

SUMADOR BINARIO DE TRES BITS n.k.

El diseño consiste en realizar un sumador binario de tres bits usando el GAL22V10.

Para esto usaremos sumadores half adder (H.A) y full adder (F.A) sabiendo sus ecuaciones, sin antes recordar que el semisumador H.A. tiene dos entradas y dos salidas, sin embargo el sumador F.A. tiene tres entradas (la tercera es el acarreo anterior) y dos salidas.

Sabemos las siguientes ecuaciones para estos sumadores:

H.A:

$$S = A \oplus B$$

$$C = A \cdot B$$

F.A:

$$S_n = A_n \oplus B_n \oplus C_{n-1}$$

$$C_n = A_n B_n + B_n C_{n-1} + A_n C_{n-1}$$

Entonces usando esta lógica podemos realizar básicamente la siguiente programación para el gal usando el programa WinCupl:

```
Name      SUMADOR ;
PartNo    00 ;
Date      13/06/2017 ;
Revision  01 ;
Designer  Engineer ;
Company   umsa ;
Assembly  None ;
Location  ;
Device    G22V10 ;

/* ***** INPUT PINS ***** */
PIN      2 =    a1          ; /*
PIN      3 =    a2          ; /*
PIN      4 =    a3          ; /*
PIN      5 =    b1          ; /*
PIN      6 =    b2          ; /*
PIN      7 =    b3          ; /*

/* ***** OUTPUT PINS ***** */
PIN      23 =    a          ; /*
PIN      22 =    b          ; /*
PIN      21 =    c          ; /*
PIN      20 =    d          ; /*

a=a1$b1;
c1=a1&b1;

b=(a2$b2)$c1;
c2=(a2&b2)#(b2&c1)#(a2&c1);
```

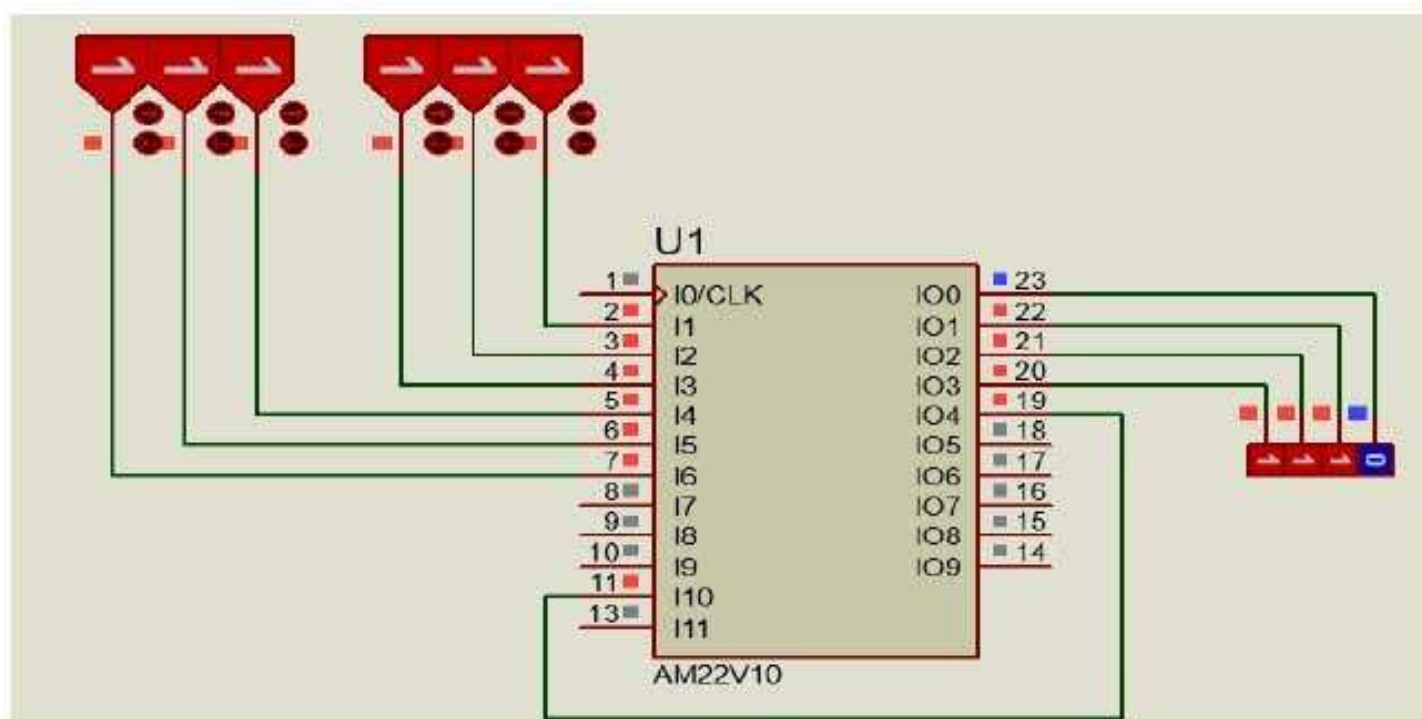
```
c=(a3$b3)$c2;  
d=(a3&b3)#(b3&c2)#(a3&c2);
```

Sin embargo esto no funcionará debido a que a la componente 'c' estamos metiendo demasiadas variables ya que 'c' tiene una variable 'c2' y esta es toda una funcion, por lo que nos marcara error.

Para corregir el anterior problema usaremos un artificio, este consiste en usar una salida extra (PIN 19) que irá a una entrada extra (PIN 11) para que no se cargue demasiadas variables como lo hicimos anteriormente, es decir ahora nos queda:

```
Location ;  
Device G22V10 ;  
  
/* ***** INPUT PINS ***** */  
PIN 2 = a1 ; /* */  
PIN 3 = a2 ; /* */  
PIN 4 = a3 ; /* */  
PIN 5 = b1 ; /* */  
PIN 6 = b2 ; /* */  
PIN 7 = b3 ; /* */  
PIN 11 = za ; /* */  
  
/* ***** OUTPUT PINS ***** */  
PIN 23 = a ; /* */  
PIN 22 = b ; /* */  
PIN 21 = c ; /* */  
PIN 20 = d ; /* */  
PIN 19 = zb ; /* */  
  
a=a1$b1;  
c1=a1&b1;  
  
b=(a2$b2)$c1;  
zb=(a2&b2)#(b2&c1)#(a2&c1);  
  
c=(a3$b3)$za;  
d=(a3&b3)#(b3&za)#(a3&za);
```

Ahora esto lo simularemos en Proteus y la circuitería o el circuito final será:



INK