظر گرفته شود و FSM های طراحی شده تکمیل شود. پس از تکمیل FSM ها نیاز است کدهای verilog ماژول های توضیح داده شده در بخش توضیحات کلی نوشته شود.

تمامی ماژول ها باید قابل سنتز و پیاده سازی^{۱۱} باشند چرا که می‌بایست به صورت عملی بر روی بُرد پروگرام شده و تست شود.

اکیدا توصیه می‌شود مرحله به مرحله کدها را نوشته و برای هر ماژول Testbench بنویسید و از صحت عملکرد ماژول مطمئن شوید، اما اجباری به نوشتن Testbench برای تک تک ماژول‌ها نیست ولی برای کل سیستم لازم است Testbench بنویسید.

^{۱۱} Implementation



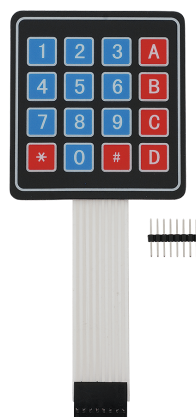
بخش‌های امتیازی (۱۵ + ۳۰) نمره

از این قسمت به بعد وارد بخش‌های امتیازی پروژه می‌شویم. دقت کنید که این بخش‌های هم بسیار آموزنده و کاربردی هستند و هم می‌تواند به نمره پروژه شما به اندازه قابل توجهی اضافه کند. (:

وضعیت دریافت ورودی (۱۵) نمره

در حالت اجباری و غیر امتیازی پروژه، تمامی ورودی‌ها با استفاده از Dip switch ها و Push button ها داده می‌شد. در این بخش می‌توانید به صورتی طراحی را تغییر دهید که سیستم شما ورودی‌ها را با استفاده از صفحه کلید^{۱۲} دریافت کند.

بُرد های آزمایشگاه مجهز به صفحه کلید نمی‌باشند اما می‌توانید آن را به صورت مجزا از مسؤل آزمایشگاه تهیه کرده و ماژولی به نام keypad.v نوشته و آن را به بُرد متصل کنید. نمونه ای از صفحه کلید های رایج در ادامه آورده شده است.



شکل ۵: صفحه کلید

^{۱۲}Keypad



وضعیت نمایش (۳۰) نمره

در این قسمت می‌توانید برای نمایش وضعیت موجودی اجناس، قیمت‌ها، شماره کاربر و مانده حساب از LCD کاراکتری استفاده نمایید. بُرد های آزمایشگاه مجهز به LCD کاراکتری هستند.

برای انجام این قسمت می‌بایست مازولی به نام lcd.v بنویسید. در ادامه نمونه‌ای از LCD های رایج آورده شده است.



شکل ۶: LCD کاراکتری