

# Calculadora

Zhu Alfredo, Dan Arturo y Flores Giovanna.

{A01651980, A01650672, A01338827}@tec.mx

Instituto Tecnológico de Estudios Superiores Monterrey

**Extracto—En esta práctica se diseñó una calculadora con circuitos lógicos. Consta de 6 operaciones, suma, resta, multiplicación, división y módulo. Para ellos se ocuparon los conceptos básicos aprendidos en clases, tales como suma, resta flip flops, memorias, conversores y multiplexores. Para su realización se utilizó el software quartus y el cual fue de utilidad ya que todo fue simulado y no tuvimos que hacer conexiones reales. Los resultados fueron exitoso ya que salieron como se esperaban.**

## I. NOMENCLATURA

FPGA: dispositivo programable que contiene bloques de lógica cuya interconexión y funcionalidad puede ser configurada

DE0: tarjeta de desarrollo basado en la FPGA de la marca Altera.

Quartus II: aplicación que permite programar el DE0

MM74C922: decodificador de teclado matricial 4x4 a BCD.

## II. MARCO TEÓRICO

Una calculadora es un dispositivo que se utiliza para realizar cálculos aritméticos. Aunque las calculadoras modernas incorporan a menudo un ordenador de propósito general, se diseñan para realizar ciertas operaciones más que para ser flexibles. Por ejemplo, existen calculadoras gráficas especializadas en campos matemáticos. También suelen ser más portátiles que la mayoría de los computadores, si bien algunas PDAs tienen tamaños similares a los modelos típicos de calculadora

El impacto económico que tuvo el surgimiento de la calculadora como herramienta para el estudio y el trabajo por el desempeño que esta tiene en miles de ámbitos, si fue trascendente dentro de la sociedad que hace uso de este ya que los costos de venta y elaboración son un poco altos como para que tenga una repercusión económicamente alta dentro de las familias o empresas que hacen uso de este objeto.

## III. DESCRIPCIÓN DE ACTIVIDADES

Se marcaron como objetivos la correcta implementación de los comandos de una calculadora y utilizar todos los temas vistos en clase. Para la realización de la calculadora lo desarrollamos en 7 subtemas, suma, resta, multiplicación, división, módulo, memoria y notación polaca. Por lo que se explicara cada uno por separado. La hipótesis de la práctica es: La calculadora se puede desarrollar sin necesidad de programar.

Lo primero que se hizo fue conectar el teclado a un circuito integrado MM74C922 el cual nos permite leer las líneas y columnas arrojando un código BCD.

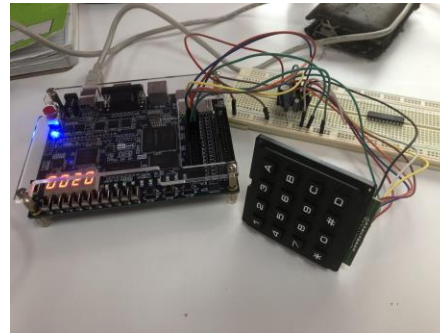


Fig. 1 Conexiones del teclado



Fig.2 Conexiones unidas

### A. Suma

Para la realización de la suma se decidió hacer en binario. Dentro de Quartus II hay circuitos integrados los cuales nos ayudan a realizar simplificar las conexiones entre compuertas.

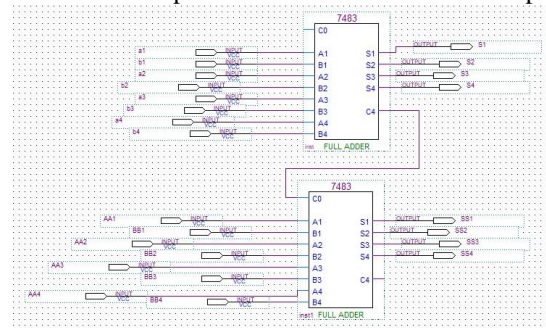


Fig.3 Diagrama de conexiones de la suma

Como se puede observar se utilizó el circuito integrado 7483, el cual permite hacer sumas binarias de cuatro bits, se interconectan los carrys para que tuvieran una relación lógica y la suma sea la adecuada. También se le conectó un decodificador bcd-binario al principio, ya que el teclado matricial nos arroja números bcd. Y al final se le conecto otro decodificador esta vez de binario a bcd.

### B. Resta

Para la implementación de la resta la hicimos por partes. Primero se hizo un comparador para saber si la suma se va a ser  $a$  menos  $b$  o  $b$  menos  $a$ . Según fuera el caso se iba a

complementar a 9 para la siguiente etapa. Ya casi para finalizar se hizo un sumador/restador bcd de cuatro bits, que cuando lo juntaba se hacía de 8 bits. Los datos resultantes se meten a multiplexor el cual es controlado por el comparador (así se sabe que resta es la que corresponde).

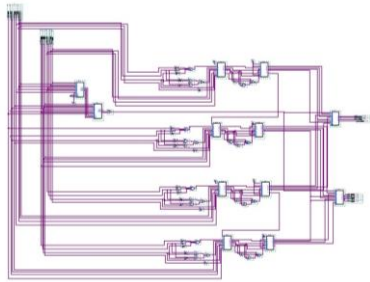


Fig 4 diagrama de circuito de la resta

### C. Multiplicación

De la misma manera se le conecta una un decodificador binario a bcd. Para implementar este caso se utilizó el árbol de wallace el cual consiste en ir sumando bit por bit con sumadores completos y medios sumadores.

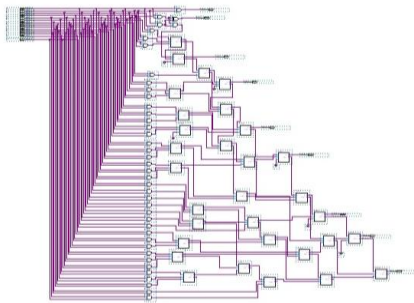


Fig. 5 Diagrama de multiplicación

### D. División

Para el caso de la división primero se metió el decodificador binario-bcd para después utilizar un comparador para de ahí determinar si se le iba a restar un cero o un uno. Este mismo proceso se repite 8 veces (ocho veces porque son ocho bits en binario). para que después todos los datos entren a un conversor de datos binario-bcd



Fig. 6 diagrama de la división

### E. Módulo

Para este caso se siguió el mismo orden que en la división, solo en lugar que de suba va a hacerse todas las restas seguidas.

### F. Memoria

Para la memoria se utilizó latches y flip flops tipo D y JK. El JK es el contador de los estados, mientras que los D nos sirve para hacer un delay y el estado anterior no influya en el siguiente.

Fig. Diagrama de la implementación de la memoria.

### G. Notación polaca

La notación polaca consta de poner los números de la siguiente manera: número, número, número, número y operando, 1205+, esto se logra gracias a la memoria la cual nos ayuda a mostrar los números en el display si que se borren. el operando no aparece en la pantalla, hace la operación directa.

Para poder decidir por qué operación iba a mostrar se decidió usar multiplexores, 16 para ser exactos. Con su ayuda las entradas de selección de datos son activados por los “números” bcd del 10 en delante del teclado.



Fig. 7 diagrama de la selección de operación

## IV. PRESENTACIÓN DE MEDICIONES, MEMORIA DE CÁLCULO Y RESULTADOS

Para poder obtener una metodología se desarrolló distintas lógicas para que pudiera concretarse.

### A. Suma

En el caso de la suma se siguió la lógica de un full-adder la cual realiza la adición de números.

Su diagrama es el siguiente:

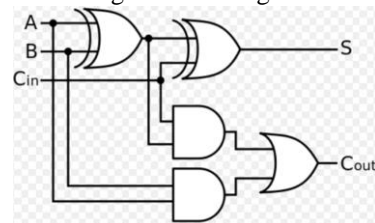


Fig. 8 Full adder

Estos se conectan en cascado de ocho bits para que nos de la suma de los ocho bits.

### B. Resta.

En este caso seguimos la siguiente lógica. Cuando los datos pasan por el comparador esto para indicarnos a cual dato es mayor, menor o si son iguales. Los datos son introducidos a dos sumador/restador bcd, donde se resta a menos b y en el otro donde se suma b menos a. Uno de los resultados está mal por lo que se decidió usar multiplexores 2 a 1 y que este fuera controlado por el comparador. Se complementaron los sustraendos a 9: para hacer esto primero se hizo la tabla de verdad, mapas de Karnaugh y la implementación del diagrama.

D	C	B	A	X	W	Y	Z
---	---	---	---	---	---	---	---

0	0	0	0	1	0	0	1
0	0	0	1	1	0	0	0
0	0	1	0	0	1	1	1
0	0	1	1	0	1	1	0
0	1	0	0	0	1	0	1
0	1	0	1	0	1	0	0
0	1	1	0	0	0	1	1
0	1	1	1	0	0	1	0
1	0	0	0	0	0	0	1
1	0	0	1	0	0	0	0
1	0	1	0	X	X	X	X
1	0	1	1	X	X	X	X
1	1	0	0	X	X	X	X
1	1	0	1	X	X	X	X
1	1	1	0	X	X	X	X
1	1	1	1	X	X	X	X

Tabla. 1 Tabla de verdad

Mapas de karnaugh y sus expresiones:

w:

		b, a			
		00	01	11	10
d, c	00	1	1	0	0
	01	0	0	0	0
	11	0	0	0	0
	10	0	0	0	0
		$\bar{d} \bar{c} \bar{b}$			

Fig.9 Mapa de Karnaugh de w y su expresión.

x:

		b, a			
		00	01	11	10
d, c	00	0	0	1	1
	01	1	1	0	0
	11	1	1	0	0
	10	0	0	1	1
		$\bar{c} b + c \bar{b}$			

Fig. 10 Mapa de Karnaugh de x y su expresión.

y:

		b, a			
		00	01	11	10
d, c	00	0	0	1	1
	01	0	0	1	1
	11	0	0	1	1
	10	0	0	1	1
		b			

Fig.11 Mapa de Karnaugh de y y su expresión.

z:

		b, a			
		00	01	11	10
d, c	00	1	0	0	1
	01	1	0	0	1
	11	1	0	0	1
	10	1	0	0	1
		a			

Fig. 12 Mapa de Karnaugh de w y su expresión.

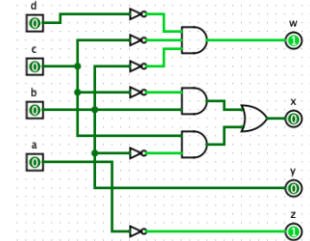


Fig. 13 Diagrama del complemento a 9.

Diagrama del sumador bcd de cuatro bits:

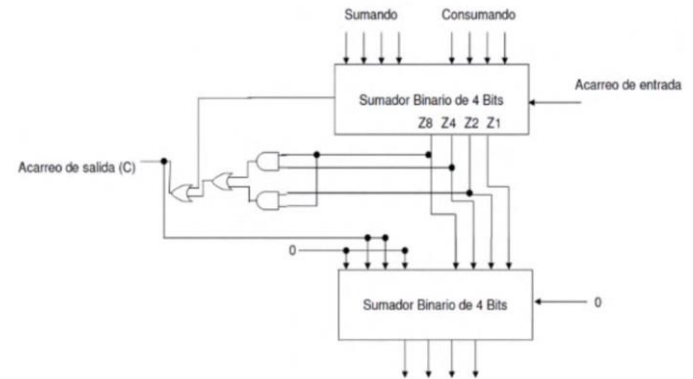


Fig.14 Diagrama sumador- restador

## C. Multiplicación

Desarrollamos el árbol de Wallace el consiste en tres pasos:

1. Multiplicar (es decir - Y) cada bit de uno de los argumentos, por cada bit de la otra, produciendo n2 resultados.

Dependiendo de la posición de los bits multiplicados, los cables llevan diferentes pesos, por ejemplo de alambre de llevar poco resultado de  $a_2b_3$  es 32. 2.Reducir el número de productos parciales a dos por capas de sumadores completos y medio. 3.Grupo de los cables en dos números, y añadirlos con un convencional sumador .

La segunda fase funciona, mientras hay tres o más cables con el mismo peso agregar una capa siguiente: 1.Tome cualquiera de los tres cables con los mismos pesos, e introducirlos en un sumador completo . El resultado será un cable de salida del mismo peso y un cable de salida con un mayor peso por cada tres cables de entrada. Si hay dos cables del mismo peso a la izquierda, introducirlos en un medio sumador. Si sólo hay un hilo izquierdo, conectarlo a la siguiente capa. Visualmente se ve de la siguiente manera:

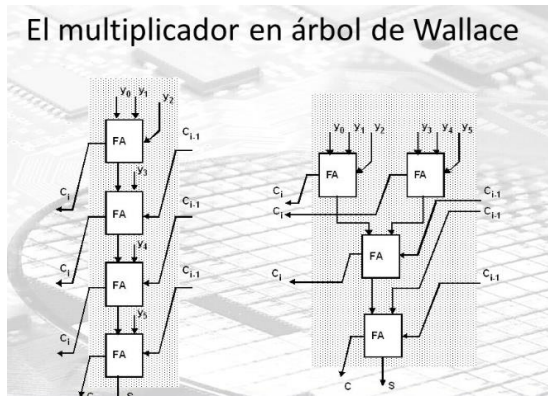


Fig 15. Diagrama de árbol de Wallace(izq. una sola rama. der. conexión de varias)

#### D. División

La máquina de estados es un conjunto de estados que sirve de intermediario en esta relación de entradas y salidas, haciendo que el historial de señales de entrada determine, para cada instante, un estado para la máquina, de forma tal que la salida depende únicamente del estado y las entradas actuales.

En este caso la lógica que seguimos:

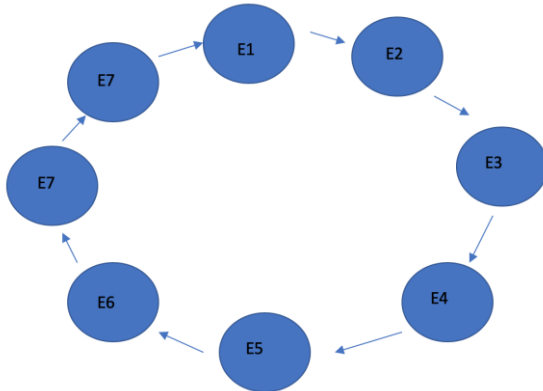


Fig. 16 diagrama de máquina de estados

La lógica que está siguiendo es la siguiente: se va a comparar el primer bit del dividendo con el divisor, si no cabe se resta un cero y se baja el siguiente bit por lo que vuelve a comparar y se repite el mismo ciclo siete veces más. En caso de que sí quepa sale un uno y se baja el siguiente bit y vuelve a entrar a un comparador y así lo hace hasta que termina el ciclo.

#### E. Módulo

El módulo se hace a consecuencia de la división. El resultado se obtiene de hacer todas las restas seguidas. Bajando el siguiente bit en cada estado.

#### F. Memoria

Los latches son un tipo de dispositivo de almacenamiento temporal de dos estados (biestable), que se suele agrupar en una categoría diferente a la de los flip-flops. Básicamente, los latches son similares a los flip-flops, ya que son también dispositivos de dos estados que pueden permanecer en cualquiera de sus dos estados gracias a su capacidad de realimentación, lo que consiste en conectar (realimentar) cada una de las salidas a la entrada opuesta. La diferencia

principal entre ambos tipos de dispositivos está en el método empleado para cambiar de estado.

Entendiendo cómo funcionan los latches se procederá explicar el funcionamiento de estos. Los flip-flops tipo JK que vemos conectados primero nos permite separar los cinco estados que se van a dar sin necesidad de que se sustituyan uno con el otro. A continuación lo que se ve conectado son los latches los cuales nos permiten que los datos sean guardados y que cuando se ingrese el operando pasen a los módulos. Mientras los flip-flops tipo D nos permite que se guarden los datos en su lugar y no se desplacen.

#### G. Notación polaca

Para la notación polaca tuvimos que hacer varias tareas por separado. La primera es que el teclado matricial muestra números de 0-9 y las otras teclas muestran combinaciones inexistentes por lo que no se van a ocupar. A partir de ahí se le van a asignar el operando que se va a realizar, esto se hace mediante un combianacion de puertas lógicas para asegurarnos que no sea un número. Después esto todos los datos entran a todos los módulos y para finalizar se metieron los todos los datos de salida a 16 multiplexores los cuales son controlados por los la combinación bcd previamente elegida.

#### V. ANÁLISIS, INTERPRETACIÓN Y DISCUSIÓN DE RESULTADOS

Los resultados tiene como fin que se hagan las operaciones sin importar lo que hay adentro. Por lo que probando cada módulo por separado funcionan a la perfección. Esto se debe a que la lógica selecciona fue previamente simulada en el programa Logisim.

Cada módulo nos permite ver las operaciones correspondientes por lo que podemos interpretar que todas las operaciones están correctas. No se tuvo ningún percance por lo que no tenemos errores que explicar.

#### VI. CONCLUSIONES

Giovanna: Considero que esta práctica fue muy compleja, y se puede notar el trabajo que hay de por medio. Sin embargo, estoy muy satisfecha con el resultado ya que además de comprobar la hipótesis y el objetivo logre aterrizar todo lo aprendido en la clase de teoría. Quartus sin duda es una herramienta que optimiza el trabajo y evita los errores de fallas ajenas a nosotros. Me sorprendí de que aunque aun me falta bastante por ver, con lo que llevo ya se pueden desarrollar grandes cosas.

Arturo: Hacer una calculadora fue una experiencia académica única. Me permitió profundizar en los temas que ya había aprendido en clase. Me gusto desarrollar un proyecto desde cero ya que no hay un instructivo como tal de como armarla por lo que puede explotar mis capacidades de diseñador. El trabajo realizado por mi equipo y yo fue productivo ya que logramos resolver las adversidades que tuvimos. Se cumplieron todos los objetivos y me sorprendió

poder comprobar la hipótesis ya que al principio se veía casi imposible.

Alfredo: Esta práctica final me sirvió como repaso general de los temas adquiridos durante todo el curso de Sistemas Digitales. Logre comprender la lógica que conlleva un calculadora y las diferentes variantes que hay para solucionar un mismo problema. Considero que fue una práctica retadora porque me permitió desarrollar habilidades que en el lado teórico no hubiera podido hacer. También considero que Quartus fue una herramienta que me dio la capacidad de hacer circuitos muy grandes de manera digital ahorrándome errores de eléctricos como malas conexiones, cables sin continuidad o circuitos integrados quemados. Considero que los métodos que utilizamos fueron los mejores ya que todo funcionó correctamente. Los objetivos e hipótesis se cumplieron satisfactoriamente.

### Referencias

#### *Libros:*

- [1] A. Rodrigo, "FPGA : Nociones básicas e implementación," 2004.
- [2] S. M. Trimberger, "Three Ages of FPGAs : A Retrospective on the First Thirty Years of FPGA Technology," *Proc. IEEE*, vol. 103, no. 3, pp. 318–331, 2015.
- [3] Terasic Technologies. "DE0 User Manual", 2011.

#### *Artículo:*

- [4] M. Melgarejo, "Diseño e implementación de un controlador difuso basado en fpga,," no. 1, pp. 1–10.