طراحی کنترل کننده‌ی آسانسور به کمک پیاده‌سازی ماشین حالت در زبان VHDL

پارسا حجابی

چکیده

در این نوشتار نحوه طراحی یک مدار کنترل کننده‌ی آسانسور که دارای ۸ کلید ورودی اصلی، ۴ ورودی سنسور و یک ورودی ریست غیرهمزمان می‌باشد به کمک ماشین حالت بررسی می‌شود.

کلمات کلیدی

Structural، Behavioral، Dataflow، Elevator Control، State Machine، Moore، Mealy

۱- مقدمه

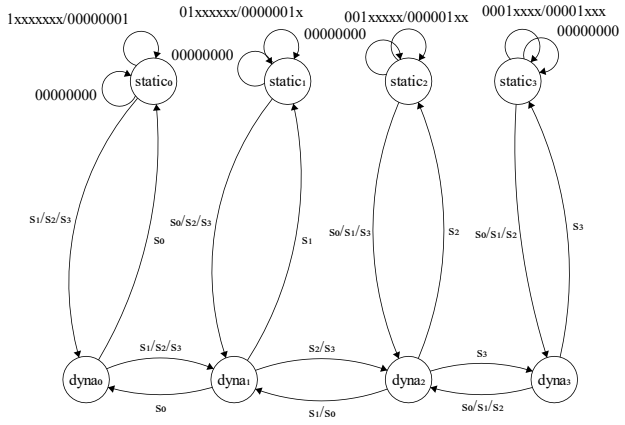
برای پیاده‌سازی بسیاری از مدارهای ترتیبی که وضعیت آن‌ها را می‌توان حالت بندی کرد و برای هر حالت مدار باید عملکرد خاصی داشته باشد از ماشین حالت استفاده می‌شود.

ماشین حالت دو نوع دارد: ماشین حالت میلی و ماشین حالت مور.

در ماشین حالت میلی خروجی در هر لحظه به حالت آن لحظه و ورودی آن لحظه بستگی دارد ولی در ماشین حالت مور خروجی در هر لحظه فقط به وضعیت آن لحظه بستگی دارد.

در این تمرین یک مدار کنترل کننده آسانسور را که وظیفه فرستادن فرمان به موتور آسانسور و کنترل آن را بر عهده دارد و دارای ۸ سیگنال کنترلی اصلی به همراه ۴ ورودی سنسور می‌باشد به وسیله ماشین حالت طراحی و پیاده‌سازی می‌کنیم.

مدل استفاده شده برای پیاده‌سازی این مدار به این صورت است که ۴ حالت برای سکون‌های آسانسور در ۴ طبقه و ۴ حالت برای آماده به حرکت بودن آسانسور در نظر گرفته شده است که ماشین حالت زیر را دارا می‌باشد:



شکل (۱): ماشین حالت با ۸ حالت برای مدار کنترل کننده‌ی آسانسور

۲- مطالب اصلی

۲-۱- طراحی ماشین حالت

همانطور که گفته شد، یک ماشین حالت با ۸ وضعیت اصلی داریم. ۴ وضعیت با نام‌های static0 تا static3 مربوط به حالت‌های سکون اتاقک آسانسور در ۴ طبقه و ۴ وضعیت دیگر با نام‌های dyna0 تا dyna3 مربوط به حالت‌های آماده به حرکت و در حال حرکت اتاقک آسانسور در ۴ طبقه می‌باشد.

برای پیاده‌سازی اولویت‌های گفته شده در سوال ۸ سیگنال مربوط به ۸ کلید در داخل و خارج آسانسور را به صورت یک متغیر ۸ بیتی با نام status و به صورت and کردن ۸ بیت به ترتیب زیر ایجاد می‌کنیم:

C0 & C1 & C2 & C3 & PB3 & PB2 & PB1 & PB0

برای حالت‌های static در هر طبقه در صورتی که هیچ درخواستی در داخل و خارج آسانسور نشود یا درخواست همان طبقه شود آسانسور در همان وضعیت یعنی در وضعیت static همان طبقه می‌ماند.

قرارداد ما برای پیاده‌سازی به این صورت است که در وضعیت‌های static اگر نیازی به حرکت اتاقک به طبقه‌های دیگر بود ابتدا تمام سیگنال‌های خروجی را در همان وضعیت static مقداردهی می‌کنیم و در حالات dynamic اتاقک آسانسور فقط با چک کردن سنسورهای موجود به وضعیت‌های dynamic دیگر می‌رود.

برای مثال اگر از طبقه همکف بخواهیم به طبقه دوم برویم ابتدا کسی باید در بیرون و یا در داخل آسانسور دکمه طبقه دوم را زده باشد. پس status ما در بیت C2 یا در بیت PB2 دچار تغییر خواهد شد. ما در وضعیت static0 هستیم بنابراین با خواندن status درب آسانسور را می‌بندیم و موتور آسانسور را برای بالا کشیدن تنظیم می‌کنیم. یک متغیر flag تعریف می‌کنیم که در صورتی که ۱ باشد نشان‌دهنده آزاد بودن آسانسور برای پاسخ به درخواست‌ها است و در صورتی که ۰ باشد نشان‌دهنده در حال انجام وظیفه بودن آسانسور می‌باشد. پس از تنظیم موتور آسانسور به وضعیت dyna0 می‌رویم. در صورتی که سنسور طبقه همکف خاموش شود به وضعیت dyna1 می‌رویم و باقی‌کار را به این وضعیت می‌سپاریم و به همین ترتیب جلو می‌رویم.

۲-۲- پیاده‌سازی

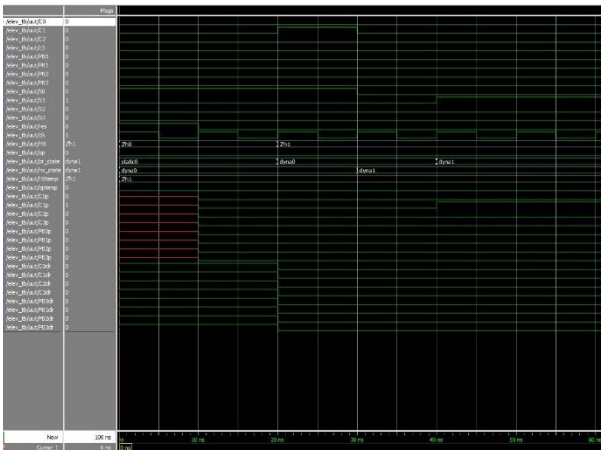
برای پیاده‌سازی از Design Style شماره دوم ماشین‌های حالت بهره می‌بریم. در این پیاده‌سازی ۳ Process مجزا داریم که یکی وظیفه چک کردن سیگنال‌های Rst و Clk را دارد و در آن وضعیت فعلی را مشخص می‌کنیم. در دومی که به حالت فعلی و دوازده سیگنال اصلی ورودی بالا حساس است، به ازای وضعیت فعلی و سیگنال‌های ورودی وضعیت آینده و خروجی‌ها را مشخص می‌کنیم و در سومی که باز هم به Rst و Clk حساس می‌باشد مقادیر خروجی را روی سیگنال‌های خروجی قرار می‌دهیم.

در صورتی که سیگنال Rst یک شده باشد ما باید وضعیت فعلی را به static0 یا وضعیت شروع تغییر دهیم و در غیر این صورت وضعیت فعلی را مقدار دهی می‌کنیم.

در Process بعدی هم با یک Case به ازای حالت‌های مختلف وضعیت فعلی و با چک کردن سیگنال‌های ورودی در هر حالت مقادیر مشخصی را به خروجی‌ها می‌دهیم و وضعیت بعدی را مشخص می‌کنیم.

۲-۳- تست

کلاک را هر ۵ نانوثانیه not می‌کنیم و بنابراین طول هر کلاک ۱۰ نانوثانیه می‌شود. در Process اصلی برنامه ابتدا Rst را یک می‌کنیم و تا ۱۰ نانوثانیه صبر می‌کنیم. سپس این سیگنال را صفر می‌کنیم و دکمه طبقه اول را از درون آسانسور می‌فشاریم. سپس ۱۰ نانوثانیه صبر می‌کنیم و دستمان را از این دکمه برمی‌داریم و سنسور طبقه همکف را صفر می‌کنیم و در نهایت سنسور طبقه اول را یک می‌کنیم.



شکل (۲): شکل موج تست گفته شده.

۳- نتیجه

برای پیاده‌سازی بعضی مدارهایی که حالت‌های زیادی برای بررسی دارند و همچنین ترتیبی هستند استفاده از ماشین حالت هم پیاده‌سازی را آسان می‌کند و هم بهینه‌ترین حالت پیاده‌سازی برای مدار می‌باشد.

مراجع

[1]