

ل 1 D 1: فرایس کار بر روی دهای مدارهای منطقی:

token production:

2: مدار تقطیع توکن به خود و در این مدار؛ طبق توکنهای دستورات باید توکن بر اساس یک پارک و یک اوتو تولید شود در این فرآیند پارک خود و خروجی از پارک؛ کار ساز باشد.

بنابراین دردی ها به صورت park-number و pattern هستند هر دو عین سببی بوده و توسط این دو ورودی قصد رمزنگاری و تولید token که آن نیز عین سببی است را داریم.

طبق اصول و روابط زیر برای XOR داریم:

$$A \oplus 0 = A, A \oplus A = 0, (A \oplus B) \oplus C = A \oplus (B \oplus C), (B \oplus A) \oplus A = B \oplus 0 = B$$

همچنان طبق مثال زیر: حاصل XOR کردن دو ورودی است به طوری که عمل XOR طبق  $(A+B) \cdot (A+B)$  به صورت بیت به بیت انجام شده و خروجی به صورت زیر است. بنابراین از بیت XOR می توان برای رمزنگاری استفاده کرد. بنابراین در این مدار در ورودی verilog سه ورودی که دو ورودی park-number و pattern را به صورت بیت به بیت XOR می نماید تا خروجی مورد نیاز تولید شود.

(همچنین علاوه بر توکنهای دستورات می توان بیت رمزنگاری انواع هفتاد و دو بیتی را به طوری که رمزنگاری ای نوعی از رمزها است که بدلیل سادگی و سرعت مفید است و در پیاده سازی سخت افزاری نیز به صرفه بوده و توسط آشکارساز قابل پیاده سازی است به طوری که یکی از ورودی های پیاده سازی و تولید رمزنگاری این استفاده از بیت XOR می باشد.)

مدار 3: مدار خروجی خود و Exit park:

در قدم اول در خروجی خود و از پارک ابتدا باید ترجمه شود. آیا پارک در حال خروج می باشد یا خیر به طوری که توسط exit این بررسی انجام می شود و فرض ما بر این است بوده که  $exit = 1$  می باشد در حال خروج است (مقدار خروج دارد)  $exit = 0$  می باشد در حال توقف خروجی از پارک را ندارد.

در قدم بعد (یعنی از فرض این می باشد وجود دارد که مقدار خروجی داشته باشد) عمل رمزنگاری باید صورت گیرد.

Decrypt: عمل رمزنگاری؛ منظور تولید park number می باشد به طوری که ورودی ها token, pattern هستند و بنابراین به خروجی رسد از آنجایی که رمزنگاری معکوس عمل رمزنگاری می باشد؛ با داشتن و وارد کردن خروجی و یک ورودی (در مدار قبل) می توان عمل رمزنگاری را انجام داد به طوری که با جای گذاری دردی و خروجی عین عمل انجام می شود و حتی به عنوان مثال (طبق مثال مدار 2) داریم:

بنابراین طبق Decrypt نیز می تواند توسط بیت به بیت XOR کردن حاصل شود. در ورودی verilog نیز توصیف شده است.

در مدار 3: مدار خروجی خود و Exit park:



در ادامه نیاز است که با استفاده از خروجی decrypt که اکنون در ادامه Ddata داریم؛ park location  
 که هست یعنی است برسم؛ park location؛ محل قرارگیری خود را تعیین کنند؛ هست یعنی بودن آن  
 به این معنا است که موقعیت خودرو مورد توجه در مقایسه با سایر خودروها (از نظر پارک بودن و نبودن) به خالی بودن یا نبودن  
 پارکینگ (مقایسه می شود).

Ddata در واقع شماره پارک خودرو است و سه بیت برای این ورودی به خروجی هست یعنی موقعیت خودرو را تعیین  
 می کند برسم از decoder استفاده می کنیم که 3:8 است.

Ddata [2:0] park location [7:0]

x y z	D <sub>0</sub>	D <sub>1</sub>	D <sub>2</sub>	D <sub>3</sub>	D <sub>4</sub>	D <sub>5</sub>	D <sub>6</sub>	D <sub>7</sub>
0 0 0	1	0	0	0	0	0	0	0
0 0 1	0	1	0	0	0	0	0	0
0 1 0	0	0	1	0	0	0	0	0
0 1 1	0	0	0	1	0	0	0	0
1 0 0	0	0	0	0	1	0	0	0
1 0 1	0	0	0	0	0	1	0	0
1 1 0	0	0	0	0	0	0	1	0
1 1 1	0	0	0	0	0	0	0	1

مطابق جدول مقابل از decoder  
 3 به 8؛ ما توان از شماره پارک خودرو به  
 محل پارک آن رسید؛ به عنوان که برای مثال  
 اگر شماره پارک به صورت 000 باشد معادل  
 باینری به دیجیتال آن 0 است بنابراین بیت مقتم  
 از park location محل پارک این خودرو بوده و  
 اکنون که خودرو پارکینگ را ترک کرده و از آنجا می  
 جایی پارک خالی با یک و جای پارک پر شده با منفی مشخص می شود؛ بیت منفی یک خواهد شد.  
 ما برای اشاره به شش داریم؛ سفر دوم (000) معادل باینری به دیجیتال آن؛ یک است بنابراین بیت دوم از  
 park location؛ یک خواهد شد.  
 بنابراین؛ طبق توضیحات بالا 3:8 decoder دقیقاً می تواند در Exit parking؛ کار ساز باشد.  
 در ورودی verif؛ توسط آن انجام شد.

این مدار توسط دریافت ورودی های parking capacity و park location در ورودی New Capacity؛  
 هست یعنی هستند؛ خروجی New capacity تولید خواهد شد.  
 parking capacity؛ عددی هست یعنی است که وضعیت کل پارکینگ را طبق خودروهای پارک شده در پارکینگ توسط  
 هوا با توجه به دریا خالی بودن حرکت از جای پارک ها؛ ارائه می دهد.  
 park location؛ از مدار قبل این خروجی هست یعنی حاصل شده که طبق توضیحات قبل؛ توسط یک و منفی  
 محل خودروای که به تازگی از پارکینگ خارج شده؛ مشخص می شود.

حال برای این که از دو ورودی بالا یک بشیریم و وضعیت پارکینگ را به روز رسانی کنیم و جای پارک خالی جدید را نیز مشخص  
 کنیم؛ ما توانیم از OR استفاده کنیم زیرا طبق جدول اینتر بیت های این بیت داریم؛  
 خاصیت لیت OR بین صورت است که اگر حداقل یکی از  
 ورودی ها؛ یک باشد؛ خروجی نیز یک خواهد شد.  
 برای مثال؛ اگر parking capacity = 00000000 و park location = 00000000  
 اگر بیت به بیت و OR کنیم؛ خروجی به صورت؛ New Capacity = 00010010  
 موقعیت بروز رسانی شد

parking capacity	park location	output	New Capacity
A	B		
0	0	0	
0	1	1	
1	0	1	
1	1	1	



مادرول ۴:

در این مادرول ما می خواهیم مدت زمان رسیدن به جایی در پارتیت ما بوده است را به دست آوریم برای

این کار ما ورود  $time\ in$  و تایم خروج  $time\ out$  را می گیریم و از تفاضل این دو مدت

زمان به ما می شن در پارتیت بوده است به دست می آید .  $time\ total = time\ out - time\ in$

پس در این حالت با استفاده از  $Full\ address$  می آسیم به بد بیت های  $time\ out$  را از  $time\ in$

کم می کنیم . برای این کار  $Sel$  را برای هر فول اید ، ۱- قرار می دهیم تا بیت های را کم کنند و همین

تعداد  $Carry$  فول اید اول را نیز می نذاریم . پس در واقع ما سرزیر می داریم :

$$time\ total = time\ out + time\ in + 1$$

Parking Capacity [0]	Parking Capacity [1]	Parking Capacity [2]	Parking Capacity [3]	Parking Capacity [4]	Parking Capacity [5]	Parking Capacity [6]	Parking Capacity [7]	Parking Capacity [8]	True-Capacity
1	x	x	x	x	x	x	x	x	1
x	1	x	x	x	x	x	x	x	1
x	x	1	x	x	x	x	x	x	1
x	x	x	1	x	x	x	x	x	1
x	x	x	x	1	x	x	x	x	1
x	x	x	x	x	1	x	x	x	1
x	x	x	x	x	x	1	x	x	1
x	x	x	x	x	x	x	1	x	1
x	x	x	x	x	x	x	x	1	1
0	0	0	0	0	0	0	0	0	0

Entry	True-Capacity	Enable
0	0	0
0	1	0
1	0	0
1	1	1

⇒ and بیت

or بیت

Enable	Edata [V]	Edata [4]	Edata [5]	Edata [6]	Edata [3]	Edata [2]	Edata [1]	Edata [0]	Park Number
0	x	x	x	x	x	x	x	x	xxx
1	1	x	x	x	x	x	x	x	111
1	0	1	x	x	x	x	x	x	110
1	0	0	1	x	x	x	x	x	101
1	0	0	0	1	x	x	x	x	100
1	0	0	0	0	1	x	x	x	011
1	0	0	0	0	0	1	x	x	010
1	0	0	0	0	0	0	1	x	001
1	0	0	0	0	0	0	0	1	000



ماژول ۱:

در این ماژول ما می خواهیم معین کنیم که آیا پارکینگ ما جا دارد که ماشین دارد آن شود راننده جا دارد نزدیک ترین محل خالی برای پارک ماشین چیست. این ماژول از دو بخش متصل شده در بخش

اول آن که entry checker داریم. دوم ورودی entry و parking capacity را می گیرد.

این دو بخش به ture capacity و parking capacity را می گیریم و به دست می دهیم به ی

ماژول ۲ می کنیم علت این کار این است که اگر در این ۸ بیت هر یک عدد داشته باشیم

جواب ۲ ماژول می کند و ما می فهمیم که یک جای خالی برای پارک وجود دارد یا اگر ماشین

برای پارک در پارکینگ مانده باشد ما جای خالی داریم پس باید بیت ۱ and entry را که می گذارد

کننده حضور ماشین چیست ture capacity می گوییم اگر هر دو درست بود ماشین می تواند داخل

باید پس Enable مربوط به ماژول بعد را می گیریم. در بخش دوم ما یک encoder

الویت دارد صراحت می کنیم که En آن جواب از بخش اول و data آن نیز parking capacity

است و خروجی آن یک عدد می باشد که نزدیک ترین محل خالی برای پارک را نشان می دهد.