

# PRIMERA ACTIVIDAD EVALUABLE

Asignatura: **FUNDAMENTOS de SISTEMAS DIGITALES**

*Título de la Actividad:*

**Diseño, Implementación, Simulación y Validación de un Circuito en Lógica Combinacional**

*Datos personales:*

*Código de la actividad que le ha correspondido realizar:* **A-E-1\_036.doc**

*Enunciado.*

## Enunciado A-E-1\_036.doc

Disponemos de una palabra de 4 bits ( $P_3, P_2, P_1$  y  $P_0$ ) y de una señal adicional,  $x$ , con las que queremos controlar el funcionamiento de una Unidad Aritmético Lógica, pero la programación de la ALU no depende directamente de estas señales sino de la paridad o no de la palabra junto con el valor "0" o "1" de la variable  $x$ . Así, el criterio para controlar las operaciones que realiza la ALU sobre las 2 palabras de 4 bits [ $A(A_3, A_2, A_1, A_0)$  y  $B(B_3, B_2, B_1, B_0)$ ] es el siguiente:

- a) Si la palabra es par y  $x=1$ , la ALU hace la operación aritmética sin acarreo  $A PLUS \bar{A}B$ .
- b) Si la palabra es par y  $x=0$ , la ALU hace la operación lógica  $\bar{A} \oplus B$ .
- c) Si la palabra es impar y  $x=1$ , la ALU hace la operación aritmética con arrastre  $A PLUS \bar{A}B PLUS 1$ .
- d) Si la palabra es impar y  $x=0$  entonces la ALU debe ponerse a 0.

Diseñe el circuito del codificador y úselo para controlar las operaciones de la ALU que se han especificado.

## SOLUCION:

Lo primero que debemos hacer para diseñar el circuito, es centrarnos en la tabla de verdad de la ALU SN74181 que es la que emplearemos para realizar el ejercicio, dicha tabla es la mostrada a continuación:

SELECCIÓN	DATO ACTIVO EN ALTA		
S3 S2 S1 S0	M = H	M = L Operaciones Aritméticas	
	Funciones Lógicas	$\overline{C_n} = H$ (sin acarreo)	$\overline{C_n} = L$ (con acarreo)
L L L L	$F = \overline{A}$	$F = A$	$F = A \text{ PLUS } 1$
L L L H	$F = \overline{A + B}$	$F = A + B$	$F = (A + B) \text{ PLUS } 1$
L L H L	$F = \overline{A}B$	$F = A + \overline{B}$	$F = (A + \overline{B}) \text{ PLUS } 1$
L L H H	$F = 0$	$F = \text{MINUS } 1$	$F = \text{Cero}$
L H L L	$F = \overline{A}\overline{B}$	$F = A \text{ PLUS } \overline{A}\overline{B}$	$F = A \text{ PLUS } \overline{A}\overline{B} \text{ PLUS } 1$
L H L H	$F = \overline{B}$	$F = (A + B) \text{ PLUS } \overline{A}\overline{B}$	$F = (A + B) \text{ PLUS } \overline{A}\overline{B} \text{ PLUS } 1$
L H H L	$F = A \oplus B$	$F = \text{MINUS } B \text{ MINUS } 1$	$F = A \text{ MINUS } B$
L H H H	$F = \overline{A}\overline{B}$	$F = \overline{A}\overline{B} \text{ MINUS } 1$	$F = \overline{A}\overline{B}$
H L L L	$F = \overline{A} + B$	$F = A \text{ PLUS } AB$	$F = A \text{ PLUS } AB \text{ PLUS } 1$
H L L H	$F = \overline{A} \oplus \overline{B}$	$F = A \text{ PLUS } B$	$F = A \text{ PLUS } B \text{ PLUS } 1$
H L H L	$F = B$	$F = (A + \overline{B}) \text{ PLUS } AB$	$F = (A + \overline{B}) \text{ PLUS } AB \text{ PLUS } 1$
H L H H	$F = AB$	$F = AB \text{ MINUS } 1$	$F = AB$
H H L L	$F = 1$	$F = A \text{ PLUS } A$	$F = A \text{ PLUS } A \text{ PLUS } 1$
H H L H	$F = A + \overline{B}$	$F = (A + B) \text{ PLUS } A$	$F = (A + B) \text{ PLUS } A \text{ PLUS } 1$
H H H L	$F = A + B$	$F = (A + \overline{B}) \text{ PLUS } A$	$F = (A + \overline{B}) \text{ PLUS } A \text{ PLUS } 1$
H H H H	$F = A$	$F = A \text{ MINUS } 1$	$F = A$

Una vez que tenemos la tabla de la ALU, debemos determinar cada una de las salidas que queremos implementar en el circuito generador de señales, en la siguiente tabla podemos ver cuales son las configuraciones de cada señal según la palabra de entrada y la señal X y que función se activaría en la ALU para cada una de ellas, para ello debemos tener en cuenta el circuito que nos produce la generación de paridad para los 4 bits de entrada además del valor de la señal X.

A continuación podemos ver la tabla de verdad del circuito generador de paridad.

P3	P2	P1	P0	Z
0	0	0	0	0
0	0	0	1	1
0	0	1	0	1
0	0	1	1	0
0	1	0	0	1
0	1	0	1	0
0	1	1	0	0
0	1	1	1	1
1	0	0	0	1
1	0	0	1	0
1	0	1	0	0
1	0	1	1	1
1	1	0	0	0
1	1	0	1	1
1	1	1	0	1
1	1	1	1	0

Se puede observar que el generador produce como salida un 0 si el numero de bits de la palabra que están a 1 es par.

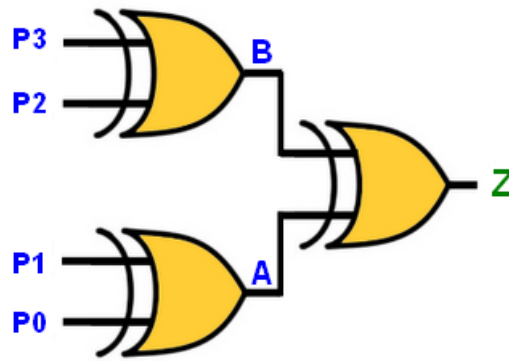
Podemos tratar de reducir la función de salida mediante la siguiente tabla de karnaugh

<div> <div>P3</div> <div>P1</div> <div>P2</div> <div>P0</div> </div>		00	01	11	10
		00	01	11	10
00	00	0	1	0	1
01	01	1	0	1	0
11	11	0	1	0	1
10	10	1	0	1	0

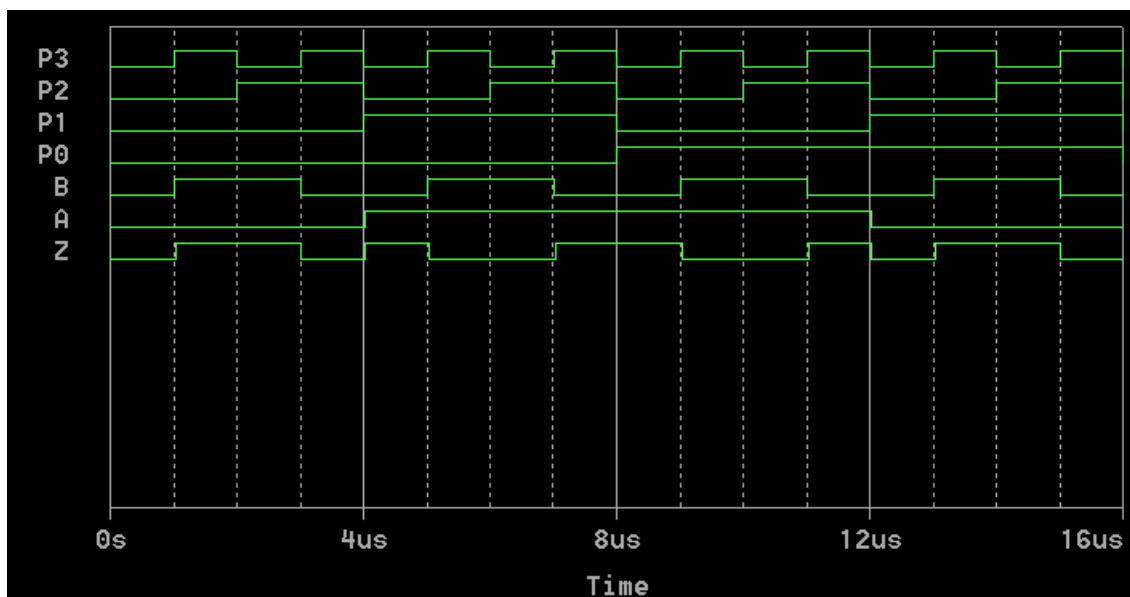
Pero como se puede observar la función de salida es irreducible, siendo esta igual a la función OR EXCLUSIVE de cada uno de los bits de entrada.

$$Z = P3 \oplus P2 \oplus P1 \oplus P0$$

Dicha función puede ser implementada mediante tres puertas OR EXCLUSIVE, una de ellas realizara la operación lógica OR EXCLUSIVE de los dos bits de menos peso, otra realizara la misma operación con los dos bits de mayor peso y una tercera realizara la operación OR EXCLUSIVE de los resultados obtenidos, el circuito seria el mostrado en la figura.



Cuyo cronograma es el siguiente:



Para la realización del cronograma se han empleado cuatro relojes con los siguientes valores de señal:

P0: ONTIME = 1 us , OFFTIME = 1 us.

P1: ONTIME = 2 us , OFFTIME = 2 us.

P2: ONTIME = 4 us , OFFTIME = 4 us.

P3: ONTIME = 8 us , OFFTIME = 8 us.

Estos valores han sido establecidos para poder observar todas las posibles configuraciones de entrada.

Una vez que tenemos definidas todas las salidas del generador de paridad para cada configuración de entrada, debemos realizar la tabla de verdad del circuito generador de señales, en la figura se muestra la tabla de verdad completa.

P3	P2	P1	P0	Z	X	S3	S2	S1	S0	M	$\overline{Cn}$
0	0	0	0	0	0	H	L	L	H	1	X
0	0	0	1	1	0	L	L	H	H	1	X
0	0	1	0	1	0	L	L	H	H	1	X
0	0	1	1	0	0	H	L	L	H	1	X
0	1	0	0	1	0	L	L	H	H	1	X
0	1	0	1	0	0	H	L	L	H	1	X
0	1	1	0	0	0	H	L	L	H	1	X
0	1	1	1	1	0	L	L	H	H	1	X
1	0	0	0	1	0	L	L	H	H	1	X
1	0	0	1	0	0	H	L	L	H	1	X
1	0	1	0	0	0	H	L	L	H	1	X
1	0	1	1	1	0	L	L	H	H	1	X
1	1	0	0	0	0	H	L	L	H	1	X
1	1	0	1	1	0	L	L	H	H	1	X
1	1	1	0	1	0	L	L	H	H	1	X
1	1	1	1	0	0	H	L	L	H	1	X
0	0	0	0	0	1	L	H	L	L	0	1
0	0	0	1	1	1	L	H	L	L	0	0
0	0	1	0	1	1	L	H	L	L	0	0
0	0	1	1	0	1	L	H	L	L	0	1
0	1	0	0	1	1	L	H	L	L	0	0
0	1	0	1	0	1	L	H	L	L	0	1
0	1	1	0	0	1	L	H	L	L	0	1
0	1	1	1	1	1	L	H	L	L	0	0
1	0	0	0	1	1	L	H	L	L	0	0
1	0	0	1	0	1	L	H	L	L	0	1
1	0	1	0	0	1	L	H	L	L	0	1
1	0	1	1	1	1	L	H	L	L	0	0
1	1	0	0	1	1	L	H	L	L	0	1
1	1	0	1	0	1	L	H	L	L	0	0
1	1	1	0	0	1	L	H	L	L	0	0
1	1	1	1	1	1	L	H	L	L	0	1

	Palabra par y $X = 0$ (Funcion operación lógica $A \oplus B$ )
	Palabra par y $X = 1$ (Funcion operación aritmética si acarreo $A PLUS A\bar{B}$ )
	palabra impar y $X = 1$ (Funcion aritmética con arrastre $A PLUS A\bar{B} PLUS 1$ )
	palabra impar y $X = 0$ (Funcion logica 0)

Como se puede observar, cuando M esta a nivel alto, da igual cual sea el valor de  $\overline{Cn}$  ya que la ALU realizara una función lógica.

Simplemente observando la tabla de verdad, podemos determinar que el valor que debe tomar la señal M tan solo depende del valor que tome la señal X, siendo este igual a  $M = \overline{X}$

Por la sencillez de las señales que se deben implementar, se pueden deducir a simple vista el resto de las señales que necesitamos.

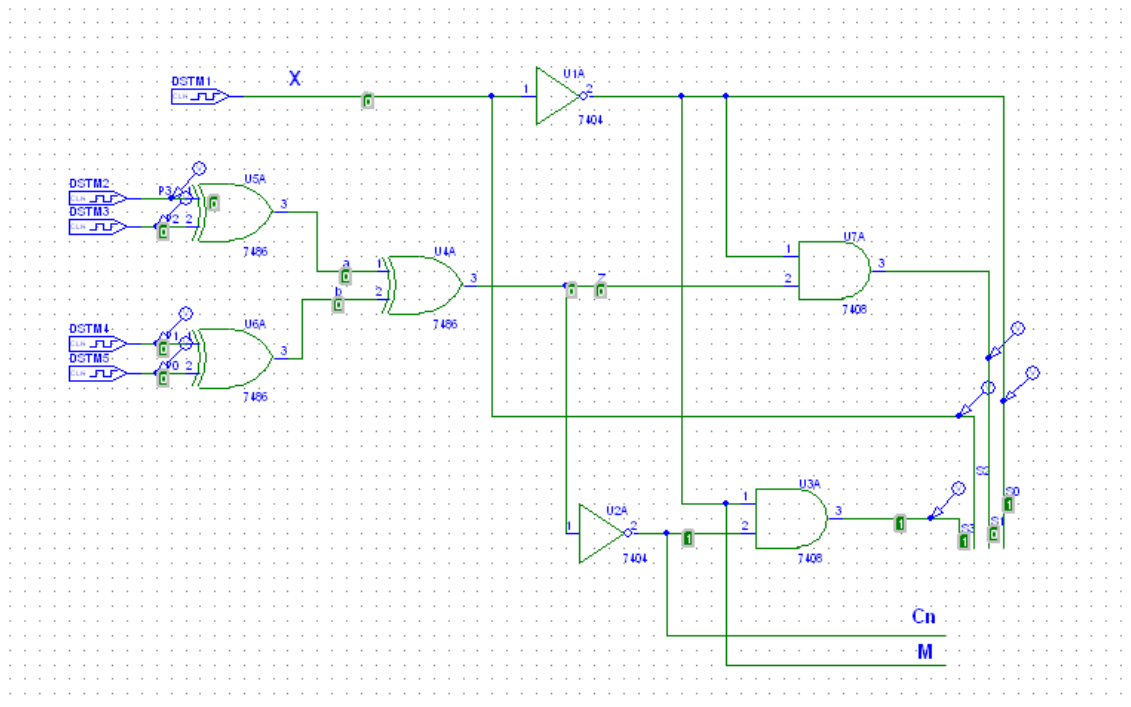
Observamos que la señal S0 solo depende del valor de X, siendo este igual a  $S0 = \overline{X}$

De la misma forma determinamos que el e valor de S1 es el siguiente  $S1 = Z\overline{X}$

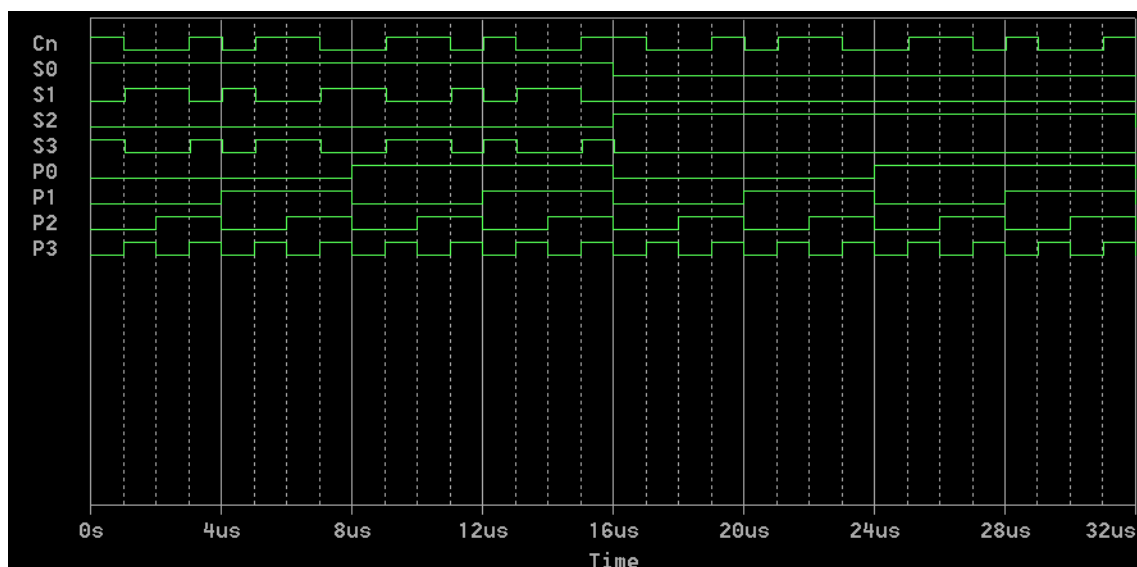
El valor de S2 solo depende del valor de X siendo igual a  $S2 = X$

Y el valor de S3 es igual a  $S3 = \overline{Z}X$

Una vez determinadas las salidas necesarias del circuito generador de señales, procedemos a su implementación, quedándonos el siguiente circuito:



El cronograma de este circuito es el siguiente:

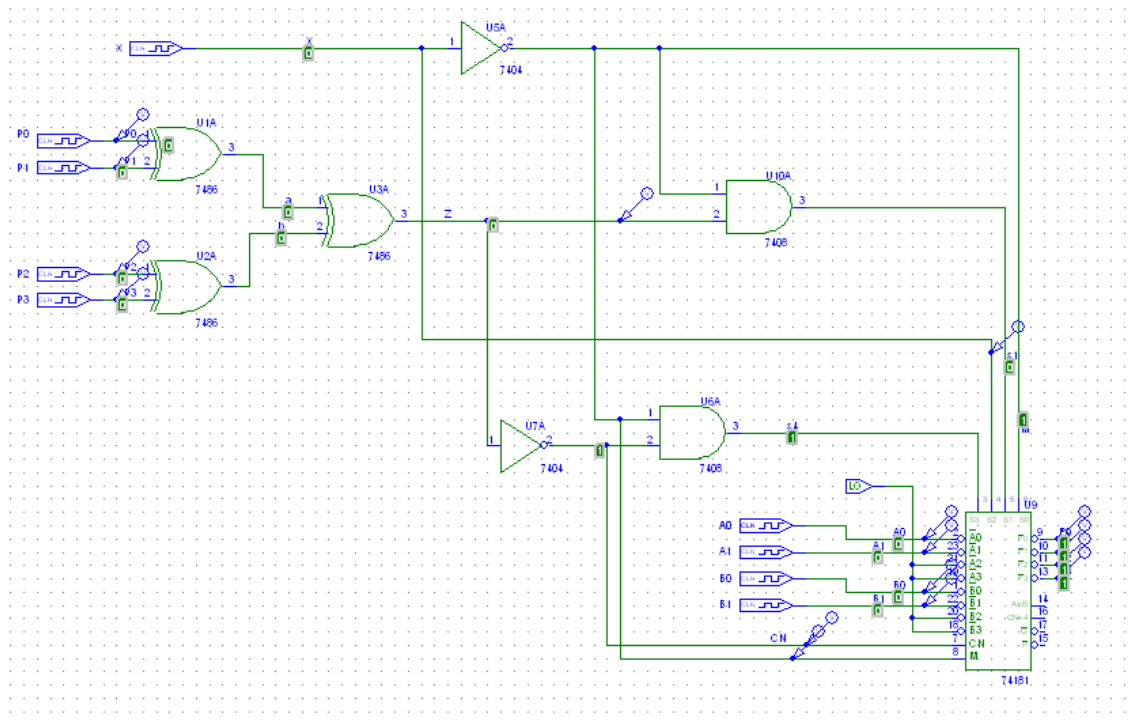


La señal M coincide con S0.

En cuanto a los valores empleados en las entradas, son los mismos que en el circuito generador de paridad visto anteriormente y para la señal de X (S2 en el cronograma) se han empleado las siguientes señales:

X: ONTIME = 16 us , OFFTIME = 16 us.

Estando definidas todas las señales, ya podemos conectar el circuito de control con la ALU como se muestra en la siguiente figura.



Para que el resultado sea mas fácil de entender, los dos bits mas significativos de cada palabra se han puesto a 0, realizándose las operaciones con los dos bits menos significativos, por este motivo la señal de salida Cn +4 siempre es 0, esta señal la emplearíamos si necesitamos conectar nuestro circuito con otra ALU para realizar operaciones con palabras de mas de 4 bits.

Las tablas de verdad de cada una de las diferentes funciones que hemos implementado son las siguientes:

# **A PLUS $\overline{A\overline{B}}$**

A3	A2	A1	A0	B3	B2	B1	B0	F3	F2	F1	F0	Cn
0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	0	0	0	1	0	0
0	0	1	0	0	0	0	0	0	1	0	0	0
0	0	1	1	0	0	0	0	0	1	1	0	0
0	0	0	0	0	0	0	1	0	0	0	0	0
0	0	0	1	0	0	0	1	0	0	0	1	0
0	0	1	0	0	0	0	1	0	1	0	0	0
0	0	1	1	0	0	0	1	0	1	0	1	0
0	0	0	0	0	0	1	0	0	0	0	0	0
0	0	0	1	0	0	1	0	0	0	1	0	0
0	0	1	0	0	0	1	0	0	1	0	0	0
0	0	1	1	0	0	1	0	0	1	0	0	0
0	0	0	0	0	0	1	1	0	0	0	0	0
0	0	0	1	0	0	1	1	0	0	0	1	0
0	0	1	0	0	0	1	1	0	0	1	0	0
0	0	1	1	0	0	1	1	0	0	1	1	0

# **$\overline{A\oplus B}$**

A3	A2	A1	A0	B3	B2	B1	B0	F3	F2	F1	F0
0	0	0	0	0	0	0	0	1	1	1	1
0	0	0	1	0	0	0	0	1	1	1	0
0	0	1	0	0	0	0	0	1	1	0	1
0	0	1	1	0	0	0	0	1	1	0	0
0	0	0	0	0	0	0	1	1	1	1	0
0	0	0	1	0	0	0	1	1	1	1	1
0	0	1	0	0	0	0	1	1	1	0	0
0	0	1	1	0	0	0	1	1	1	0	1
0	0	0	0	0	0	1	0	1	1	0	1
0	0	0	1	0	0	1	0	1	1	0	0
0	0	1	0	0	0	1	0	1	1	1	1
0	0	1	1	0	0	1	0	1	1	1	0
0	0	0	0	0	0	1	1	1	1	0	0
0	0	0	1	0	0	1	1	1	1	0	1
0	0	1	0	0	0	1	1	1	1	1	0
0	0	1	1	0	0	1	1	1	1	1	1

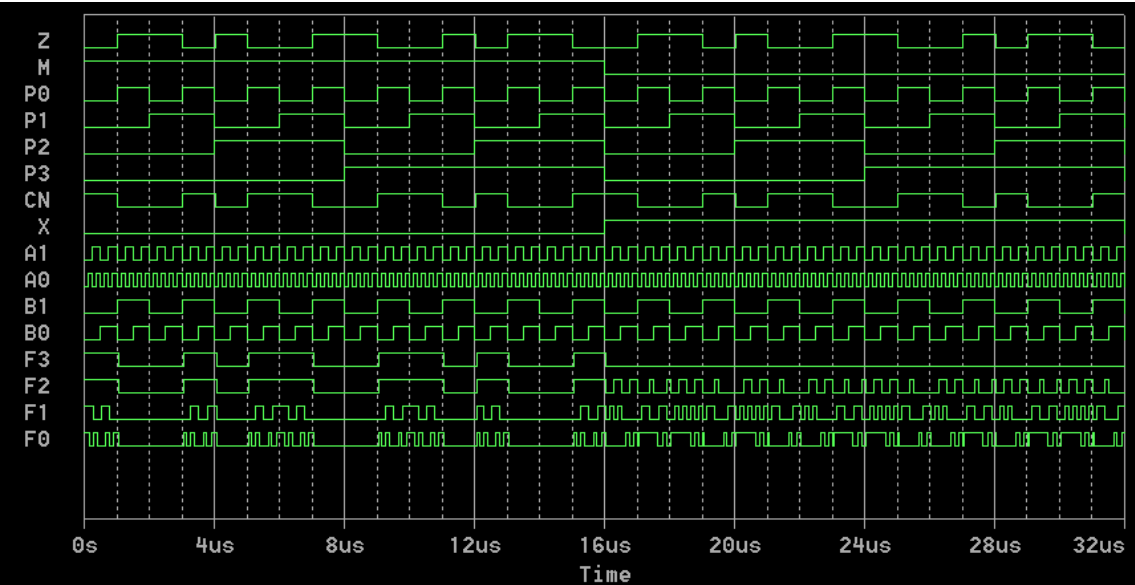


A PLUS A<sup>̄</sup>B PLUS 1

A3	A2	A1	A0	B3	B2	B1	B0	F3	F2	F1	F0	Cn
0	0	0	0	0	0	0	0	0	0	0	1	0
0	0	0	1	0	0	0	0	0	0	1	1	0
0	0	1	0	0	0	0	0	0	1	0	1	0
0	0	1	1	0	0	0	0	0	1	1	1	0
0	0	0	0	0	0	0	1	0	0	0	1	0
0	0	0	1	0	0	0	1	0	0	1	0	0
0	0	1	0	0	0	0	1	0	1	0	1	0
0	0	1	1	0	0	0	1	0	1	1	0	0
0	0	0	0	0	0	1	0	0	0	0	1	0
0	0	0	1	0	0	1	0	0	0	1	1	0
0	0	1	0	0	0	1	0	0	0	1	1	0
0	0	1	1	0	0	1	0	0	1	0	1	0
0	0	0	0	0	0	1	1	0	0	0	1	0
0	0	0	1	0	0	1	1	0	0	1	0	0
0	0	1	0	0	0	1	1	0	0	1	1	0
0	0	1	1	0	0	1	1	0	1	0	0	0

En cuanto a la tabla de verdad de puesta a 0 no se implementa, por ser demasiado trivial.

El cronograma del circuito en funcionamiento es el siguiente:



Para generar las señales, se han empleado además de los valores citados anteriormente estos otros:

A0: ONTIME = 125 ns , OFFTIME = 125 ns.

A1: ONTIME = 250 ns , OFFTIME = 250 ns.

B0: ONTIME = 500 ns , OFFTIME = 500 ns.

B1: ONTIME = 1000 ns , OFFTIME = 1000 ns.

Como se puede comprobar la tabla de verdad teórica se corresponde con la tabla de verdad práctica.

Este documento se ve complementado con los siguientes archivos adjuntos:

1. Circuito del generador de paridad : [generador de paridad.sch](#)
2. Circuito generador de señales de control: [generador de control.sch](#)
3. Circuito completo: [circuito conectado.sch](#)