

ازمایش 3

process ها و چگونگی پیاده سازی شمارنده ها و ماشینهای حالت

روژینا کاشفی-هلیا سادات هاشمی پور

تاریخ: 1400/1/18

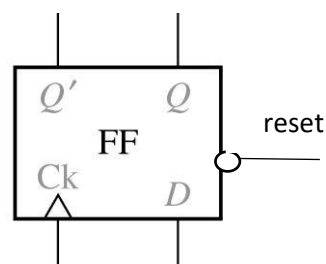
DFF:

همانطور که می دانید DFF یک حافظه برای نگهداری اطلاعات است و یک مدار ترتیبی است و در زمانی که کلاک در لبه بالا رونده است می توان رفتار آن را با وجود reset (active low) اسنکرون به صورت جدول زیر خلاصه کرد:

Reset	D	Q	Q+
0	X	0	0
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	1

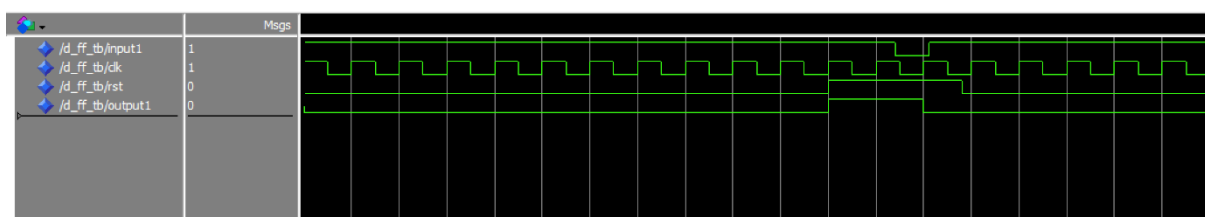
دقت شود reset اکتو لو است؛ بنابراین وقتی 0 است به معنای آن است که فعال است و زمانی که reset فعال باشد خروجی صرف نظر از آنکه ورودی چیست 0 می شود.

در حالت های دیگر مشاهده می کنیم که هر چی D باشد در لبه بالا رونده کلاک به خروجی منتقل می شود.



(a) Rising-edge trigger

خروجی کد نوشته شده:



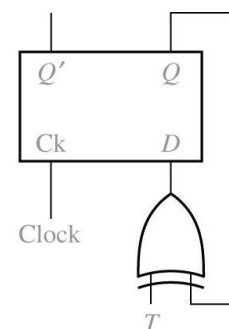
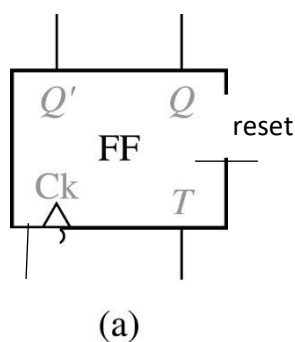
- همانطور که مشاهده می کنید زمانی که $reset = 0$ است صرف نظر از اینکه ورودی یا همان D چه هست خروجی 0 باقی می ماند.
- زمانی که $reset = 1$ است و غیرفعال در لبه بالا رونده کلاک ورودی را گرفته و در خروجی آن را ظاهر می کند تا در لبه بالا رونده کلاک بعدی ورودی دیگری بگیرد.
- در لبه بالا رونده بعدی ورودی همان 1 است بنابراین تا لبه بالا رونده بعدی 1 باقی می ماند.
- در لبه بالا رونده بعدی ورودی 0 است بنابراین بلافاصله خروجی صفر می شود.
- در لبه بالا رونده بعدی $reset$ فعال می شود و خروجی دوباره صفر می شود.

T_FF

همانطور که می دانیم T_FF یک نوع فلیپ فلاپ دیگر است که از آن به عنوان حافظه استفاده می شود.

رفتار TFF با وجود $reset$ استکرون به صورت زیر خواهد بود:

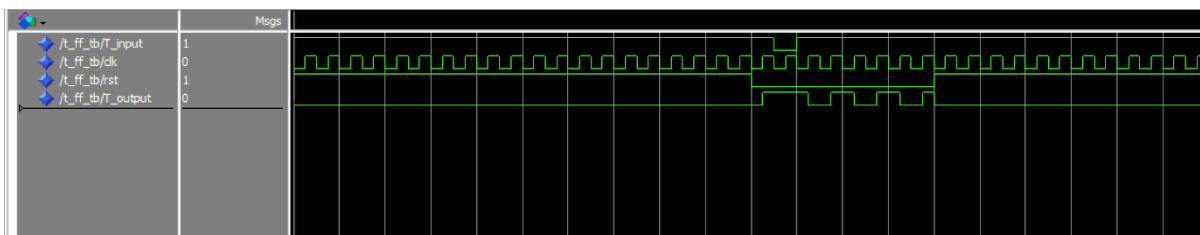
reset	T	Q	Q+
1	X	0	0
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	0



و مدار فوق نمایان گر نمایش T_FF است و می توانیم مقدار بعدی کلاک را به صورت مدار مقدار خروجی xor آن با t نوشت.

$$Q^+ = T'Q + TQ' = T \oplus Q$$

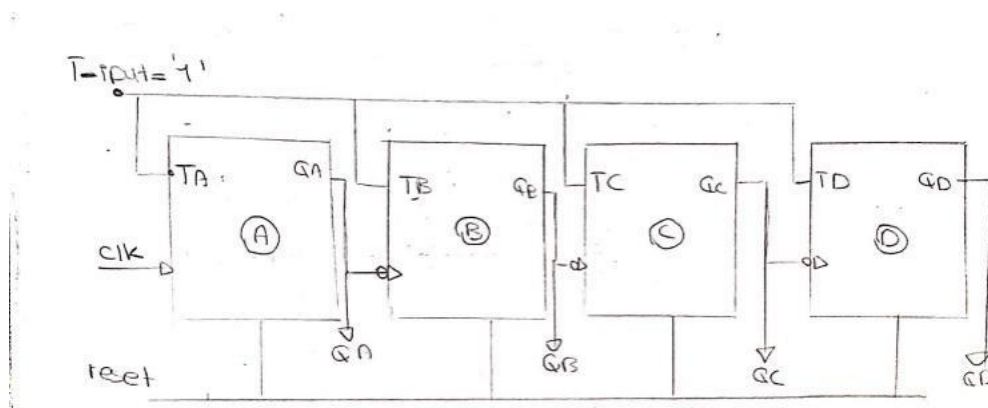
همانطور که مشاهده می کنید خروجی مدار به صورت زیر است



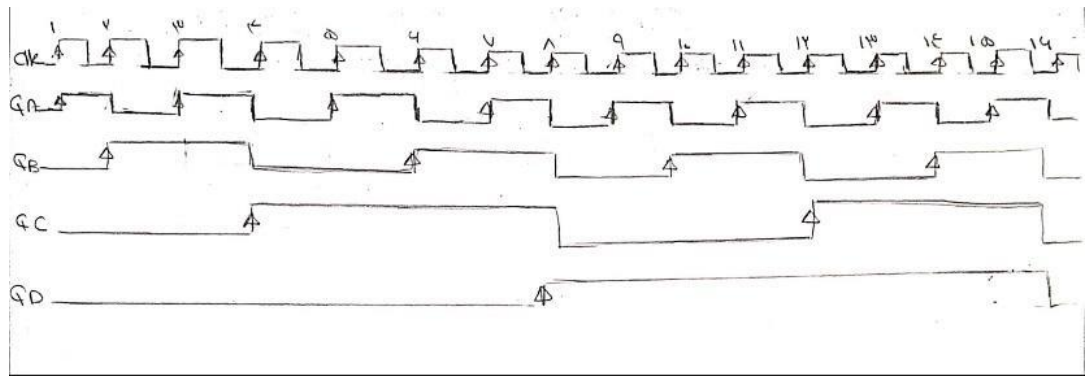
- زمانی که $reset = 1$ است هر مقداری ورودی باشد مدار غیرفعال است و خروجی 0 میشود.
 - زمانی که $reset = 0$ و ورودی مدار در کلاک اول 1 است پس $1 \text{ xor } (T)$ و $0(Q)$ را حساب می کنیم و خروجی 1 میشود.
 - در کلاک بعدی مقدار $T = 0$ است و مقدار خروجی قبلی (Q) نیز می دانیم 1 است بنابراین 1 xor این دو نیز 1 و همچنان خروجی 1 می شود
- این روش تا آخر ادامه پیدا می کند تا زمانی که $reset$ فعال می شود و خروجی 0 می شود.

Ripple counter

همانطور که می دانید ripple counter یک شمارنده آبشاری است که اگر بخواهیم آنرا به صورت ترتیبی نمایش دهیم به صورت زیر می شود و خروجی T فلیپ فلاپ قبل not می شود و به عنوان کلاک به t فیلپ بعدی داده می شود. دقت کنید که کلاک های T فلیپ فلاپ ها یکسان نیستند.



اگر شکل مدار خروجی را رسم کنیم مشاهده می کنیم:



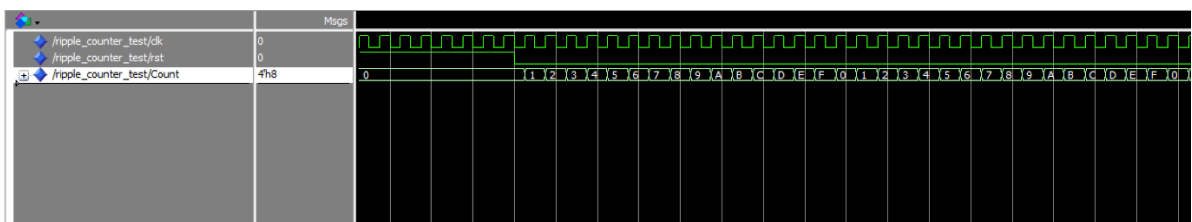
مشاهده می کنیم در کلاک بعدی مقادیر 1 دونه افزایش پیدا کردند و toggle ها را از نمودار فوق نیز مشاهده می کنیم.

Ripple Counter Concept

Present state				Next state			
Q ₃	Q ₂	Q ₁	Q ₀	Q ₃	Q ₂	Q ₁	Q ₀
0	0	0	0	0	0	0	1
0	0	0	1	0	0	1	0
0	0	1	0	0	0	1	1
0	0	1	1	0	1	0	0
0	1	0	0	0	1	0	1
0	1	0	1	0	1	1	0
0	1	1	0	0	1	1	1
0	1	1	1	1	0	0	0
1	0	0	0	1	0	0	1
1	0	0	1	1	0	1	0
1	0	1	0	1	0	1	1
1	0	1	1	1	1	0	0
1	1	0	0	1	1	0	1
1	1	0	1	1	1	1	0
1	1	1	0	1	1	1	1
1	1	1	1	0	0	0	0

27

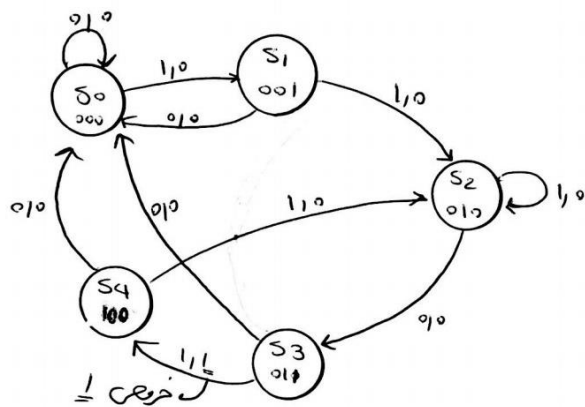
شکل موج خروجی:



مشاهده می کنیم که تا reset غیرفعال می شود خروجی در کلاک های بالارونده بعدی به ترتیب به دونه افزایش پیدا می کنند.

1101 mealy sequence detector

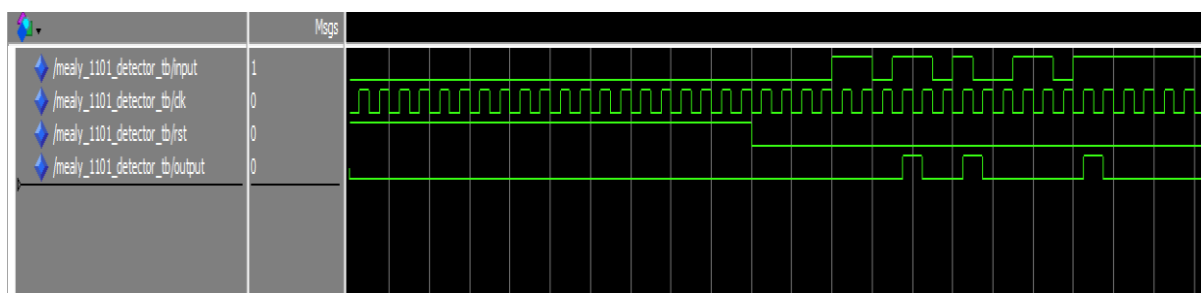
همانطور که می دانیم در مدار میلی ورودی و خروجی ها روی هر فلش نوشته می شود و برای پیدا کردن دنباله 1101 از الگوریتم زیر استفاده می کنیم:



نکات قابل توجہ:

- چون هر دفعه 1 بیت ورودی میگیریم پس دو حالت برای آن پیش می آید یا 0 باشد یا 1 بنابراین 2 تا فلش خروجی از هر state باید وجود داشته باشد
- همچنین ممکن است state ها به هر دلیلی از این 5 state خارج شوند و وارد state دیگر شوند بنابراین در کد لحاظ شده هر موقع این اتفاق افتاد به state s0 برگردد.
- هر جا 1 reset باشد مدار در حالت 000 قرار میگیرد.

خروجی، کد به صورت زیر است:



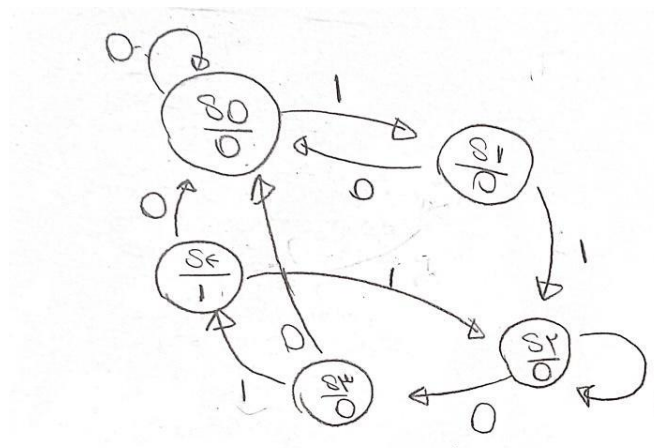
همانطور که مشاهده میکنیم ابتدا عدد 1 را پیدا می‌کند (s1) سپس در کلاک بعدی نیز دوباره عدد 1 را پیدا می‌کند (s2) بعد در کلاک بعدی 0 مشاهده می‌کند (s3) و بعد در کلاک بعدی 1 مشاهده می‌کند (s4) در اینجا خروجی 1 می‌شود.

بلافاصله در کلاک بعدی نیز عدد خوانده شده 1 است بنابراین overlapping رخ داده است و تا الان 11 داریم و از s4 وارد s2 میشویم و در کلاک بعدی نیز 0 دریافت شده وارد s3 می شویم و بعد آن نیز دوباره 1 پیدا می شود(s4) و باری دیگر خروجی 1 می شود.

در کلاک بعدی ورودی 0 است از s4 به s0 منتقل می شویم و باری دیگر پس از ورودی های 1101 دنباله را پیدا می کنیم.

1101 moore sequence detector

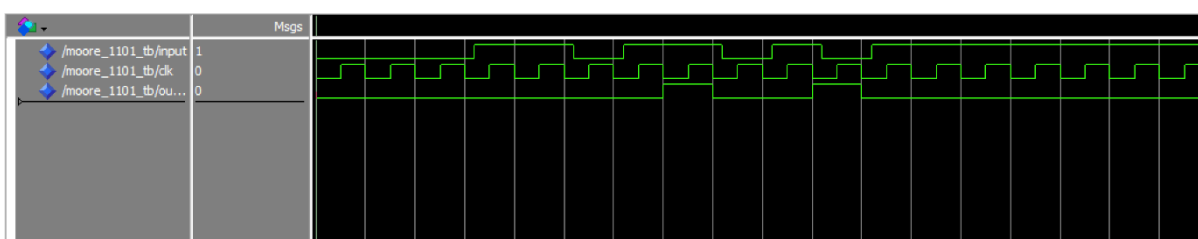
همانطور که می دانید مدار moore در خروجی هایش درون هر state مشخص می شود و اگر بخواهیم دیاگرام حالت آن را بکشیم به صورت زیر خواهد بود:



نکات قابل توجه:

- چون هر دفعه 1 بیت ورودی میگیریم پس دو حالت برای آن پیش می آید یا 0 باشد یا 1 بنابراین 2 تا فلش خروجی از هر state باید وجود داشته باشد
- همچنین ممکن است state ها به هر دلیلی از این 5 state خارج شوند و وارد state دیگر شوند بنابراین در کد لحاظ شده هر موقع این اتفاق افتاد به state s0 برگردد.

پس از رسم مدار خروجی به حالت زیر در می آید:

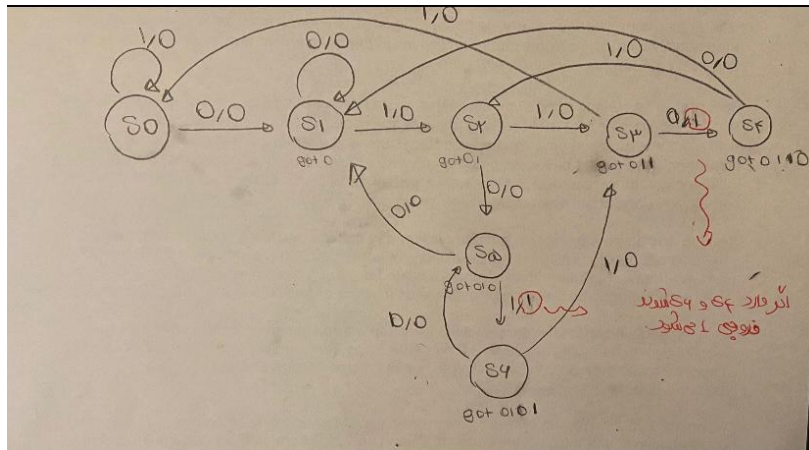


- دقت شود کلاک حساس به لبه پایین رونده است.
- همانطور که مشاهده می کنید در اولین لبه پایین رونده مقدار ورودی 0 است بنابراین در S0 می ماند.
- پس از آنکه ورودی 1 می شود در اولین لبه پایین رونده وارد S1 میشود و با دریافت 1 دوم در کلاک بعدی وارد S2 میشود پس از آن با دریافت 0 وارد S3 میشود و در کلاک بعدی نیز ورودی 1 است بنابراین دنباله 1101 پیدا شده و خروجی 1 می شود.

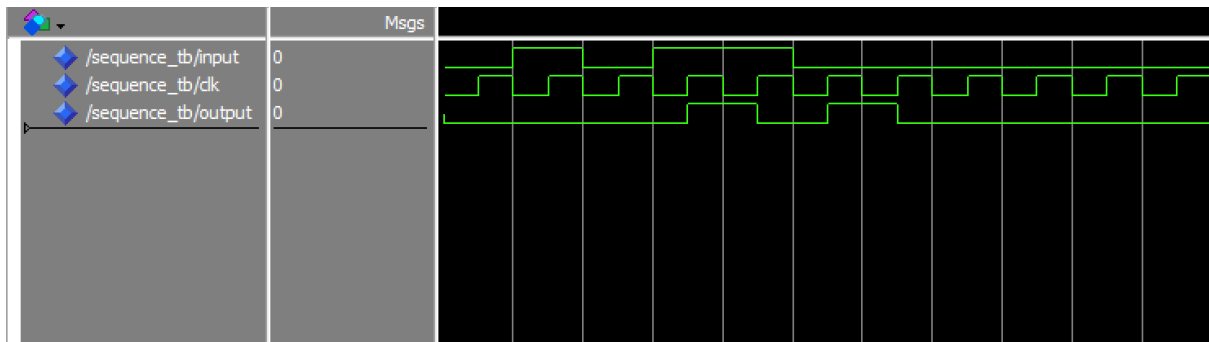
- پس از آنکه آخرین 1 را دریافت کردیم و وارد s4 شده بودیم در کلاک بعدی با دریافت 1 وارد s2 می شویم و حالت overlapping رخ داده است و تا الان 11 را داریم به دریافت 0 در کلاک بعدی وارد s3 می شویم و با دریافت 1 آخر وارد s4 می شویم و دنباله 1101 دوباره پیدا شد و خروجی باری دیگر 1 شد.

Sequence detector for 0110 and 0101

برای طراحی ماشینی که دو دنباله را با حالت overlapping درست کند باید طراحی زیر را داشته باشه باشیم که ما به صورت ماشین میلی طراحی کردیم.



پس از طراحی مشاهده می کنیم که خروجی به صورت زیر است:



- ابتدا در کلاک بالارونده اول 0 پیدا می شود پس وارد s1 میشویم در کلاک بعدی نیز 1 پیدا می شود که وارد s2 می شویم در کلاک بعدی 0 پیدا می شود پی وارد s5 می شویم و در کلاک بعدی نیز 1 پیدا می شود و وارد s6 می شویم و به این معناست که 0101 پیدا شده و خروجی 1 می شود.

- در کلاک بعدی بعد از 1 0101 پیدا می شود پس یعنی overlapping رخ داده است و وارد s3 می شویم و در کلاک بعدی نیز 0 دیده می شود که باعث وارد به s4 می شود و دنباله 0110 پیدا شده است و خروجی باری دیگر یک میشود.