**Instituto Politécnico Nacional**

**Escuela Superior de Cómputo**

*Fundamentos de Diseño Digital*

Práctica 4: Uso del ISPLEVER Classic Lattice 2.0.

Grupo: 2CM6

Miembros:

Alfredo Pérez Quiñonez

José Emiliano Pérez Garduño

Maestro:

Carlos Jesús Pastrana Fernández

Día de práctica: 29 / Marzo / 2017

Día de entrega: 12 / Marzo / 2017

1. **Objetivo General:**

Aprenderá a programar un dispositivo lógico programable (PLD), para esto estudiará y aprenderá a utilizar las herramientas adecuadas para lograr dicho objetivo. Así mismo, reafirmará el conocimiento adquirido en clase al realizar un programa que ejecute una función específica dependiendo del programa desarrollado.

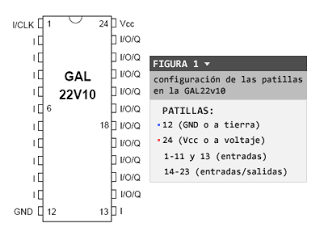
Finalmente, sabrá lo que es un GAL y como programar una función específica sobre el dispositivo, utilizando un lenguaje de descripción de hardware (VHDL).

1. **Objetivos Específicos:**
2. Distinguir los beneficios proporcionados por las herramientas CAD.
3. Confirmar el conocimiento adquirido en clase y realizar un programa en VHDL que se ejecute sobre un GAL.
4. Comprobar físicamente el correcto funcionamiento del sistema diseñado, el cual será implementando sobre un protoboard.
5. **Material y equipo empleado:**

* Programador universal
* Fuente de 5V
* 1 GAL22V10
* DIP switch de 4
* 3 Resistencias de 1KΩ
* 7 Resistencias de 220Ω
* 8 Leds

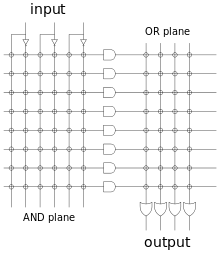
1. **Introducción teórica:**

GAL: (Generic Array Logic) (Matriz lógica genérica) Desarrollada por “Lattice Semiconductor” en 1985. Este dispositivo tiene las mismas propiedades lógicas que un PAL, pero puede ser borrado y reprogramado. La GAL es muy útil en la fase de prototipo de un diseño, cuando un fallo en la lógica puede ser corregido por reprogramación. Las GAL se programan y reprograman utilizando el programa o programador PAL, o utilizando la técnica de programación circuital en chips secundarios.

Una GAL permite implementar cualquier expresión en suma de productos con un número de variables definidas. El proceso de programación consiste en activar o desactivar cada celda E2CMOS con el objetivo de aplicar la combinación adecuada de variables a cada compuerta AND y obtener la suma de productos. Las celdas E2CMOS activadas conectan las variables deseadas o sus complementos con las apropiadas entradas de las puertas AND. Las celdas E2CMOS están desactivadas cuando una variable o su complemento no se utiliza en un determinado producto. La salida final de la puerta OR es una suma de productos. Cada fila está conectada a la entrada de una puerta AND, y cada columna a una variable de entrada o a su complemento. Mediante la programación se activa o desactiva cada celda E2CMOS, y se puede aplicar cualquier combinación de variables de entrada, o sus complementos, a una puerta AND para generar cualquier operación producto que se desee. Una celda activada conecta de forma efectiva su correspondiente fila y columna, y una celda desactivada desconecta la fila y la columna. Celdas se pueden borrar y reprogramar eléctricamente. Una celda E2CMOS típica puede mantener el estado en que se ha programado durante 20 años o más. 



PAL: Una matriz lógica programable o PAL es un dispositivo diseñado por la empresa “Monolithic Memories” y ha sido el circuito programable más utilizado durante décadas. La estructura de una PAL, derivada de las PLA, se basa en que cualquier función lógica puede ser descrita por sumas de productos lógicos. Un dispositivo PAL tiene una matriz de entrada formada por una puerta AND conectada a una serie de puertas OR, con las salidas conectadas a un pin del dispositivo. La matriz de puertas AND es programable y, por tanto, permite efectuar el producto lógico con cualquier combinación de las entradas primarias.



PLD: En electrónica y computación la Lógica programada es un tipo de diseño implementado en chips que permite la reconfiguración de los circuitos con el simple cambio del software que incorpora, es lo contrario de la lógica cableada. La lógica programada se basa en dispositivos lógicos programables (PLD), los cuales tienen una función no establecida, al contrario que las puertas lógicas que tienen una función fija en el momento de su fabricación. Antes de poder utilizar el PLD en un circuito, este debe ser programado.

En 1970, Texas Instruments desarrolló un CI de máscara programable basado en la memoria asociativa de sólo lectura (ROAM) de IBM. Este dispositivo, el TMS2000, era programado alterando la capa metálica durante la producción del CI. El TMS2000 tenía hasta 17 entradas y 18 salidas con 8 biestables JK como memoria. Texas Instruments acuño el término “Programmabble Logic Array” para este dispositivo. En 1973 “National Semiconductor” introdujo un dispositivo PAL de máscara programable (DM7575) con 14 entradas y 8 salidas sin registros de memoria.

En 1971, General Electric desarrollaba un PLD basado en la nueva tecnología PROM. Este dispositivo experimental mejoró el ROAM de IBM permitiéndole realizar lógica multinivel. Intel acababa de introducir la PROM de puerta flotante borrable por UV por lo que los desarrolladores en General Electric incorporaron esa tecnología. El dispositivo de General Electric era el primer PLD jamás desarrollado, antecesora del EPLD de Altera en una década.

VHDL es un lenguaje definido por el IEEE (Institute of Electrical and Electronics Engineers) usado por ingenieros y científicos para describir circuitos digitales o modelar fenómenos científicos respectivamente. VHDL es el acrónimo que representa la combinación de VHSIC y HDL, donde VHSIC es el acrónimo de Very High Speed Integrated Circuit y HDL es a su vez el acrónimo de Hardware Description Language. Para el modelado físico existe la derivación del lenguaje VHDL-AMS. Originalmente, el lenguaje VHDL fue desarrollado por el departamento de defensa de los Estados Unidos a inicios de los años 80's basado en ADA, con el fin de realizar simulación de circuitos eléctricos digitales; sin embargo, posteriormente se desarrollaron las herramientas de síntesis e implementación en hardware a partir de los archivos .VHD. Aunque puede ser usado de forma general para describir cualquier circuito digital se usa principalmente para programar PLD, FPGA (Field Programmable Gate Array), ASIC y similares.

1. **Procedimiento de la práctica:**

1.- Realice el ejercicio de ejemplo hecho paso a paso incluyendo señales de prueba para comprobar su funcionamiento, así como aprender a manejar el software (tome capturas de pantalla) una vez realizado programe su GAL y compruebe su funcionamiento de forma práctica.

**Edición, Simulación y Síntesis con ispLEVER**

Ejecutar ispLEVER Classic 2.0 para comenzar a usarlo:

En la opción de menú **File**, se selecciona la opción **Nuevo Proyecto**.

Se asigna un nombre al proyecto y un subdirectorio para contener los archivos del proyecto.

Sólo para este ejemplo se llamará (compuerta) y el subdirectorio será: (C:\tmp\ISPLever\)

Se selecciona el tipo de diseño y las herramientas de síntesis y de simulación:

1.-VHDL

2.-Sinplify

3.-Active-HDL

Y se continua con; Siguiente>

En la siguiente ventana lo más importante es seleccionar el checkbox de “Mostrar dispositivos obsoletos”, esto permitirá visualizar la Familia de Dispositivos GAL, de la cual se podrá seleccionar el dispositivo GAL22V10D (para este caso).

Por el tipo de encapsulado se debe seleccionar: 24PDIP

Condiciones de operación: Comercial.

Potencia: Baja(Low)

Y se verifica que el número de parte sea: GAL22V10D

Se continua con: Siguiente>

Como no se tiene ningún archivo que agregar, simplemente se selecciona: Siguiente>

En la siguiente ventana se presentará un resumen de las configuraciones seleccionadas para crear el nuevo proyecto, en esta ventana solo se debe seleccionar el botón: Finalizar.

En la ventana del navegador de ispLEVER aparecerá el proyecto creado con los documentos fuente del proyecto y los procesos que pueden ser aplicados a los documentos fuente.

En el Frame inferior del navegador de ispLEVER aparecerá el proyecto creado con los documentos fuente del proyecto y los procesos que pueden ser aplicados a los documentos fuente.

En el Frame inferior aparecerá una bitácora de los procesos ejecutados con el software.

En la opción de menú Source se selecciona la opción New para adicionar un nuevo archivo de código.

En este ejemplo el nuevo código fuente corresponderá a un módulo en VHDL: Se selecciona: VHDL Module y OK.

En la ventana de nuevo archivo fuente VHDL, el archivo, la entidad y la arquitectura deberán tener un nombre, para este ejemplo:

Nombre de archivo: CompuertaY

Entidad: entY

Arquitectura: arqY

Y se selecciona: OK

En el editor de texto aparecerá la estructura básica (y para este laboratorio “Más que suficiente”) del archivo de descripción de hardware del proyecto:

En este punto sólo resta agregar la descripción del sistema que se desea implementar y se guarda en el archivo fuente.

Antes de simular, en el Frame de procesos para el archivo fuente actual, se debe ejecutar el proceso: Generate Schematic Symbol.

Primero se debe seleccionar el componente GAL22V10D-10LP con el botón derecho y se selecciona la opción: New

Esto abre la ventana de nuevo archivo fuente en la que se selecciona la opción: Waveform Stimulus y después: OK.

Cuando se abre la ventana del editor de formas de onda Waveform Editor se abren dos ventanas adicionales:

Nada Seleccionado (Nothing Selected)

Nuevos archivos de estímulos de forma de onda (New Waveform Stimulus).

Primero se asigna un nombre al nuevo archivo: stimY (para este ejemplo) y a continuación se selecciona: OK.

Ahora se debe iniciar la entidad con los puertos que se desea simular seleccionando la opción: Import Wave… de la opción de menú: Edit.

Cuando se abre la ventana Import se seleccionan los puertos (señales) de la entidad que serán incluidos en la ventana Waveform Editor y se selecciona el botón Show.

En el Editor de Formas de Onda aparecen los puertos de la entidad seleccionados con una letra I/O que indican cuando son puertos de Entrada o Salida.

La ventana Nothing Selected se puede mover para que no interfiera con las formas de onda que se desea simular.

Para editar las formas de onda de los puertos de entrada se puede simplemente hacer doble click justo al lado derecho del puerto y desplazar el cursor hasta la posición que se necesite para simular.

Cuando el estado que se agrega es solamente Estado Alto (High) se puede seleccionar el periodo de tiempo que se desea en Estado Bajo (Low)

Se hace lo mismo con el resto de los puertos de entrada y se actualiza el archivo “Salvando” las formas de Onda ya sea seleccionando Save de la opción de Menú File o seleccionando el botón de Diskette de la barra de herramientas:

En el Navegador de ispLEVER Classic se selecciona el archivo de Formas de Onda con extensión. wdl para que aparezca el proceso de Simulación Funcional (Functional Simulation) que deberá ejecutarse para realizar la simulación:

Cuando termina de ejecutarse el proceso de simulación funcional aparece una ventana de Panel de Control de Simulador (Simulator Control Panel) que indica que la simulación fue exitosa.

**Nota: En caso de que se detenga el proceso a causa de algún error, éste deberá buscarse en el Frame inferior del Navegador de ispLEVER en la Bitácora de Construcción Automática (Automake Log) y se deberá hacer doble click sobre la línea en rojo para que el Editor de Texto nos ubique en la línea de código donde se presenta el error que deberá ser corregido.**

En esta ventana de Panel de Control de Simulador se selecciona el botón de Signo de Admiración (Run) para poder Analizar el resultado de la Simulación en la ventana de Visualizador de Forma de Onda (Waveform View):

Si se desea cambiar los estímulos de entrada se debe modificar y salvar la Forma de Onda correspondiente en el Editor de Forma de Onda.

Cuando en el Editor de Forma de Onda se ha seleccionado un intervalo que comprende diversos estados diferentes, la ventana de Nothing Selected cambia por Selected Interval in Pattern bit en la que se podrá seleccionar el nuevo estado para el intervalo seleccionado.

Se ejecuta de nuevo el proceso de Simulación Funcional en el Navegador de ispLEVER y se espera a que se actualice el estado en el Panel de Control de Simulador.

Se selecciona el botón Run de la barra de herramientas y se Analiza el nuevo resultado de la simulación en el Visualizador de Forma de Onda.

**Síntesis**

Una vez que se esté seguro de que la simulación es correcta, se debe generar el Mapa de Fusibles y el Archivo JEDEC para programar el dispositivo.

Para esto se selecciona el Archivo Fuente GAL22V10D-P en el Navegador de ispLEVER:

Y se selecciona el Proceso: Crear el Mapa de Fusibles (Create Fuse Map)

En la opción de menú Window del navegador de ispLEVER se selecciona la opción Report Viewer y se abre el archivo con extensión: .RPT

En esta ventana se podrá ver las ecuaciones y la asignación de pines (Terminales/puertos)

En la misma opción de menú Window se puede seleccionar la opción Text Editor y abrir el archivo JEDEC con extensión .JED para analizar el Mapa de Fusibles.

2.-Diseñar un circuito que incluya las compuertas básicas And, Or, Exor, Nand y Nor de tres entradas llamadas A, B y C, implementados en un dispositivo programable GAL usando el programa de captura y obtener físicamente la tabla de Verdad de cada operador, realícelo de dos formas distintas.

1. Mediante componentes (primero realice el componente).
2. De forma directa a través de la reducción de la tabla de verdad.

Implemente los circuitos en el protoboard siguiente el diagrama obtenido en el archivo reporte como lo indica la figura y compruebe la tabla de verdad:

**Nota: La distribución de terminales es asignada en forma aleatoria por el programa, puede ser que su resultado sea diferente distribución obtenida a lo presentado en este dibujo (el ejemplo anterior está programado en una GAL16V8 solo como ejemplo).**

Obtenga la tabla de verdad de los valores para cada una de las salidas.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| m | A B C | And | Or | Exor | Nand | Nor |
| 0 | 0 0 0 | 0 | 0 | 0 | 1 | 1 |
| 1 | 0 0 1 | 0 | 1 | 1 | 1 | 0 |
| 2 | 0 1 0 | 0 | 1 | 1 | 1 | 0 |
| 3 | 0 1 1 | 0 | 1 | 0 | 1 | 0 |
| 4 | 1 0 0 | 0 | 1 | 1 | 1 | 0 |
| 5 | 1 0 1 | 0 | 1 | 0 | 1 | 0 |
| 6 | 1 1 0 | 0 | 1 | 0 | 1 | 0 |
| 7 | 1 1 1 | 1 | 1 | 1 | 0 | 0 |

*Recomendaciones:*

1. *Tenga cuidado al insertar correctamente el circuito integrado en el programador, el colocarlo en forma diferente a lo especificado puede dañar el dispositivo.*
2. *Es recomendable que al programar el circuito primero seleccione el circuito, borre su contenido y posteriormente cargue el archivo JEDEC y por ultimo programe el dispositivo.*

**Cuestionario:**

* ¿Cuál es el significado de la palabra GAL?
* ¿Cuántas entradas máximo puede tener el de GAL22V10?
* ¿Cuántas salidas máximo puede tener el de GAL22V10?
* ¿Cuál es el significado de JEDEC?
* Calcule el número de circuitos integrados TTL con compuertas de 2 de cada tipo que se requieren para implementar esta práctica.

Una vez hechas las modificaciones correctamente, se realiza una nueva compilación y después se procede a revisar el archivo de salida (deco.rpt). Como se puede ver en la figura 13, el propósito se cumplió. De esta manera ya se puede programar el dispositivo.

**PROGRAMACIÓN DEL DISPOSITIVO:** Para descargar el programa al dispositivo es necesario utilizar un programador. Para esto se requiere portar el archivo que se desea programar, el cual tiene terminación JEDEC (deco.jed), e ir al programador del laboratorio. En el laboratorio se tiene un programador con un software para realizar dicho propósito. En el software se selecciona el archivo e indica a que dispositivo se va a descargar, para esto se ha insertado el dispositivo en la base del programador y se procede a programarlo.

**Comprobación física:** Esta última etapa consiste en realizar la prueba física al circuito y corroborar que efectivamente hace la función para la cual fue programado.

***(Nota:******Dibujar como se interconectó el circuito realmente)***

1. **Observaciones y conclusiones:**

*José Emiliano Pérez Garduño:* Al terminar esta práctica aprendí a usar el VHDL y como se aplica a la GAL 22V10, ya que, si pudimos hacer la mayor parte el primer día y entendimos como usar muy bien el programa, por lo que la próxima vez que lo usemos no nos costará tanto trabajo y será de forma más eficiente.

*Alfredo Pérez Quiñonez:*

1. **Bibliografía:**

<https://books.google.com.mx/books?id=6mxHXqXKdFMC&pg=PA97&redir_esc=y#v=onepage&q&f=false>

<http://www.swtpc.com/mholley/PLD_History/ABEL_project/CUPL_Data_Sheet_1983_ocr.pdf>

<http://www.circuitoselectronicos.org/2009/01/implementacin-de-las-compuertas-lgicas.html>