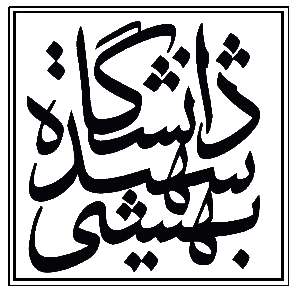
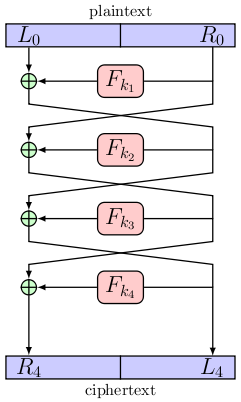
**درس طراحی کامپیوتری سیستم­های دیجیتال**

**گزارش شماره­ی: 16**

**اعضای گروه: نرگس قاسمی ، تارا برقیان**

**چکیده:** در این آزمایش، هدف پیاده سازی یک قطعه سخت افزاری بود که به حالت فیستل یک ورودی را با کلید مشخص رمز کرده و خروجی میدهد.

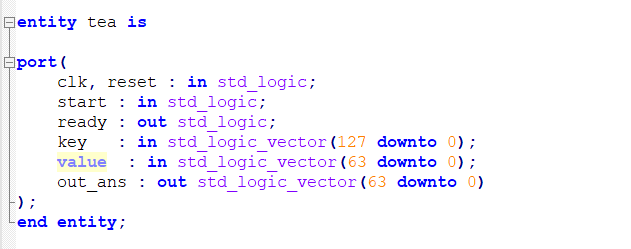
چالش اصلی پیاده سازی زیر کلید ها بود که در ادامه

به توضیح آن میپردازیم.

فهرست مطالب

|  |
| --- |
| 1. **ورودی و خروجی ها** |
| 1. **بدنه کد** |
| 1. **تست و نتیجه گیری** |

1. **ورودی و خروجی ها**

در این بخش علاوه بر سیگنال های تک بیتی کلاک، ریست، start یک کلید 128 بیتی و ورودی 64 بیتی داریم که قرار است رمز شود.

1. **بدنه کد**

کد از 2 process اصلی تشکیل شده است.

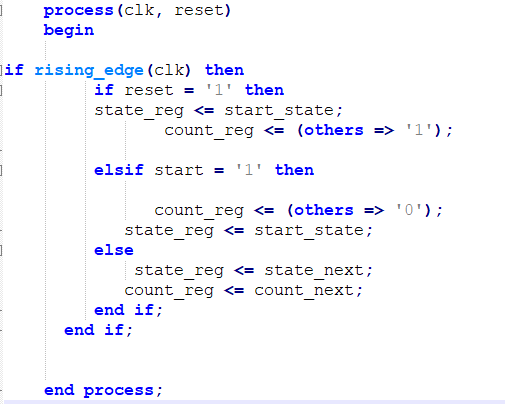
پراسس اول وظیفه ی چک کردن تغییرات کلاک و ریست و چک کردن سیگنال start را دارد.

در صورت تغییر ریست و تغییر کلاک:

اگر لبه بالا رونده باشد و ریست 1 باشد مقدار شمارنده 31 میشود و در حالت استیت start قرار میگیرد.

اگر لبه بالا رونده باشد و start 1 باشد، مقدار شمارنده 0 میشود و در حالت استیت start قرار میگرد و

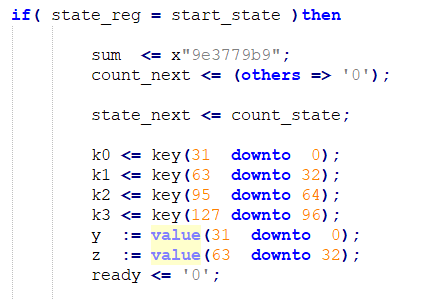
در غیر این صورت اگر اگر لبه بالا رونده باشد مقدار next\_state را قرار میدهد و مقدار شمارنده یکی زیاد میشود.



در قسمت بعدی که y و z به هم وابسته هستند باید variable باشند که حالت ترتیبی داشته باشیم.( طبق قوانین variable ها )



در این پراسس به تغییر شمارنده و استیت فعلی حساسیم. توجه شود که دقیق تر است اول کلاک زده شود و شمارنده اضافه شود، سپس تغییرات دیگر اعمال شوند که دقیقا همین روند اتفاق میفتد. ( علاوه بر جواب درست شبیه سازی صحیح هم دارد. )



اگر سیگنال استارت فعال بود، مقدار دهی های

اولیه یعنی تقسیم کلید به 4 بخش،

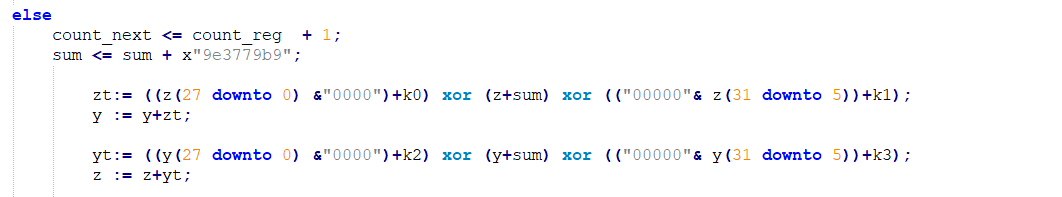
و مقدار دادن دو طرف چپ و راست کلید به

z,y صورت میگیرد.

و سیگنال ready در ابتدا 0 است.

(پس از پایان حلقه 1 میشود.)

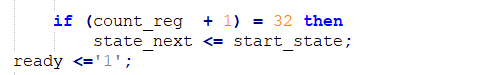
اگر در حالت استارت نباشیم، یعنی در یکی از مرحله های رمز کردن هستیم، که عملیات طبق کد c++ که داده شده بود در نظر گرفته شده است. دقت شود که تمامی سیگنال های z,zt,y,yt به صورت variable باید تعریف میشد.



در همین پراسس، تعداد حلقه های چرخیده شده چک میشوند تا از 32 کمتر باشد.

اگر 32 بار چرخیده بودیم فیسل کامل اجرا شده و جواب اماده است. همچنین مجدد قرار است به استیت استارت برگردیم پس،

next\_state با start\_state مقدار دهی میشود.



اگر هنوز 32 بار کامل نشده باشد، ready همچنان 0 خواهد بود و به شمارنده یکی اضافه میشود.



در پایان فیسل، جواب از کانکت z و yحاصل میشود :

1. **تست و نتیجه گیری**

ما 32 بار پس از 1 کردن استارت کلاک بالارونده اجرا کردیم و جواب به دست امده را با جواب کد c++ مقایسه کردیم که یکسان بود :

