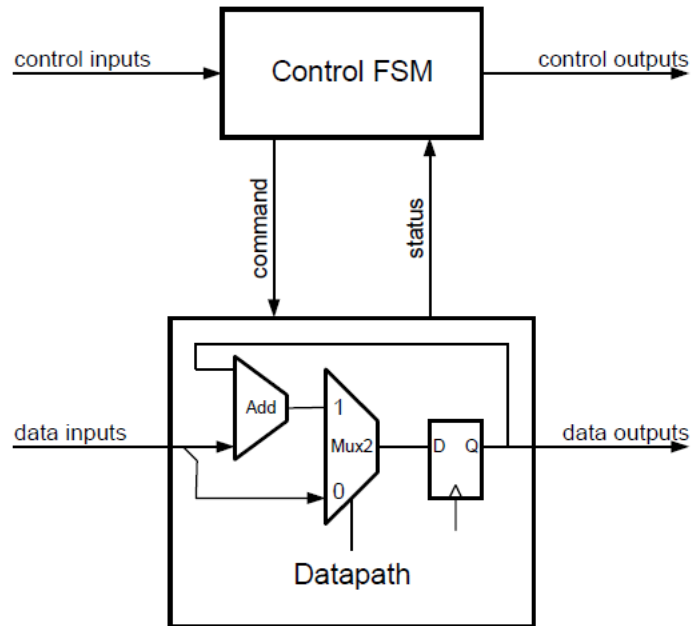


گزارش پروژه مدار منطقی ( Verilog )

زهرا مرادی – 9832077

## مقدمه

هر سیستم دیجیتال معمولاً به دو بخش تقسیم می شود. شکل 1 بلوک دیاگرام یک سیستم را نشان می دهد. بخش datapath منطق حالت بعدی و خروجی های مدار را تولید می کند. بخش کنترل حالت بعدی را بر اساس اساس حالت فعلی و ورودی های مدار تولید می کند.

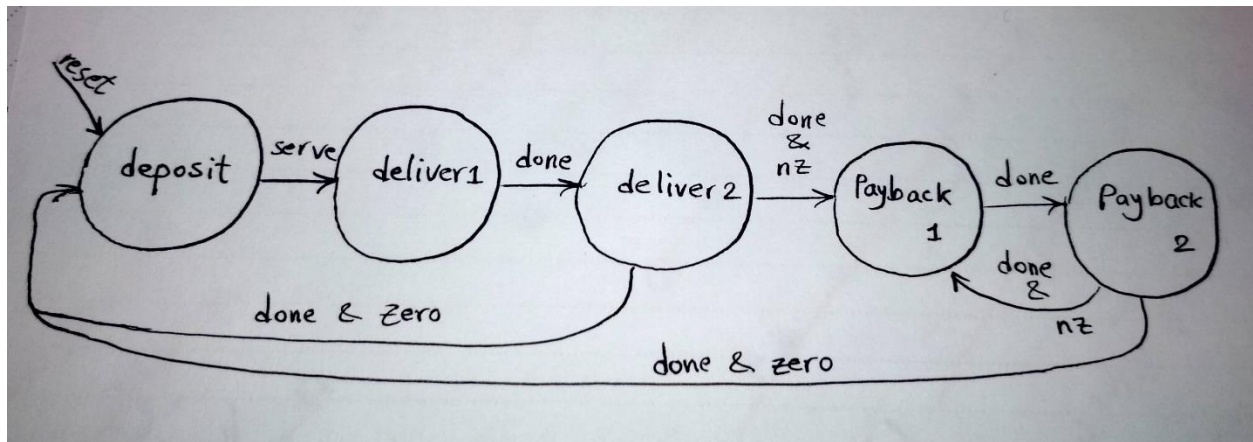


شکل 1: مثالی از یک سیستم متشکل از Datapath و Control

در نظر داریم ماشینی برای تحویل اتوماتیک پنج نوع نوشیدنی طراحی کنیم. برای سادگی فرض می‌کنیم که این ماشین تنها یک نوع سکه را می‌تواند دریافت کند. به هنگام قرار دادن یک سکه در دستگاه پالسی [coin] به طول یک پالس ساعت به عنوان ورودی دستگاه ارسال می‌شود. فرض می‌کنیم قیمت هر نوشیدنی یک عدد چهار بیتی است که در دستگاه بصورت دیپ سوئیچ تنظیم شده است. موقع راه‌اندازی این مقادیر خوانده شده، در آرایه‌ای از رجیسترهای چهار بیتی به نام price قرار می‌گیرد. تعداد هر کدام از نوشیدنی‌ها نیز در آرایه‌ای از رجیسترهای چهاربیتی به نام stock در هنگام راه‌اندازی سیستم مقداردهی می‌شود. بر فرض که این دستگاه پنج تا دکمه برای انتخاب نوشیدنی دارد که با یک سیگنال پنج بیتی one-hot به نام select به عنوان ورودی داده می‌شود. این ماشین سه مود عملیاتی دارد که با مودهای تحویل نوشیدنی (mode=00)، مود شارژ کالا (mode=01)، مود برداشت پول (mode=10) که با فشار دکمه‌ای بین مودهای مختلف جابجا می‌شود. فرض می‌کنیم این دستگاه خروجی نیز دارد که قیمت کالا و تعداد و پول ذخیره شده را به همراه چراغ LED برای نمایش خطا مناسب با مود انتخاب شده نمایش می‌دهد.

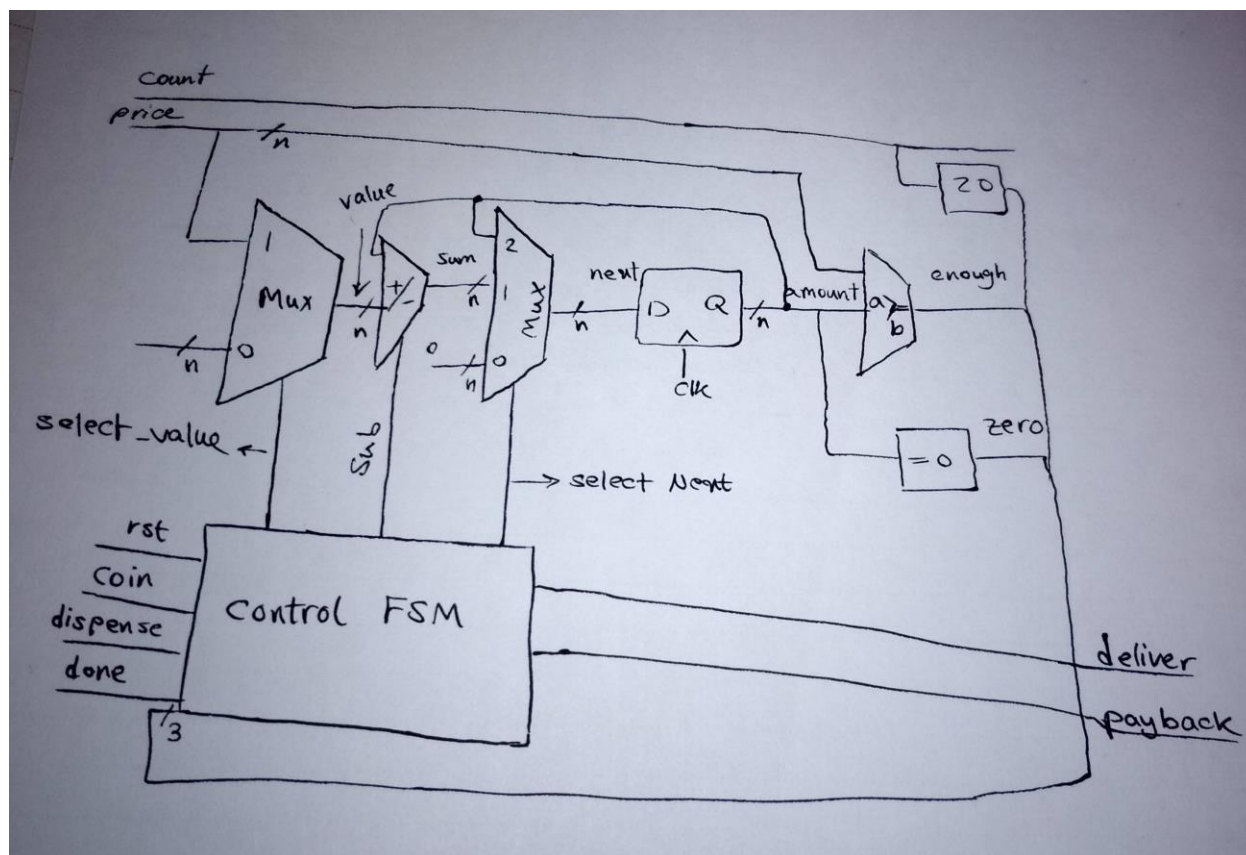
مود تحویل نوشیدنی: برای پشتیبانی این مود ماژول Unit\_Drink\_Dispenser طراحی شده است. مشتری نوشیدنی را انتخاب کرده و تعدادی سکه در دستگاه قرار می‌دهد برای نمایش کافی بودن تعداد سکه سیگنالی بنام enough وجود دارد. با فشار دکمه dispense توسط مشتری سیگنالی به همین نام به طول یک پالس ساعت فعال می‌شود. اگر پول مشتری به اندازه کافی باشد با تولیدی سیگنالی بنام deliver به طول یک پالس ساعت فرمان تحویل نوشیدنی ارسال می‌شود. مکانیسم تحویل دهنده نوشیدنی با ترکیب سیگنال‌های select و serve مشخص می‌کند که کدام نوشیدنی تحویل داده شود. بعد از تکمیل عملیات تحویل، سیگنالی بنام done توسط مکانیسم تحویل دهنده تولید می‌شود و به دستگاه امکان می‌دهد که وارد مرحله بازپرداخت بقیه پول مشتری شود. برای این کار سیگنالی به طول یک پالس ساعت بنام payback تولید می‌شود. مکانیسم تحویل بعد از بازپرداخت هر سکه سیگنال done را فعال می‌کند. در یک حلقه تکرار شونده دستگاه تمامی سکه‌های باقی مانده مشتری را یکی یکی تحویل می‌دهد.

این واحد خود از دو ماژول کنترل و مسیرداده تشکیل شده است. واحد کنترل یک FSM پنج حالتی است. که در شکل 1 نشان داده شده است. برای سادگی حلقه‌های بازخوردی به خود حالت‌ها نشان داده نشده است.



شکل 1: ماشین حالت دستگاه واحد تحویل نوشیدنی

دستگاه پس از ریست وارد حالت deposit می شود در این حالت با آمدن سیگنال coin به مقدار پول مشتری اضافه می شود. بعد از فعال شدن dispense و enough وارد حالت deliver1 می شود. در اولین سیکل که وارد این حالت می شویم، سیگنال deliver فعال می شود. سیگنالی به نام first مشخص می کند آیا برای بار اولی است که وارد این حالت شدیم یا نه. ماشین در این حالت می ماند تا با آمدن سیگنال done وارد حالت deliver2 شود. در حالت deliver2 منتظر پایین رفتن سیگنال done می شود. با پایین رفتن سیگنال done در این حالت اگر مشتری پول باقی مانده ای داشته باشد دستگاه سیگنال payback را تولید می کند و تا آمدن سیگنال done در این حالت می ماند با یک شدن سیگنال done ماشین به حالت payback2 می رود و منتظر پایین رفتن سیگنال done می شود. پایین رفتن این سیگنال مشخص می کند که آیا سکه تحویل شده است. حالا اگر باقی مانده برابر صفر نباشد باز وارد حالت payback1 می شویم و این حلقه تا زمان تحویل آخرین سکه ادامه پیدا می کند.



شکل 2: مسیر داده لازم برای واحد تحویل نوشیدنی

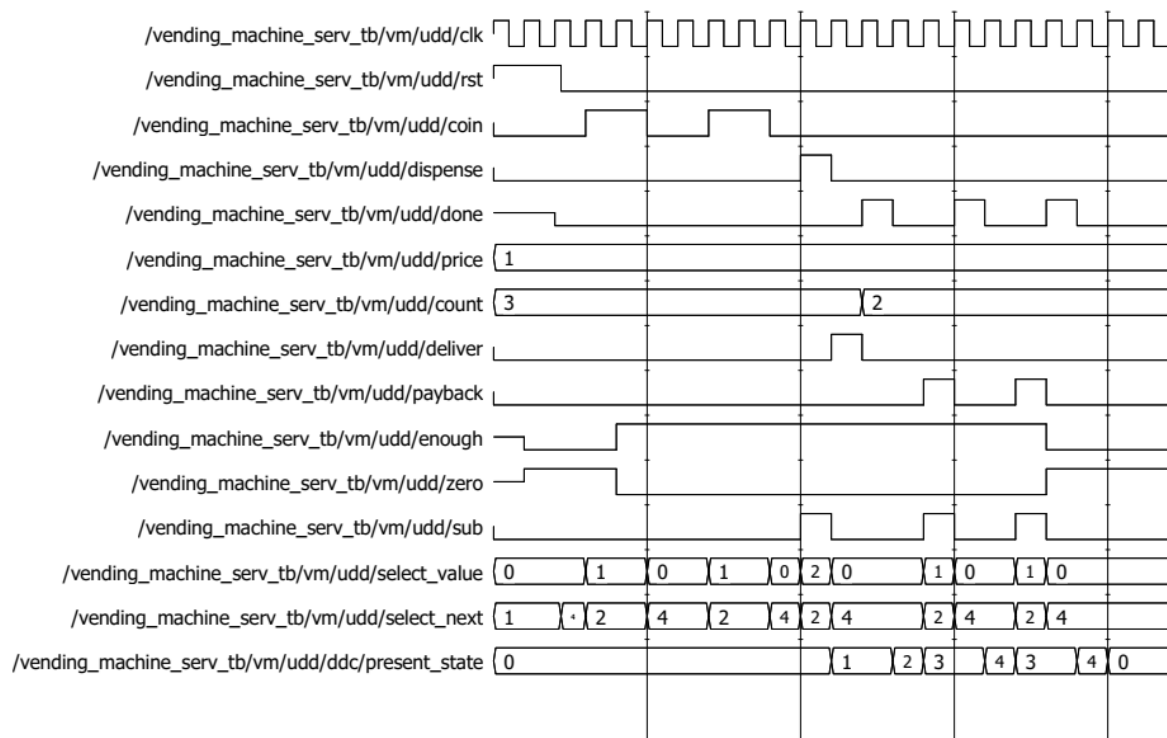
شکل 2 مسیر داده لازم را برای این واحد نشان می دهد. سیگنال های مهم کنترلی لازم عبارتند از `select_value`: سیگنال انتخاب مالتی پلکسر مشخص کننده مقدار ورودی به مالتی پلکسر است. اگر ورودی `coin` باشد یا حالت `payback` برای تولید مقدار بعدی با عدد 1 جمع یا تفریق می کنیم و اگر حالت `deliver`1 باید مقدار `price` را به ورودی جمع کننده/تفریق کننده بدهیم تا مقدار باقی مانده پول مشتری به دست بیاید.

select\_next: مشخص کننده مقدار بعدی که باید در مقدار amount یا پول مشتری ثبت شود

sub: مشخص می کند آیا باید عمل جمع یا تفریق. در حالت Deposit عمل جمع و در حالت deliver1 و payback1 عمل تفریق انجام میشود.

تشخیص اینکه آیا در سیکل اولی هستیم که وارد حالت deliver1 شدیم یا نه از طریق یک فلیپ فلاپ انجام می شود ورودی فلیپ فلاپ عبارت  $\{(present\_state == deliver1)\}$  که نشان می دهد حالت قبلی وضعیتی غیر از deliver1 بوده. به روش مشابه برای تشخیص سیکل اول payback نیز همین سیگنال به کار رفته است.

نتایج حاصل از شبیه سازی در نرم افزار ModelSim در شکل 3 نشان داده شده است.

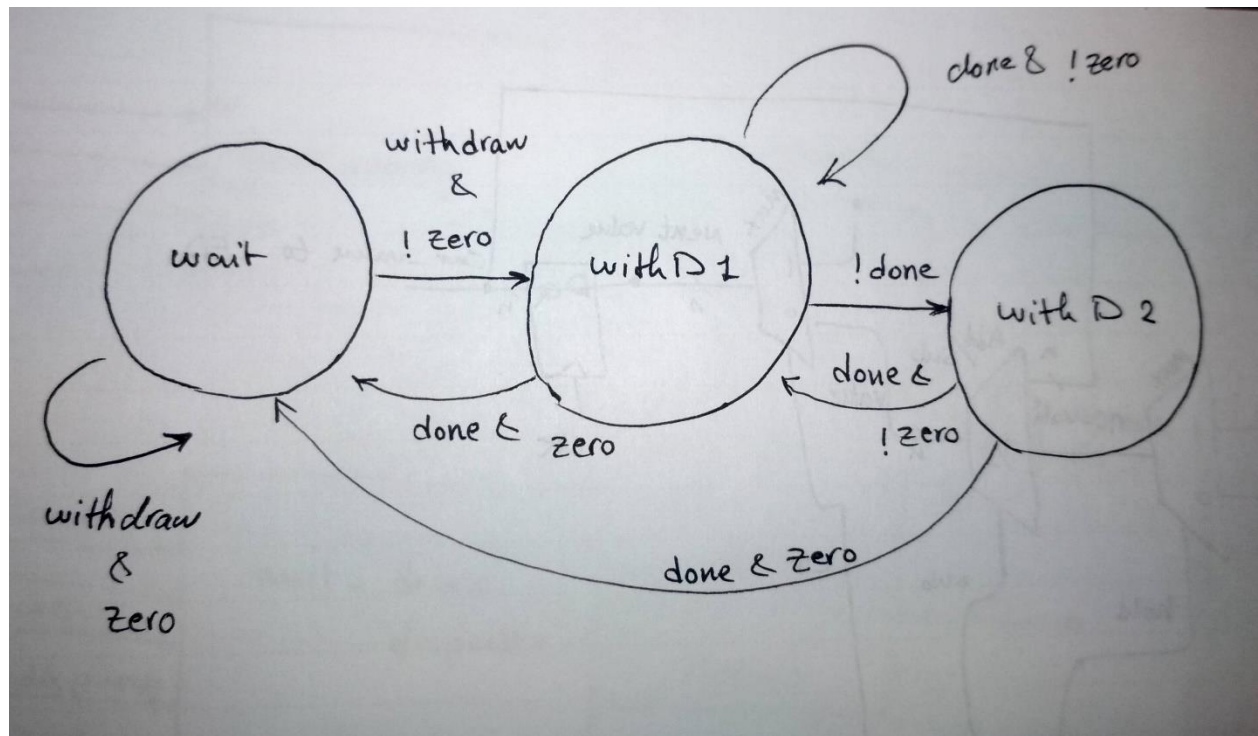


شکل 3: نتایج شبیه سازی Drink\_Dispenser\_Unit

مشاهده می کنیم مشتری تعداد چهار عدد سکه را در دستگاه قرار داده است سپس با انتخاب نوشیدنی اول و فشار دکمه dispense یکبار سیگنال deliver به نشانه تحویل نوشیدنی فعال شده است و بقیه پول مشتری نیز در طی سه سیکل بازپرداخت شده است. بعد از تحویل نوشیدنی سیگنالی بنام discharge تولید و به واحد افزایش تعداد نوشیدنی فرستاده می شود تا از مقدار نوشیدنی های موجود یکی کم شود.

مود برداشت پول: واحد unit\_money\_withdraw برای پشتیبانی از این مود طراحی شده است. این واحد دارای ثباتی به نام balance است که مقدار پول ذخیره شده در دستگاه را نشان می دهد. دستگاه با فشار دکمه Withdraw توسط تولید سیگنال payback فرمان پرداخت سکه را می دهد. همچنین سیگنال هایی به نام inc و dec اضافه شده است. فعال شدن سیگنال inc باعث می شود که یک واحد به مقدار موجودی balance اضافه شود. این سیگنال موقع اضافه کردن سکه توسط مشتری فعال می شود. هنگام بازپرداخت سکه به مشتری نیز فعال شدن سیگنال dec باعث کاهش پول ذخیره شده در دستگاه به میزان یک واحد است.

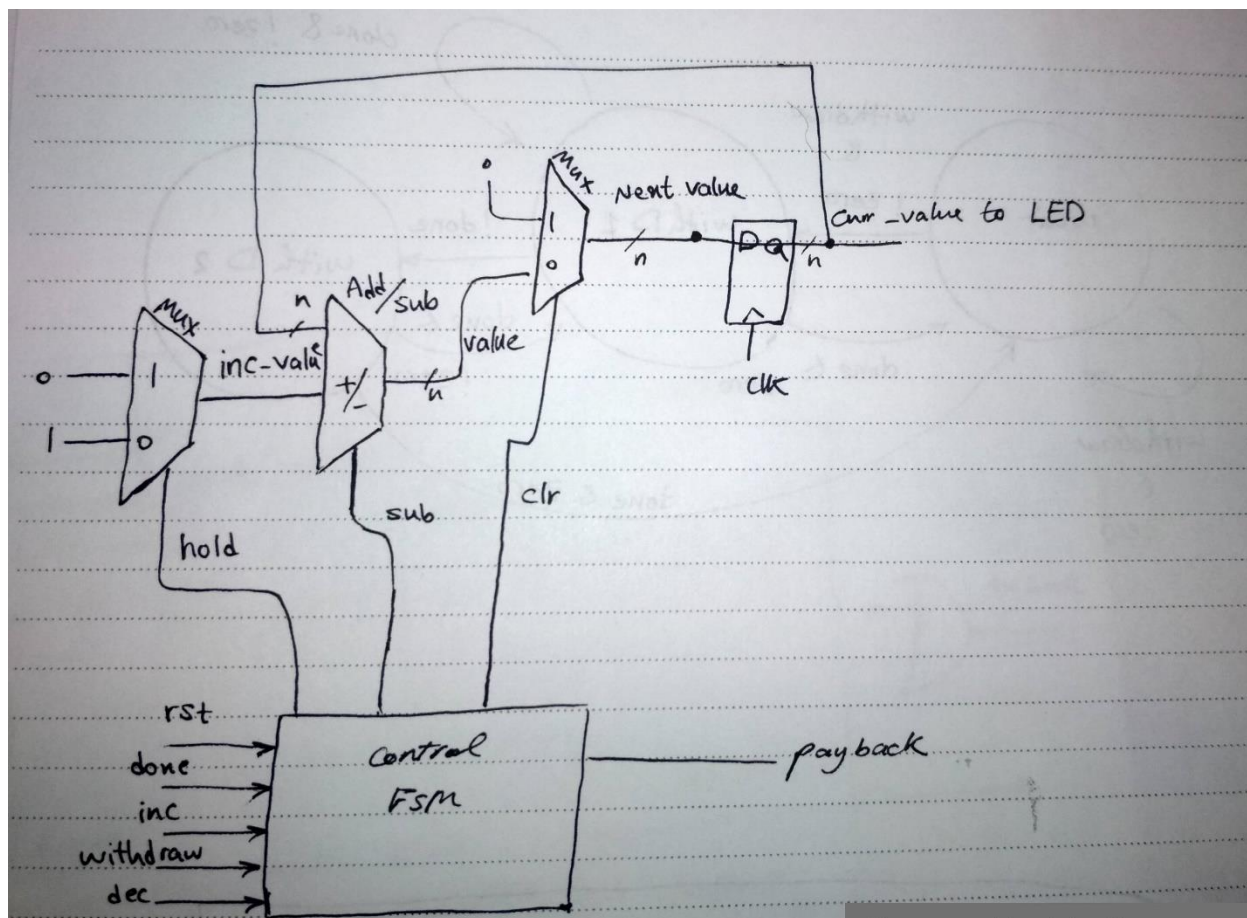
ماشین حالت این واحد دارای سه حالت است. با سوییچ به این مود همیشه وارد حالت WAIT می شویم و منتظر فشرده شدن دکمه برداشت می شود. فشردن دکمه برداشت سیگنال withdraw را فعال می کند و ماشین را به حالت WITHD1 می برد. در حالت WAIT یک سیگنال به نام payback برای ارسال فرمان پرداخت سکه تولید می شود و از مقدار موجود سکه ها در بانک دستگاه یک واحد کم می شود و ماشین تا زمان پاسخ مکانیسم تحویل سکه به حالت WITHD2 می رود با آمدن سیگنال done و موجود بودن سکه های دیگر در دستگاه که با سیگنال zero! مشخص می شود، مجدد وارد حالت WITHD1 می شود و اجرای این حلقه باعث می شود تا آخرین سکه تحویل صاحب دستگاه شود. با صفر شدن موجودی ماشین به حالت WAIT برمی گردد که فشار دادن دکمه withdraw در این حالت تاثیری نخواهد داشت. شکل 4 دیاگرام حالت این ماژول را نشان می دهد.



شکل 4: دیاگرام حالت ماژول برداشت سکه

مسیر داده این ماژول شامل متغیر *balance* که از تعدادی فلیپ فلاپ *D* برای نگهداری تعداد سکه‌های ذخیره شده، و یک جمع کننده تفریق کننده شش بیتی برای انجام تغییرات در متغیر *balance* و تعدادی مالتی پلکسر است. مالتی پلکسر اول مقدار 0 و 1 را تولید می کند که مقدار 1 برای کاهش یا افزایش یک واحد به عنوان عملوند جمع/تفریق کننده وارد می شود و مقدار 0 برای زمانی است که نباید تغییری در مقدار *balance* داده 0 شود.

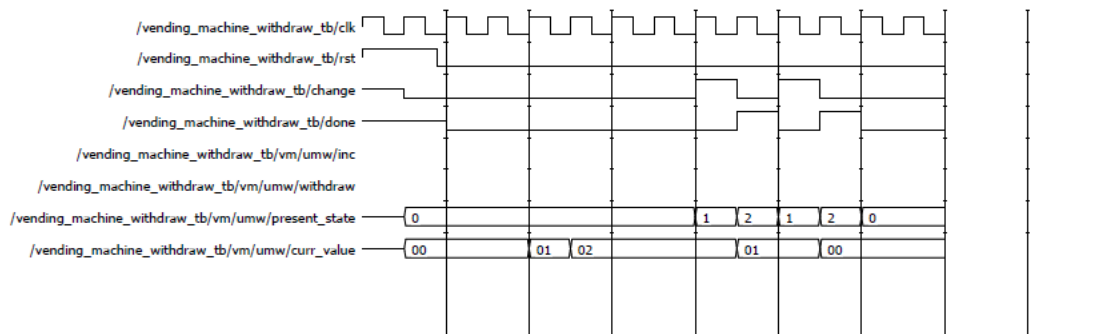




شکل 5: datapath مازول برداشت پول

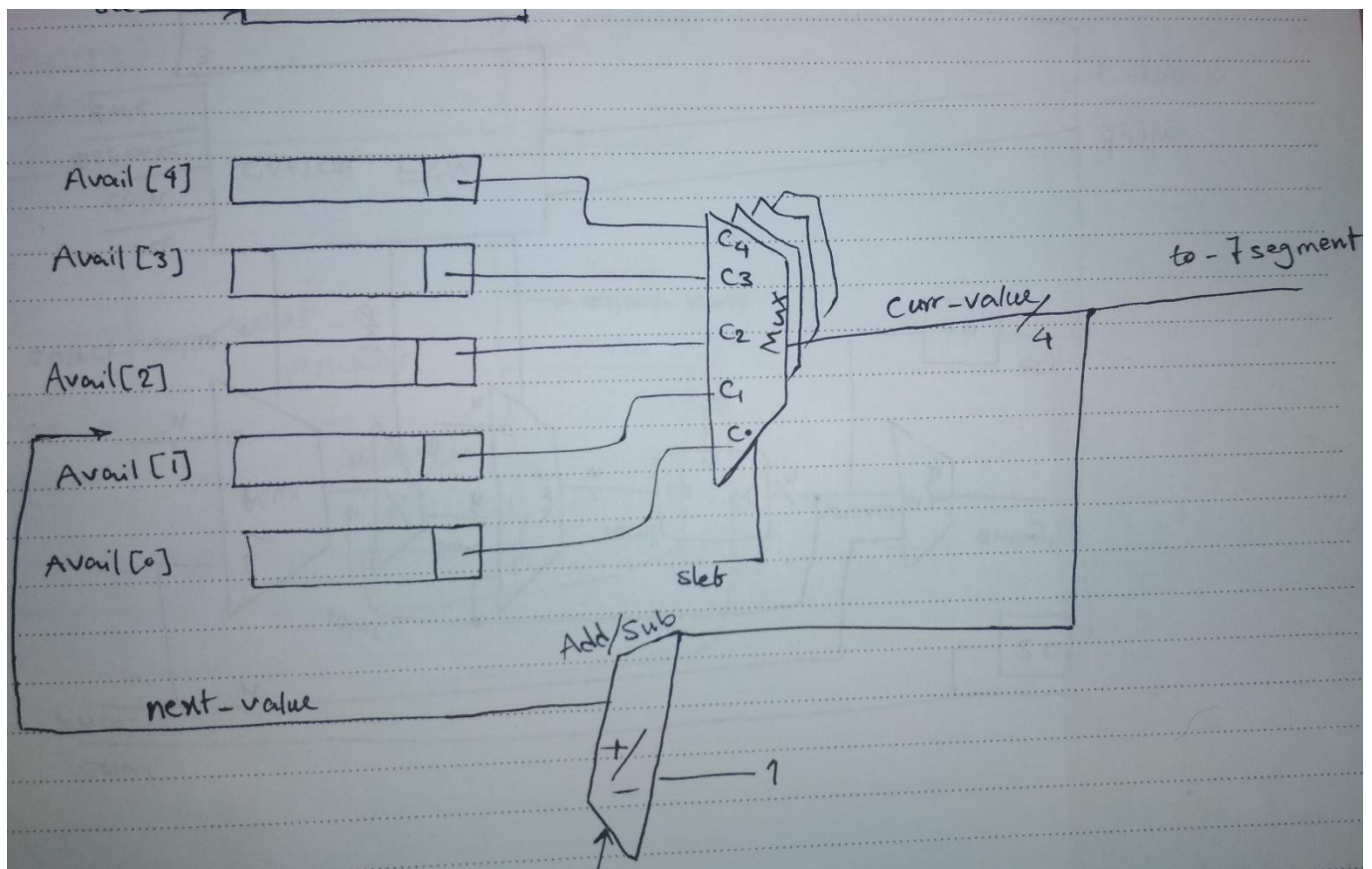
سیگنال clr باعث می شود که مقدار 0 به متغیر balance انتساب داده شود. سیگنال hold نشان می دهد که در حالت withd1 نیستیم یا اینکه هیچگدام از سیگنال های Inc یا dec فعال نیست و نباید مقدار balance تغییر داده شود. برای راحتی مقدار balance شش بیت در نظر گرفته ایم و به دلیل اینکه تعداد کالا و قیمت هر کدام چندین بیت دارد باید تعداد بیشتری بیت در نظر می گرفتیم در این حالت می توان برای سادگی مدار از جمع کننده سریال استفاده نمود.

شکل 6 نتایج شبیه سازی این واحد را نشان می دهد. با توجه به اینکه در ابتدای اجرا مقدار پول موجود در دستگاه صفر است، به مدت چندین پالس ساعت سیگنال inc را فعال کرده ایم تا پول موجود در دستگاه افزایش پیدا کند. خروجی curr\_value به led سیستم متصل است و مقدار سکه های موجود را در این مود نمایش می دهد. مشاهده می شود که تنها با یکبار فشار دادن دکمه مورد نظر سیستم در یک حلقه تمامی سکه ها را تحویل می دهد برای شبیه سازی پاسخ مکانیسم تحویل بعد از فعال شدن payback سیگنال done را به مدت یک پالس ساعت فعال کرده ایم.



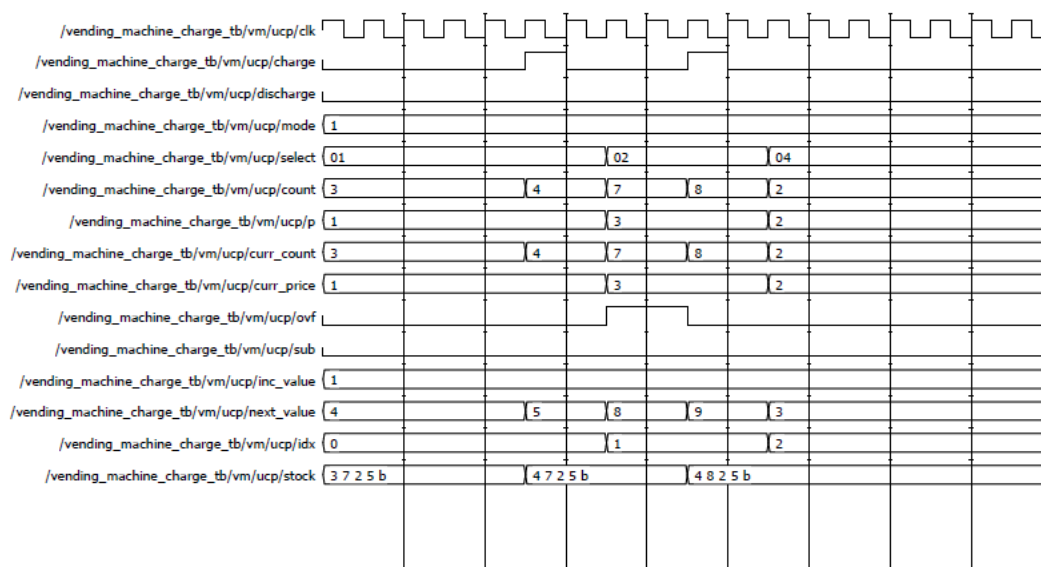
شکل 6: نتایج خروجی واحد برداشت پول

مود افزایش موجودی کالا: واحد Unic\_charge\_product برای این ماژول طراحی شده است. دو سری متشکل از 5 ثبات چهار بیتی چهاربیتی هر کدام نگهدارنده تعداد کالا ها (stock) و قیمت کالا (price) است. سیگنال select به عنوان ورودی One-hot تبدیل به یک اندیس باینری شده است که معادل یک Encoder است. در این مود فشار دادن دکمه charge با موجودی فعلی کالایی که توسط مالتی پلکسر با سیگنال select انتخاب شده یکی اضافه میشود و در لبه مثبت ساعت در ثبات تعداد مرتبط نوشته می شود و به این ماژول یک ورودی به نام discharge اضافه شده است که موقع تحویل کالا به مشتری باعث می شود یک واحد از مقدار کالای مورد نظر کم شود. مسیر داده این واحد در شکل 6 نشان داده شده است که شامل تعدادی مدار ترکیبی برای تولید تعداد صحیح کالا است. یک ردیف از فلیپ فلاپها نیز قیمت را حفظ می کند و به عنوان ورودی برای واحد تحویل کالا عمل می کند.



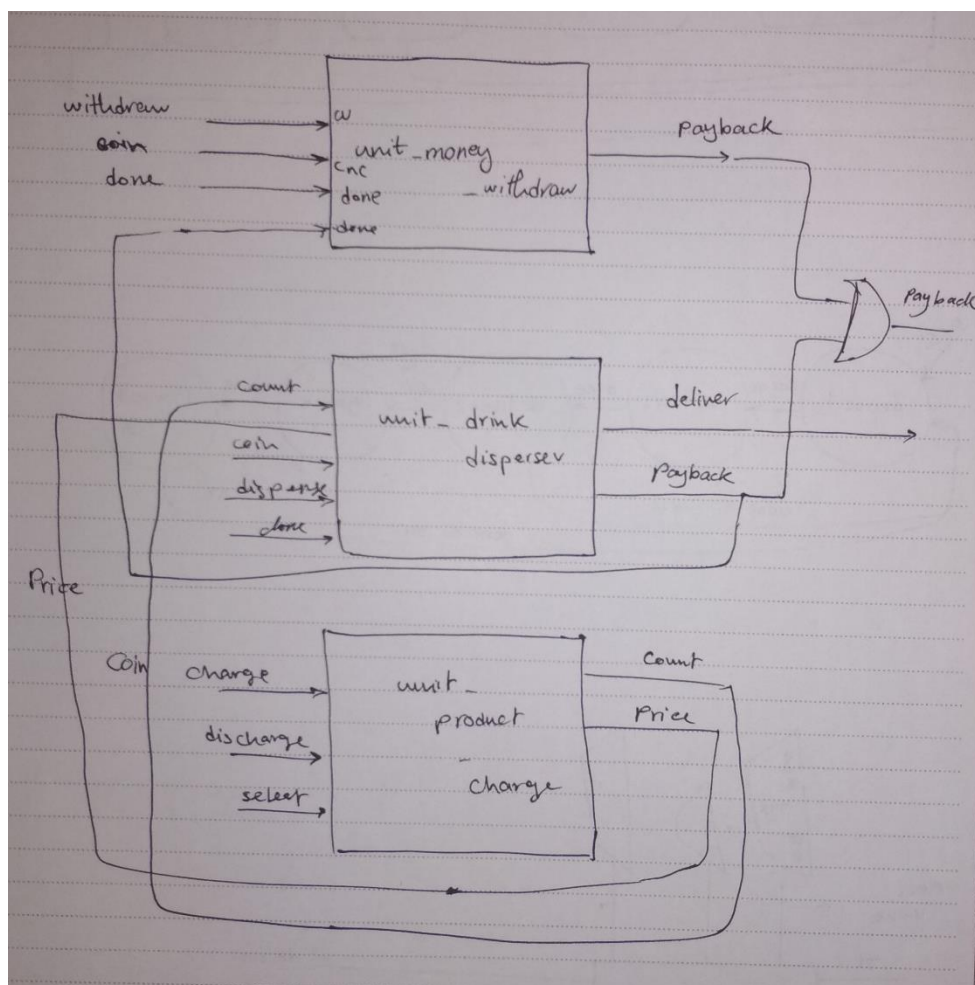
شکل 6: datapath ماژول برداشت افزایش موجودی

نتایج تست این واحد در شکل 7 نشان داده شده است. مشاهده می شود بعد از اجرای مقدار دهی اولیه از فایل stuff.txt دو کالا انتخاب شده است و با فشار دکمه charge به موجودی کالای متناظر یکی اضافه شده است.



شکل 7: نتایج شبیه سازی واحد افزایش کالا

اتصال ماژول ها: در شکل 8 بلوک دیاگرام و سیکنا مهم ارتباطی سه واحد نشان داده است. سیگنال های خروجی payback از دو واحد با همدگر or شده و به خروجی متصل است. سیگنال coin به عنوان ورودی ر دو واحد پرداخت کالا و برداشت وجه است. خروجی deliver از واحد پرداخت کالا به عنوان ورودی discharge واحد کالا وصل شده تا علاوه بر تحویل کالا از موجودی تعداد کالاها یکی کم کند. همچنین خروجی payback واحد مشتری به عنوان ورودی dec واحد تحویل پول است تا موقع بازپرداخت سکه به مشتری از موجودی بانک سکه ها یکی کم کند.



شکل 8: بلوک دیاگرام vending machine و اتصال واحد ها

برای هر ماژول تست جداگانه ای نوشته شده است و در ابتدای نوع کالا و قیمت و مقدار هر کدام را از فایلی به نام stuff خوانده و در مقادیر price و stock بارگذاری کرده ایم. نتایج شبیه سازی با نرم افزار ModelSim نشان می دهد که مدار صحیح کار می کند که در شکل های ضمیمه نشان داده شده است.