



INSTITUTO POLITÉCNICO NACIONAL



ESCUELA SUPERIOR DE CÓMPUTO

INGENIERIA EN SISTEMAS COMPUTACIONALES

MATERIA: FUNDAMENTOS DE DISEÑO DIGITAL

PRESENTA:

RAMIREZ BENITEZ BRAYAN

GRUPO: 2CM1

PRÁCTICA No. 2

Uso de Galaxy

ESTADO DE MEXICO NOVIEMBRE 2020

# 1. INTRODUCCIÓN TEÓRICA:

## 1.1 Las herramientas CAD-EDA.

En su sentido más moderno, CAD (diseño asistido por ordenador, del inglés Computer Aided Design) significa proceso de diseño que emplea sofisticadas técnicas gráficas de ordenador, apoyadas en paquetes de software para ayuda en los problemas analíticos, de desarrollo, de coste y ergonómicos asociados con el trabajo de diseño. En principio, el CAD es un término asociado al dibujo como parte principal del proceso de diseño, sin embargo, dado que el diseño incluye otras fases, el término CAD se emplea tanto como para el dibujo, o diseño gráfico, como para el resto de herramientas que ayudan al diseño (como la comprobación de funcionamiento, análisis de costes, etc.)

EDA (Electronic Design Automation) es el nombre que se le da a todas las herramientas (tanto hardware como software) para la ayuda al diseño de sistemas electrónicos. Dentro del EDA, las herramientas de CAD juegan un importante papel, sin embargo, no sólo el software es importante, workstations cada día más veloces, elementos de entrada de diseño cada vez más sofisticados, etc. son también elementos que ayudan a facilitar el diseño de circuitos electrónicos.

## 1.2 Lenguajes de descripción de hardware (HDL)

Un lenguaje de descripción de *hardware* (HDL, *hardware description language*) es un lenguaje de programación especializado que se utiliza para definir la estructura, diseño y operación de circuitos electrónicos, y más comúnmente, de circuitos electrónicos digitales, como el convertidor analógico-digital o cualquier antena satelital. Así, los lenguajes de descripción de hardware hacen posible una descripción formal de un circuito electrónico, y posibilitan su análisis automático y su simulación.

Los lenguajes de descripción de hardware se parecen mucho a otros lenguajes de programación de ordenadores tales como el C o Java: básicamente consisten en una descripción textual con expresiones, declaraciones y estructuras de control. Sin embargo, una importante diferencia entre los HDL y otros lenguajes de programación está en que el HDL incluye explícitamente la noción de tiempo.

## 1.3 Los dispositivos lógicos programables (PLD's)

Un Dispositivo Lógico Programable (PLD) es un componente electrónico usado para construir circuitos digitales reconfigurables. A diferencia de una compuerta lógica que tiene una función fija, los PLDs salen de fábrica sin una función en específico,

por lo tanto, necesitan ser programados o reconfigurados antes de poder ser usados. Los PLDs tienen varias ventajas. La primera es la habilidad de integración, que permite integrar una gran cantidad de funcionalidad en un solo chip. Los PLDs eliminan el uso de múltiples chips, así como la inconveniencia y desconfianza de usar cableado externo. La segunda ventaja es el hecho de poder cambiar el diseño. Muchos PLDs permiten ser reprogramados o reconfigurados. Existen dos ramas principales dentro de los dispositivos lógicos programables, la lógica programable de campo y la de fábrica. El término campo en este contexto implica que los dispositivos puedan ser programados en el “campo” del usuario, mientras que la lógica de fábrica puede ser programada en la misma fábrica donde se construyen, de acuerdo a los requerimientos del cliente. En este caso, la tecnología de programación usa procesos irreversibles, por lo que solo es posible hacerlo una vez.

#### 1.4 La GAL22V10

Arreglo lógico programable y borrable. El GAL22V10, en el tiempo de retardo máximo de propagación 4ns, combina un proceso CMOS de alto rendimiento con borrrable eléctricamente flotando tecnología de compuerta para proporcionar el máximo rendimiento disponible de cualquier dispositivo 22V10 en el mercado. Circuitos CMOS permite el GAL22V10 a consumir mucha menos energía en comparación con bipolares 22V10 dispositivos. E2 tecnología ofrece alta velocidad (<100 ns) Borrar tiempos, proporcionando la capacidad de reprogramar o reconfigurar el dispositivo de forma rápida y eficiente. La arquitectura genérica proporciona máxima flexibilidad de diseño por permitiendo la salida lógica macro célula (OLMC para ser configurado por el usuario.

#### 1.5 Herramientas de desarrollo

El entorno de desarrollo ISE de Xilinx posee un aspecto similar al de los entornos de programación actuales como puede ser Visual Basic o Visual C, es decir, posee diversas ventanas para visualización de tareas específicas sobre cada una de ellas. En este caso existen cuatro tipos de ventanas

1.Ventana de ficheros fuente. En esta ventana se muestran los ficheros fuente utilizados en el diseño y las dependencias entre ellos. También es aquí donde se elige el tipo de dispositivo donde se desea implementar el diseño. Esta ventana posee diversas solapas para visualizar diferentes tipos de información relativa a las fuentes de diseño empleadas.

2.Ventana de Procesos. Esta ventana muestra todos los procesos necesarios para la ejecución de cada etapa de diseño. La lista de procesos se modifica dinámicamente dependiendo del tipo de fuente seleccionado en la ventana de ficheros fuente.

3.Ventanas de edición. Al hacer doble clic sobre un fichero fuente de la ventana de ficheros fuente se abre una ventana de edición para modificar el fichero (en caso de lenguaje HDL), o bien se ejecuta el programa que permite editar el diseño (en caso de diseños esquemáticos ó máquinas de estado).

4.Ventana de información, situada en la parte inferior. Muestra mensajes de error, aviso o información emitidos por la ejecución de los programas de compilación, implementación, etc.

Quartus II es una herramienta de software producida por Altera para el análisis y la síntesis de diseños realizados en HDL.

Permite al desarrollador compilar sus diseños, realizar análisis temporales, examinar diagramas RTL y configurar el dispositivo de destino con el programador.

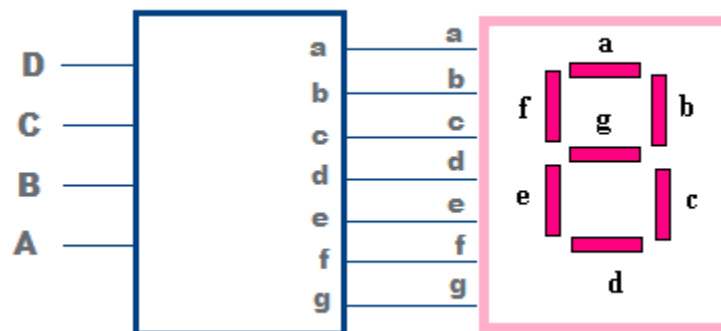
Con el programa de diseño Quartus II los diseñadores pueden usar los dispositivos HardCopy Stratix de manera que puede prever y verificar su rendimiento, el cual resulta en promedio un 1050 por ciento más rápido que su FPGA equivalente. Además, en el flujo de diseño del HardCopy Stratix, Quartus II incluye una serie de utilidades que reducen el tiempo de diseño. Como contraste adicional el bajo precio del Quartus II en comparación con otras herramientas de diseño de ASIC.

USB Blaster programador para dispositivos FPGA. El blaster es un dispositivo que funciona para poder cargar el código realizado para tu FPGA por JTAG, AS y PS.

## 2. DESARROLLO

Problema

Diagrama a bloque del decodificador



Entradas			Segmentos del Display (Salidas)							Número (Boleta)
E3	E2	E1	A	B	C	D	E	F	G	
0	0	0	0	0	0	0	0	0	0	8
0	0	1	0	0	0	1	1	0	0	9
0	1	0	0	1	0	0	1	0	0	5
0	1	1	0	1	0	0	0	0	0	6
1	0	0	0	0	0	0	0	0	1	0
1	0	1	1	0	0	1	1	1	1	1
1	1	0	0	0	0	1	1	1	1	7
1	1	1	1	0	0	1	1	1	1	1

Simulaciones

Name	Value	Sti...	
+	e	0	<... 0
+	display	00	00

Stimulators

SignalsHotkeysPredefined

Signals:

Name	Type
<input checked="" type="checkbox"/> e	<= 000

Stimulator type:

Value

Strength:

Override

Force value:

000

☐ Display paths

Apply

Close

Name	Value	Sti...	
+	e	2	0 2
+	display	24	00 24

Stimulators

SignalsHotkeysPredefined

Signals:

Name	Type
<input checked="" type="checkbox"/> e	<= 010

Stimulator type:

Value

Strength:

Override

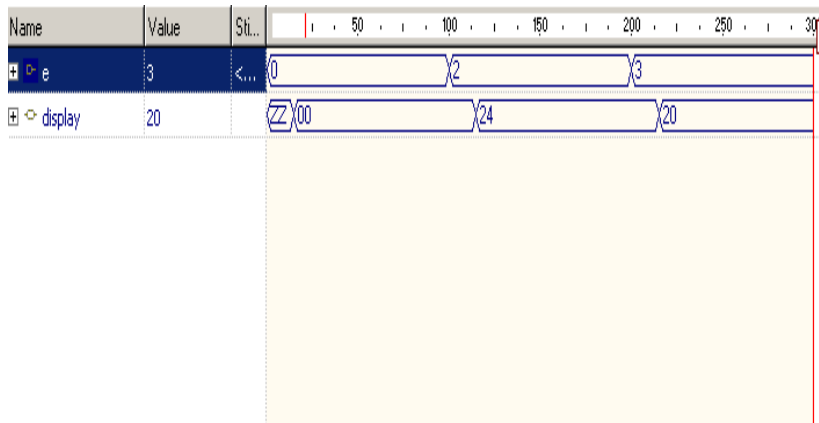
Force value:

010

☐ Display paths

Apply

Close



**Stimulators**

Signals Hotkeys Predefined

Signals: Stimulator type: Strength:

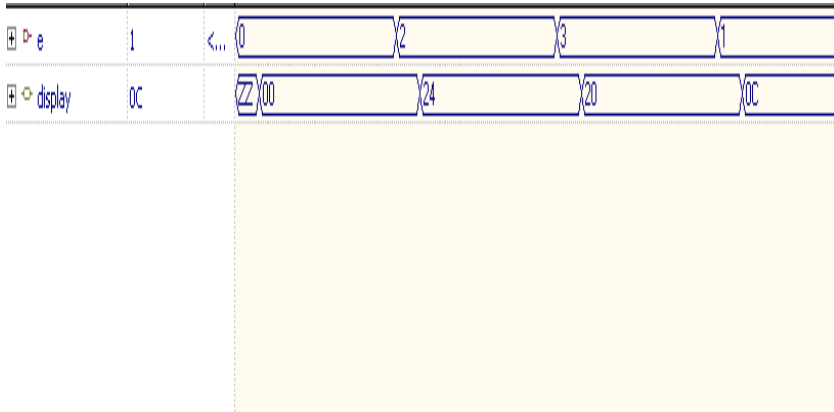
Name	Type
e	<= 011

Force value: 011

☐ Display paths

Apply

Close



**Stimulators**

Signals Hotkeys Predefined

Signals: Stimulator type: Strength:

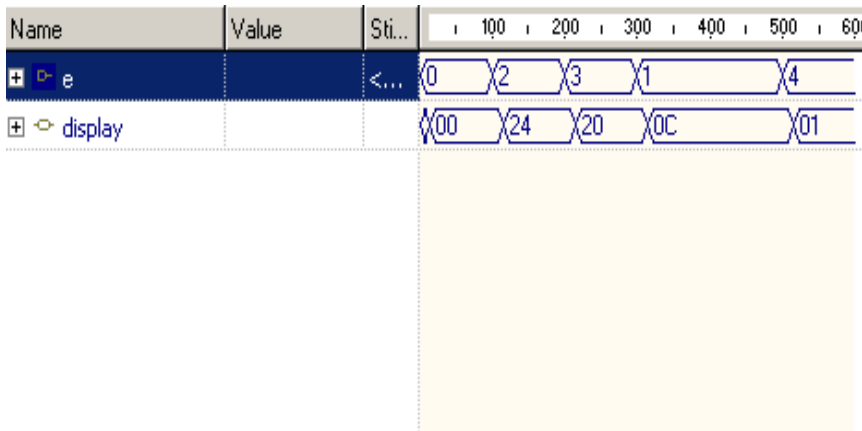
Name	Type
e	<= 001

Force value: 001

☐ Display paths

Apply

Close



**Stimulators**

Signals Hotkeys Predefined

Signals: Stimulator type: Strength:

Name	Type
e	<= 100

Force value: 100

☐ Display paths

Apply

Close

Name	Value	Sti...	100	200	300	400	500	600	70
e	5	<...	0	2	3	1	4	5	
display	4F		00	24	20	0C	01	4F	

Stimulators

Signals

Hotkeys

Predefined

Signals:

Name	Type
<input checked="" type="checkbox"/> e	<= 101

Stimulator type:

Value

Strength:

Override

Force value:

101

☐ Display paths

Apply

Close

Name	Value	Sti...	100	200	300	400	500	600	700	80
e	6	<...	0	2	3	1	4	5	6	
display	0F		00	24	20	0C	01	4F	0F	

Stimulators

Signals

Hotkeys

Predefined

Signals:

Name	Type
<input checked="" type="checkbox"/> e	<= 110

Stimulator type:

Value

Strength:

Override

Force value:

110

☐ Display paths

Apply

Close

C22V10

e(2) =	1	24  * not used
e(1) =	2	23  = display(2)
e(0) =	3	22  = display(6)
not used *	4	21  = display(4)
not used *	5	20  * not used
not used *	6	19  * not used
not used *	7	18  * not used
not used *	8	17  = display(0)
not used *	9	16  = display(5)
not used *	10	15  = display(1)
not used *	11	14  = display(3)
not used *	12	13  * not used

```

:y:
Error Count = 0      Warning Count = 0
...

```

-----

Description	Used	Max
Dedicated Inputs	2	11
Clock/Inputs	1	1
I/O Macrocells	7	10
<hr/>		
	10 /	22 = 45 %

Information: Output Logic Product Term Utilization.

Node#	Output Signal Name	Used	Max
14	display(3)	2	8
15	display(1)	2	10
16	display(5)	1	12
17	display(0)	1	14
18	Unused	0	16
19	Unused	0	16
20	Unused	0	14
21	display(4)	1	12
22	display(6)	1	10
23	display(2)	2	8
25	Unused	0	1
<hr/>			
	10 /	121	= 8 %

C22V10		
e(2) =  1		24  * not used
e(1) =  2		23  * not used
e(0) =  3		22  * not used
not used *  4		21  * not used
not used *  5		20 = display(6)
not used *  6		19 = display(5)
not used *  7		18 = display(4)
not used *  8		17 = display(3)
not used *  9		16 = display(2)
not used * 10		15 = display(1)
not used * 11		14 = display(0)
not used * 12		13  * not used

Summary:  
Error Count = 0      Warning Count = 0



## Referencias

<http://jeuazarru.com/wp-content/uploads/2014/10/VHDL.pdf>

[https://www.dte.us.es/docencia/master/micr/dapa/temas/tema\\_02/tema2\\_ldh.pdf](https://www.dte.us.es/docencia/master/micr/dapa/temas/tema_02/tema2_ldh.pdf)

<http://www.ptolomeo.unam.mx:8080/jspui/bitstream/132.248.52.100/658/8/A8.pdf>

<https://www.uv.es/rosado/courses/dcse/prac1XilinxISEintrod.pdf>

<http://ygelectronicaintegrada.blogspot.com/2008/02/sistema-de-control-utilizando-altera.html>