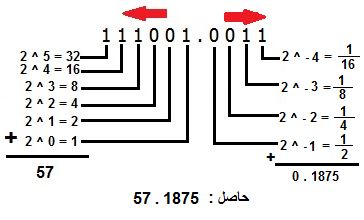
**بسم الله الرحمن الرحیم**

**سخت افزار**

***استاد جوان***

**سوال :** این عدد اعشاری را به مبنای 10 ببرید.

111001.0011



**نکته :** ارقام 0 در اعداد بالا در جمع محاسبه نمی شوند.

(46.3)10 = (101110 . 01)2

**نکته :**

**اعداد صحیح :**

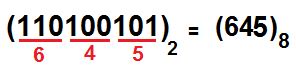
**اعداد اعشاری:**

**رابطه عدد اعشاری / عدد صحیح :**



**تبدیل مبنای 2 به 8 :**



**سوال کنکوری :** در عبارت مقابل b کدام است؟

الف ) 2 ب) 7 ج) 5 د)8

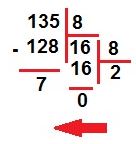
جواب =>

گزینه الف نیست چون عدد از 0 و 1 بیشتر است. گزینه ج نیست چون از 6 بزرگتر است. پس :

در نتیجه گزینه ب صحیح است .

**سوال :** حاصل عبارت را بدست اورید. (23)16 + (100)10 = (?)8

**جواب :** ابتدا عبارت را هم مبنا کرده و با هم جمع می کنیم سپس به مبنای 8 می بریم.(23)16 = (35)10 🡺 (35)10 + (100)10 = (135)10

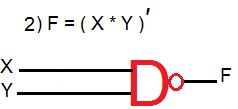
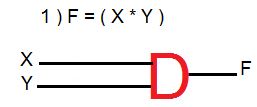
 (35)10 + (100)10 = (207)8

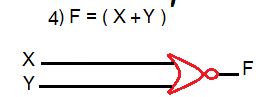
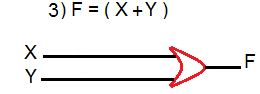
**نکته :** تبدیل گیگ به ترا یا بایت یا بالعکس امتحانی است.

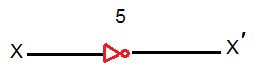
**مکمل گیری :**

مثال : عبارت روبرو را ساده کنید.

**اشکال :**

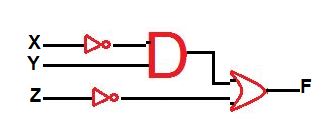
 



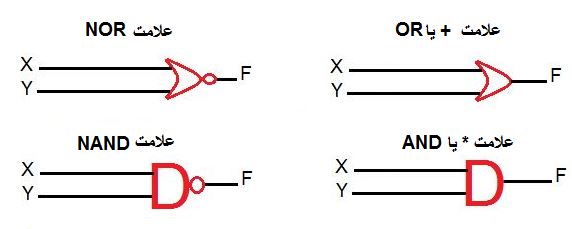
**مثال :** را پیاده سازی کنید.

12.JPG

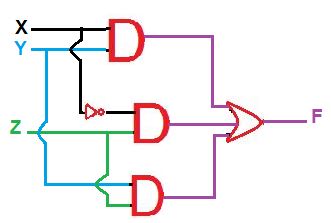
**مثال :** را پیاده سازی کنید.



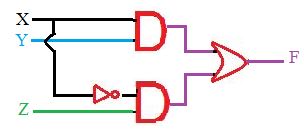
**نکته :**



**مثال :** را پیاده سازی کنید.



**مثال :** عبارت را بعد از ساده سازی، پیاده سازی کنید.



**نکته :** عبارت برابر 1 که در ضرب استفاده می شود و تاثیری در ان ندارد. عبارت برابر 0 که در جمع استفاده می شود و تاثیری در ان ندارد. هر دو عبارت در ساده سازی کاربرد دارند.

**نکته :** اگر ورودی گیت OR را مکمل بگیریم، می توانیم به جای گیت OR، گیت NAND قرار دهیم و اگر ورودی های گیت NAND را مکمل بگیریم ، می توانیم به جای ان گیت NOR قرار دهیم.

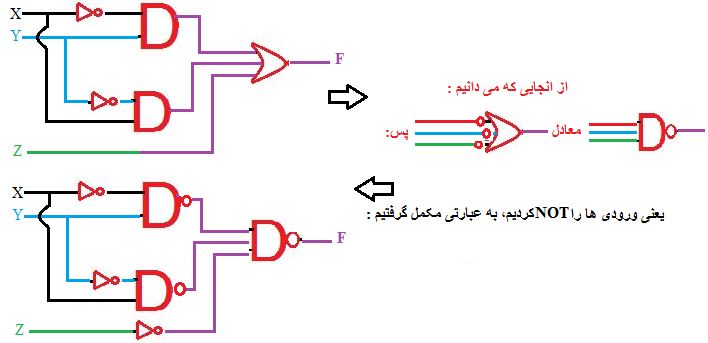
17.JPG

برای مکمل گیری داخل پرانتز هم تک تک اعضا را مکمل می گیریم، و هم عمگرها :

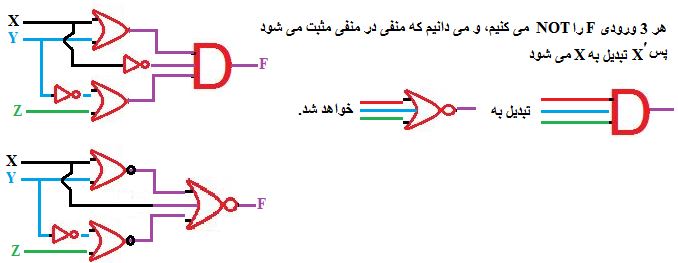
18.JPG

**سوال :** تابع را فقط با گیت NAND پیاده سازی کنید.

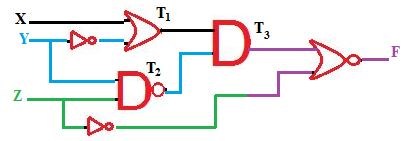
ابتدا تابع را پیاده سازی و سپس انرا به NAND تبدیل می کنیم :



**سوال :** تابع را فقط با گیت NOR پیاده سازی کنید.

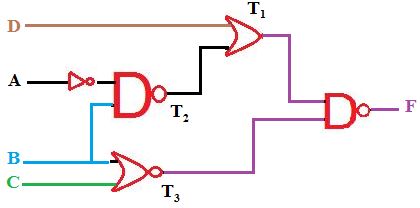


**سوال:** با استفاده از مدار زیر تابع F را بدست اوردید.



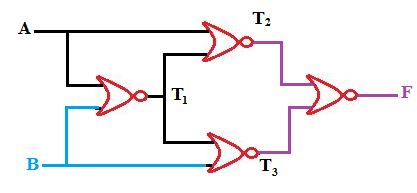
روش حل اسان برای این سوال این است که خروجی ها را نام گذاری کنید و تفکیک کنید. **مثال :**

**سوال :** در شکل داده شده رابطه منطقی F پس از ساده شدن چیست؟



**نکته :** در گیتOR اگر یک طرف 1 باشد حتما خروجی 1 می شود پس جواب برابر با 1 می شود.

**سوال :** مدار زیر معادل کدام گیت منطقی است؟

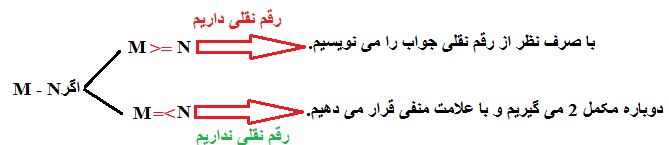


دقت شود T1 و T2 و T3 را که نامگذاری کردیم، هر سه بعد از گیت های NOR امده است. چون بعد از NOT امده پس پریم( ‘ ) برایش نمی گذاریم یعنی این پریم برای جمله قبل از گیت است پس T1 می شود و پریم محاسبه می شود پس در T2 دیگر پریم محاسبه نمی شود و دیگر نمی نویسیم و این پریم روی حرفT غلط است چون برای عبارت بوده است. دقت کنید که حتما در خروجی ها نامگذاری انجام شود.

**تفریق باینری :**



**نکته :** برای تفریق باینری از مکمل 2 استفاده کرده سپس جمع می کنیم :



**مین ترم (m) و ماکس ترم (M) :**

مین ترم حاصلضرب استاندارد است. OR معادل NAND و AND معادل NOR است.

**نکته :** جدول ابتدای فصل مربوط به همین مبحث در اسلاید ها مطالعه شود .

3 متغیر را می توان به 23 حالت نوشت :

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |
|  |  |  |  |

**قانون :**

همیشه در مین ترم ها A برابر 1 و A’ برابر 0 است ولی در ماکس ترم ها A برابر 0 و A’ برابر 1 است.

**نکته :** نتیجه گیری جدول ابتدای فصل این است که مین ترم ها و ماکس ترم ها مکمل یکدیگرند.

**مثال :** الف) مجموع مین ترم ها مطلوب است که :

ب ) مطلوب است حاصلضرب ماکسترم ها :

**سوال :** مین ترم ها و ماکسترم های F1 و F2 را بنویسید.

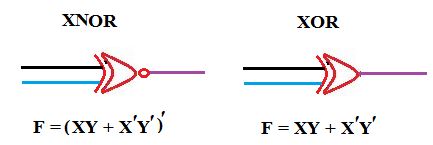
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| F2 | F1 | Z | Y | X |  |
| 1 | 0 | 0 | 0 | 0 | 0 |
| 1 | 0 | 1 | 0 | 0 | 1 |
| 1 | 1 | 0 | 1 | 0 | 2 |
| 0 | 0 | 1 | 1 | 0 | 3 |
| 0 | 1 | 0 | 0 | 1 | 4 |
| 0 | 0 | 1 | 0 | 1 | 5 |
| 0 | 1 | 0 | 1 | 1 | 6 |
| 0 | 0 | 1 | 1 | 1 | 7 |

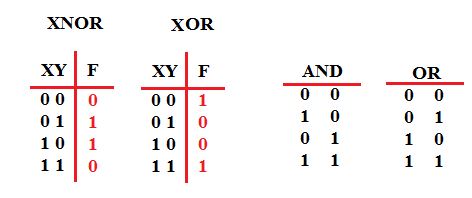
**مثال :** تابع را به صورت حاصل مین ترم بنویسید.

**نکته :** در سوال قبل اگر ماکسترم ها را می خواست، جواب 0 و 2 و 3 یعنی تمام نواقص این تابع است.

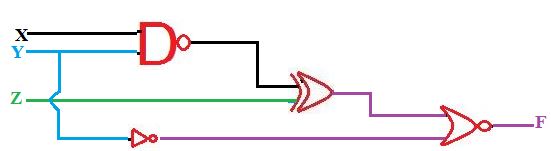
شرط این که بتوانیم مین ترم ها را بنویسیم این است که باید هر 3 متغیر را داشته باشیم. در این جا جمله اول فقط X است یعنی Y و Z نداریم در این حالت چون حاصلضرب نیاز داریم که به حاصل جمع تبدل کنیم پس هر تابع هر کمبود متغیری داشته باشد ضرب در عدد خنثی یعنی یک می کنیم و تمام عبارت بدست امده را می نویسیم. در اینجا متغیر و و حکم عدد یک را داشت ولی استفاده ابزاری کردیم.

**XOR و XNOR :**

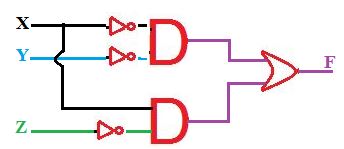




**سوال :** با توجه به مدار زیر تابع F را محاسبه کنید.



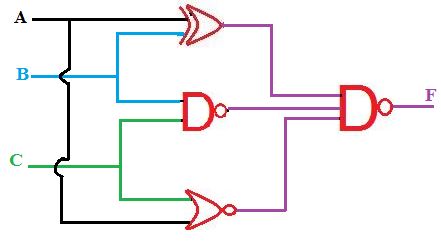
**سوال :** تابع زیر را پیاده سازی کنید.



**سوال :** تابع F را با استفاده از ماکسترم ها بنویسید.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| F | Z | Y | X |  |
| 1 | 0 | 0 | 0 | 0 |
| 0 | 1 | 0 | 0 | 1 |
| 0 | 0 | 1 | 0 | 2 |
| 1 | 1 | 1 | 0 | 3 |
| 0 | 0 | 0 | 1 | 4 |
| 1 | 1 | 0 | 1 | 5 |
| 1 | 0 | 1 | 1 | 6 |
| 1 | 1 | 1 | 1 | 7 |

**سوال :** کزینه صحیح را انتخاب کنید.



1. (گزینه صحیح)

**روش حل :** برای حل نمودار های پیچیده ای مانند این بایدداز روش جایگذاری عدد استفاده کرد.

**سوال :** کدام یک از گیت های زیر معادل گیت NOT است؟

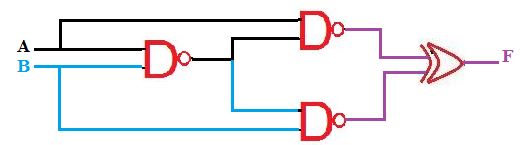
31.JPG گزینه صحیح

32.JPG

33.JPGگزینه صحیح

34.JPG

**سوال :** کزینه صحیح را انتخاب کنید.



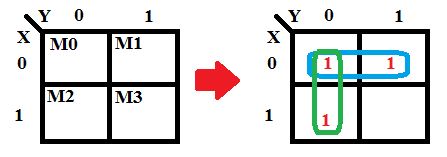
1. (گزینه صحیح)

**روش حل :** خروجی 1 است، پس گزینه 2 و 4 حذف می شود چون جوابشان 0 است. چون دو جواب 1 و 3 می ماند عدد جایگزین می کنیم، حالا می بینیم که گزینه 1 حذف می شود چون 0 \* 0 = 0 و در نتیجه گزینه 3 صحیح است.

**جدول کارنو**

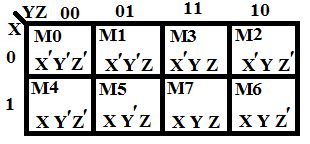
**سوال :** با استفاده از جدول کارنو عبارت ساده کنید.

با توجه به عبارت داخل خانه های مورد نظر 1 گذاشته و انها را انتخاب می کنیم و متغیر های مشترک خانه ها را می نویسیم.



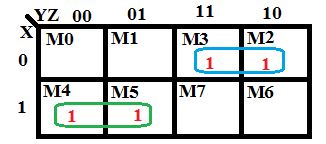
**نکته :** در جدول کارنو اگر عدد 0 باشد متغیر پریم ( ‘ ) می گیرد .

**کارنو 3 متغیره :**

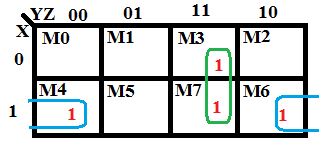


|  |  |  |
| --- | --- | --- |
| عدد | متغیر | نام خانه |
| 000 |  | M0 |
| 001 |  | M1 |
| 010 |  | M2 |
| 011 |  | M3 |
| 100 |  | M4 |
| 101 |  | M5 |
| 110 |  | M6 |
| 111 |  | M7 |

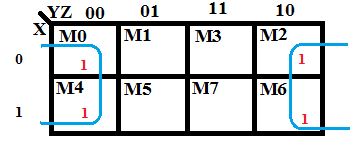
**سوال :** عبارت را ساده کنید.

**

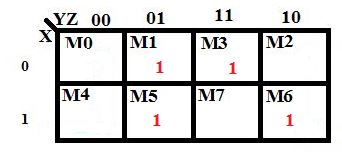
**سوال :** عبارت را ساده کنید.



**سوال :** تابع F را با توجه به جدول زیر بنویسید.

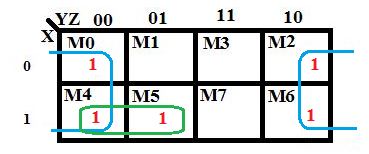


**سوال :** عبارت را ساده کنید.

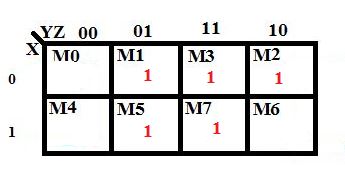


***سوال :*** *اگر عبارت ساده شده تابع 3 متغیره به روش جدول کارنو باشد، مینترم های تابع* F *بدست اورید.*

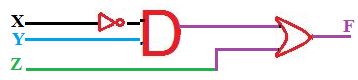
*چون* Z’ *یک متغیره است وبه استناد اسلاید 11 فصل مربوطه می فهیم که از 4 خانه استفاده کرده واز انجایی جایگاه* Z’ *را می دانیم پس 1 را داخل ان ها قرار می دهیم.* XY’ *دو متغیره است و می توان فهمید دو مربع همجوار است.*

**

**سوال :** عبارت را ساده کنید.

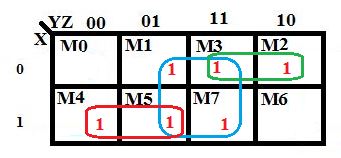
**

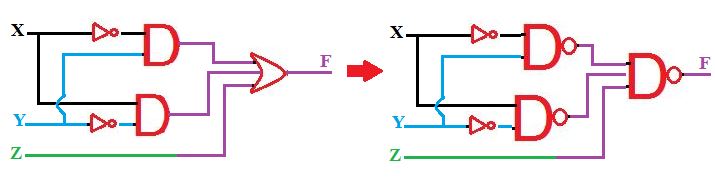
*برای حل اینگونه سوالات باید جملات را ساده کنیم چون نیاز به 3 متغیر داریم پس :*



**سوال :** عبارت را فقط با گیت NAND پیاده سازی کنید.

ابتدا شکل ساده عبارت را پیاده سازی می کنیم سپس برای پیاده سازی با NAND ورودی ها را NOT می کنیم :





**نکته :** در کارنو 3 متغیره :

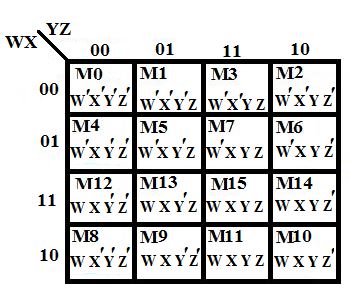
1 خانه => یک عبارت 3 متغیره بوجود می اید.

2 خانه => یک عبارت 2 متغیره بوجود می اید.

4 خانه => یک عبارت 1 متغیره بوجود می اید.

8 خانه => تابع همیشه عدد 1 است.( F = 1 )

**کارنو 4 متغیره :**



**نکته :** در کارنو 4 متغیره :

1 خانه => یک عبارت 4 متغیره بوجود می اید.

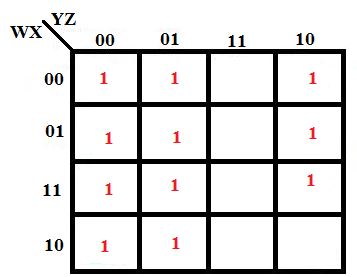
2 خانه => یک عبارت 3 متغیره بوجود می اید.

4 خانه => یک عبارت 2 متغیره بوجود می اید.

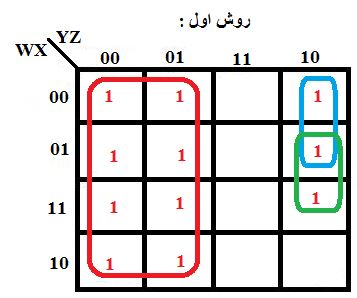
8 خانه => یک عبارت 1 متغیره بوجود می اید.

16 خانه => تابع همیشه عدد 1 است.( F = 1 )

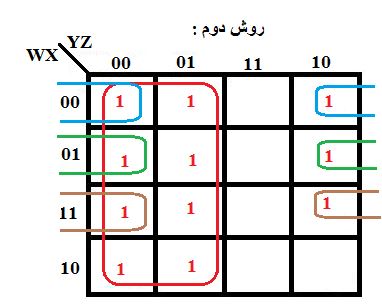
**سوال :** با استفاده از جدول زیر تابع F را بدست اورید.



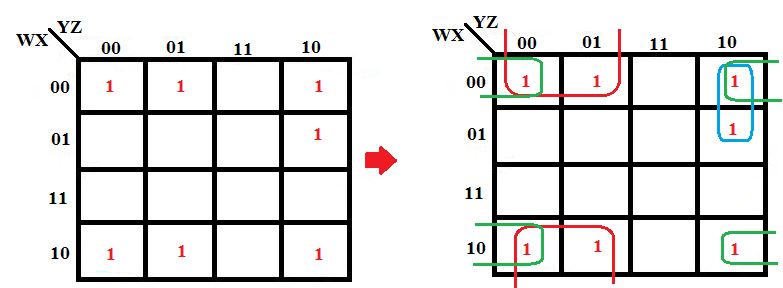
**روش اول :** صحت کمتری دارد، چون قابل فاکتورگیری است.



**روش دوم :**



**سوال :** با استفاده از جدول زیر تابع F را بدست اورید.



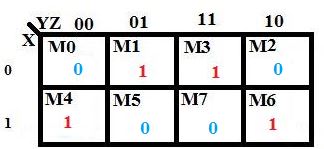
**سوال :** تابع F در مثال قبل را با استفاده از جدول کارنو ساده کنید.

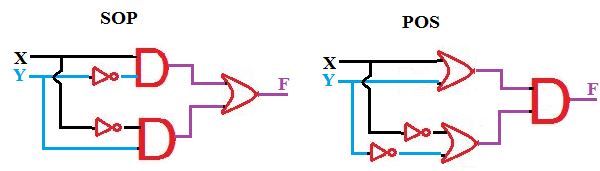
می توان 3 متغیره ها را در یک ضرب کرد یا به صورت ذهنی روی جدول مکان یابی کرد:

**SOP و POS :**

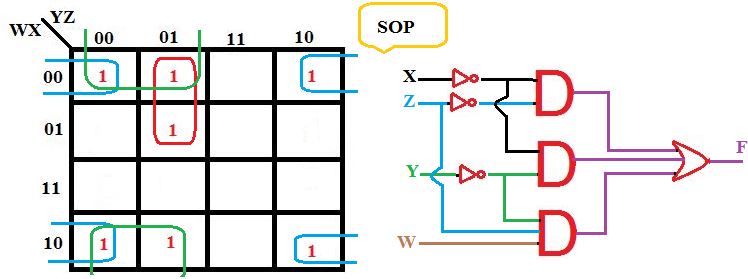
تمام مثال هایی که درباره جدول کارنو انجام دادیم SOP گفته می شود یعنی مجموعه داده های موجود، حالا اگر خانه های ناموجود را حساب کنیم سپس پریم بگیریم به ان POS یعنی داده های ناموجود می گویند.

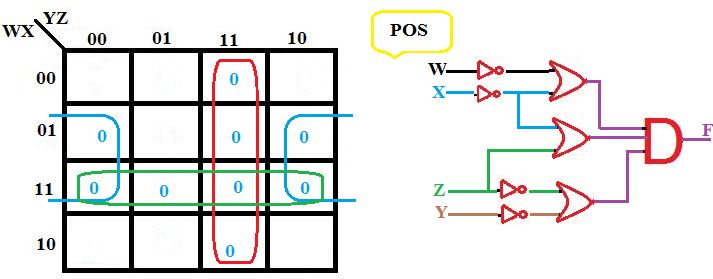
**مثال :** POS عبارت را بدست اورید و پیاده سازی کنید.



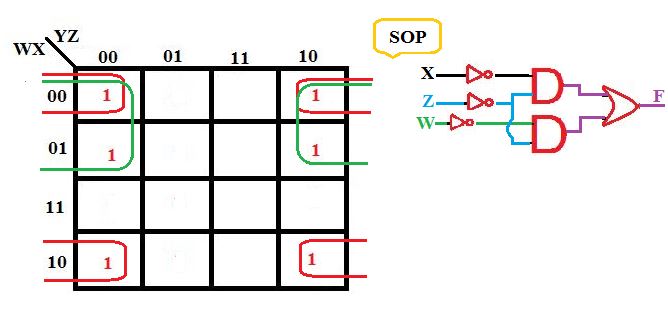


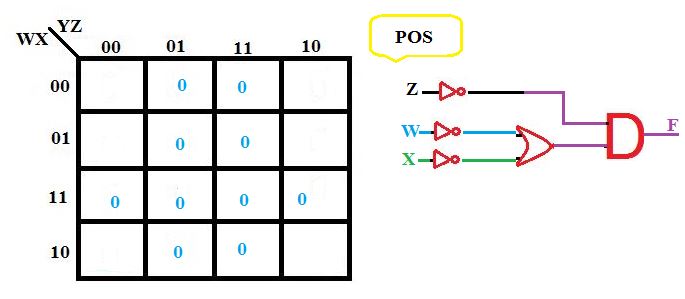
**مثال :** POS و SOP عبارت را بدست اورید و پیاده سازی کنید.





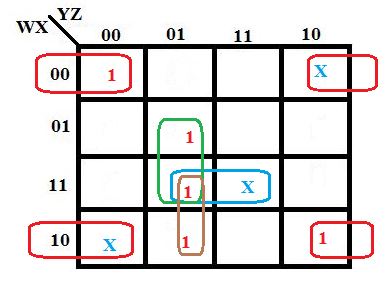
**مثال :** POS و SOP عبارت را بدست اورید و پیاده سازی کنید.



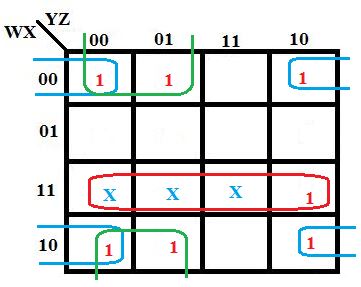


**شرایط بی اهمیت(DON’T CARE )**

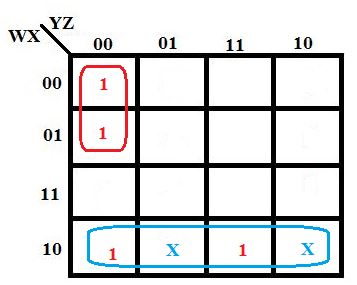
**سوال :** شرایط بی اهمیت را برای تابع مشخص کنید.



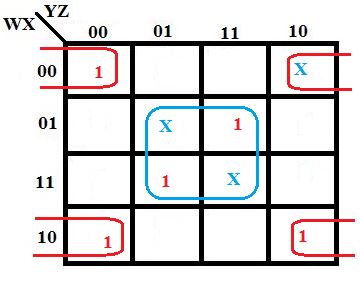
**سوال :** شرایط بی اهمیت را برای تابع مشخص کنید.



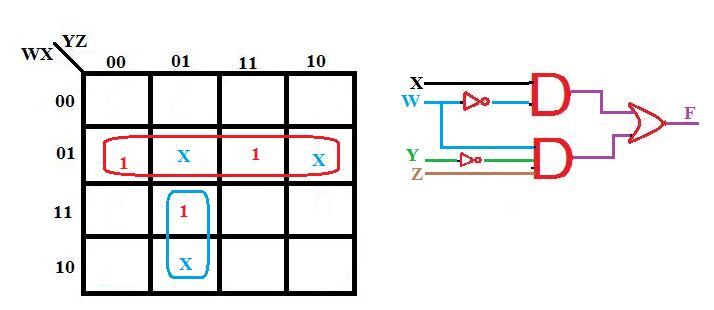
**سوال :** حداقل مقدارDC برای زمانی که خروجی برابر شود، کدام است؟



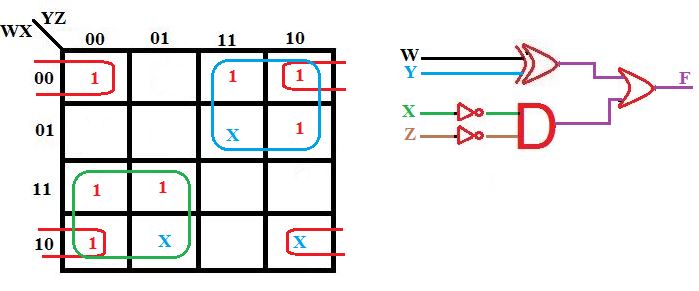
**سوال :** شرایط بی اهمیت را برای تابع مشخص کنید.



**سوال :** شرایط بی اهمیت را برای تابع مشخص کنید.



**سوال :** شرایط بی اهمیت را برای تابع مشخص کنید.



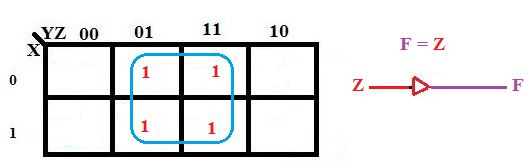
**یاد اوری :**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **NAND** | **XOR** | **OR** | **AND** | **Y** | **X** |
| **1** | **0** | **0** | **0** | **0** | **0** |
| **1** | **1** | **1** | **0** | **1** | **0** |
| **1** | **1** | **1** | **0** | **0** | **1** |
| **0** | **0** | **1** | **1** | **1** | **1** |

**انالیز مدار های ترکیبی**

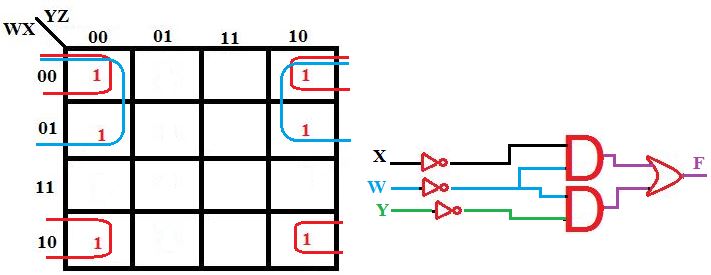
**سوال :** مداری با 3 ورودی و 1 خروجی طراحی کنید به نحوی که هنگامی که مقدار معادل عددی ورودی ها عدد فرد باشد، 1 برابر شود؟

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **F** | **Z** | **Y** | **X** | **عدد** |
| **0** | **0** | **0** | **0** | **0** |
| **1** | **1** | **0** | **0** | **1** |
| **0** | **0** | **1** | **0** | **2** |
| **1** | **1** | **1** | **0** | **3** |
| **0** | **0** | **0** | **1** | **4** |
| **1** | **1** | **0** | **1** | **5** |
| **0** | **0** | **1** | **1** | **6** |
| **1** | **1** | **1** | **1** | **7** |



**سوال :** مداری با 4 ورودی و 1 خروجی طراحی کنید به نحوی که هنگامی که مقدار معادل عددی ورودی ها اعداد زوج کوچکتر از 11 باشد، 1 برابر شود؟

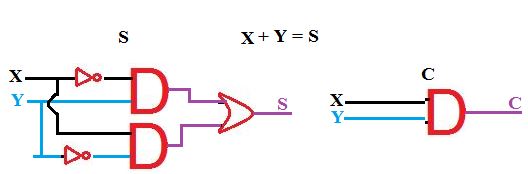
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **F** | **W** | **Z** | **Y** | **X** | **عدد** |
| **1** | **0** | **0** | **0** | **0** | **0** |
| **0** | **1** | **0** | **0** | **0** | **1** |
| **1** | **0** | **1** | **0** | **0** | **2** |
| **0** | **1** | **1** | **0** | **0** | **3** |
| **1** | **0** | **0** | **1** | **0** | **4** |
| **0** | **1** | **0** | **1** | **0** | **5** |
| **1** | **0** | **1** | **1** | **0** | **6** |
| **0** | **1** | **1** | **1** | **0** | **7** |
| **1** | **0** | **0** | **0** | **1** | **8** |
| **0** | **1** | **0** | **0** | **1** | **9** |
| **1** | **0** | **1** | **0** | **1** | **10** |
| **0** | **1** | **1** | **0** | **1** | **11** |
| **0** | **0** | **0** | **1** | **1** | **12** |
| **0** | **1** | **0** | **1** | **1** | **13** |
| **0** | **0** | **1** | **1** | **1** | **14** |
| **0** | **1** | **1** | **1** | **1** | **15** |

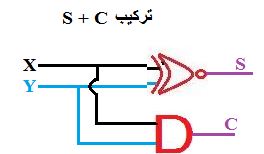


**جمع کننده ها (ADDER)**

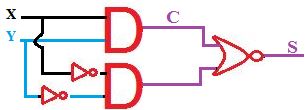
**نیم جمع کننده (HALF ADDER) :**

|  |  |  |  |
| --- | --- | --- | --- |
| C (CARRY) | S (SUM) | Y | X |
| 0 | 0 | 0 | 0 |
| 0 | 1 | 1 | 0 |
| 0 | 1 | 0 | 1 |
| 1 | 0 | 1 | 1 |





عبارت S را ساده کردیم، حالا پیاده سازی می کنیم :



**جمع کننده کامل (FULL ADDER) :**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **C** | **S** | **Z** | **Y** | **X** | **عدد** |
| **0** | **0** | **0** | **0** | **0** | **0** |
| **0** | **1** | **1** | **0** | **0** | **1** |
| **0** | **1** | **0** | **1** | **0** | **2** |
| **1** | **0** | **1** | **1** | **0** | **3** |
| **0** | **1** | **0** | **0** | **1** | **4** |
| **1** | **0** | **1** | **0** | **1** | **5** |
| **1** | **0** | **0** | **1** | **1** | **6** |
| **1** | **1** | **1** | **1** | **1** | **7** |



**نمونه سوال امتحانی :** با استفاده از نیم جمع کننده (HALF ADDER) ، یک جمع کننده کامل (FULL ADDER) بسازید.

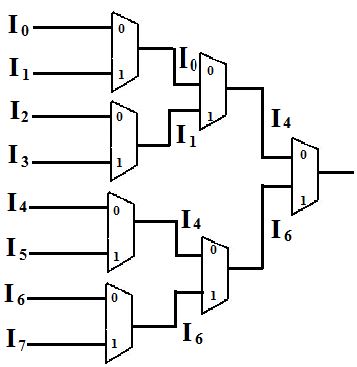
مدار های ترکیبی

مقایسه کننده ها(comprator) (امتحانی نیست!) :

A = 111 B = 111

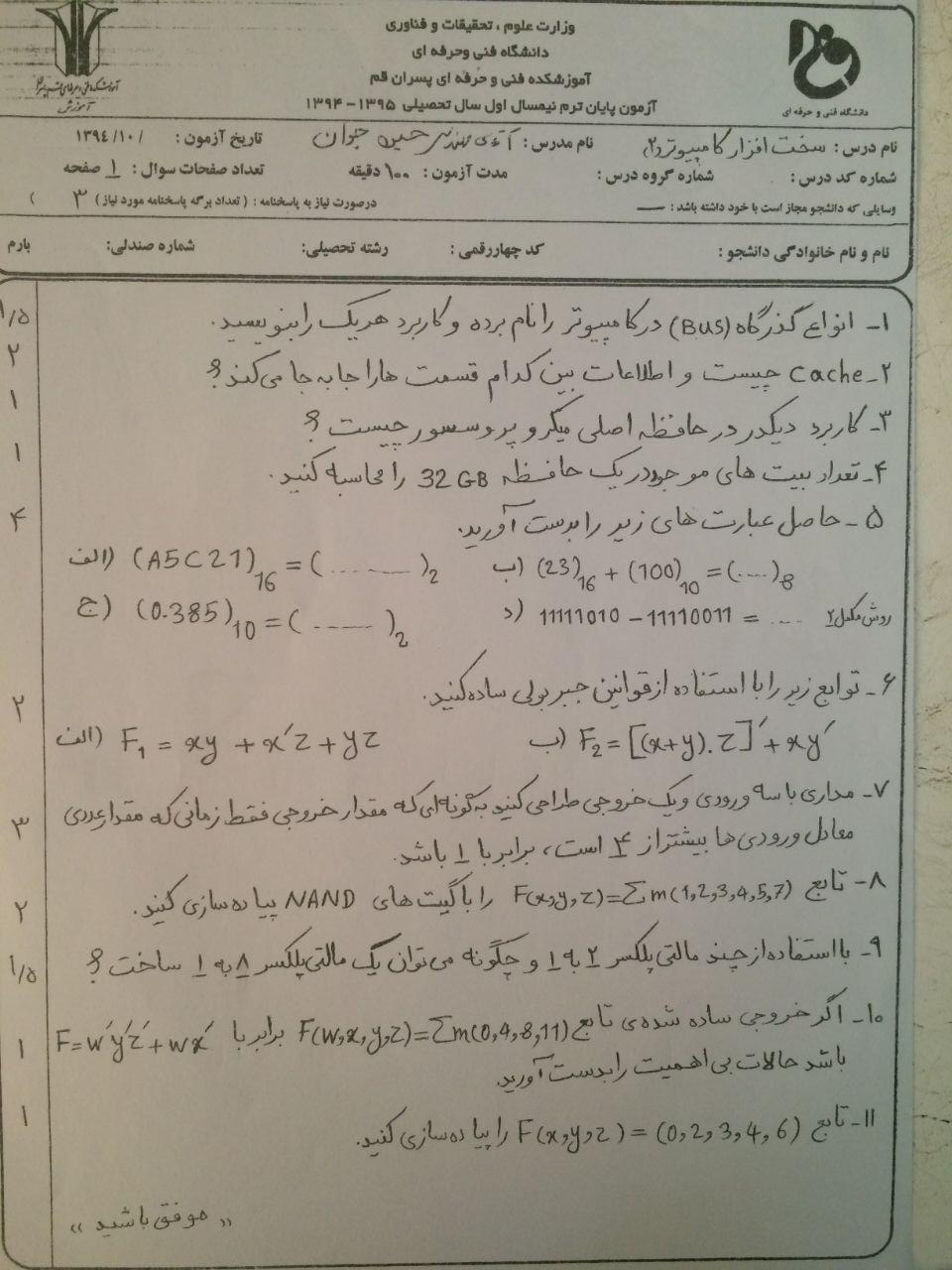
<= فرمول

چون 1=1=1 پس A و B مساوی هستند.



مثلا اگر I6 را بخواهیم، طبق پیگیری خط ها 6=110 .

نکته : DECODER امتحانی است.



نکاتی درباره عکس بالا :

سوال 3 : کاربرد دیکدر یا احتمالا کاربرد دیگری را بخواهد.

سوال 7 : یا همین سوال است یا اعداد تغییر می کنند.

سوال 8 : عینا همین سوال را می دهد.

سوال 9 : عینا همین سوال را می دهد.

سوال 10 : احتمالا اعداد تعویض شوند .

سوال 11 : احتمال تعویض تابع است.