**Instituto Politécnico Nacional**

**Escuela Superior de Cómputo**

*Fundamentos de Diseño Digital*

Proyecto de Segundo Parcial

ASL y Números Primos

Grupo: 2CM6

Equipo 12

Miembros:

Alfredo Pérez Quiñonez

José Emiliano Pérez Garduño

Maestro:

Carlos Jesús Pastrana Fernández

Día de entrega: 3 / Mayo / 2017

1. **Objetivo General:**

Al termino del proyecto los alumnos sabrán programar una GAL22V10 para poder detectar si un número es primo y podrán realizar un ASL.

1. **Introducción teórica:**

* ASL: En programación computacional, un shift aritmético es un operador de shift. Los dos tipos básicos son izquierdos y derechos.

Para números binarios es una operación en dirección contraria a la que se escriben los bits, ya que funciona de manera que toma el bit de cada extremo y lo desplaza al extremo contrario, cambiando el número que se tenía.

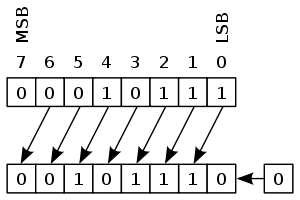


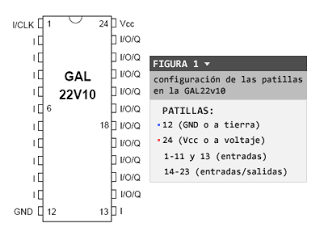
Ilustración 1 Arithmetic shift left

* Números Primos: En matemáticas, un número primo es un número natural mayor que 1 que tiene únicamente dos divisores distintos: él mismo y el 1.

Por el contrario, los números compuestos son los números naturales que tienen algún divisor natural aparte de sí mismos y del 1 y por lo tanto, pueden factorizarse. El número 1, por convenio, no se considera ni primo ni compuesto.

Los 54 números primos menores de 255 son: 2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, 41, 43, 47, 53, 59, 61, 67, 71, 73, 79, 83, 89, 97, 101, 103, 107, 109, 113, 127, 131, 137, 139, 149, 151, 157, 163, 167, 173, 179, 181, 191, 193, 197, 199, 211, 223, 227, 229, 233, 239, 241, 251.

* VHDL: 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.
* 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. 



1. **Material y equipo empleado:**

* 1 circuito integrado GAL22V10
* 15 LEDS de colores
* 15 resistores de 330Ω
* 10 resistores de 1KΩ
* 1 Dip switch de 8
* Alambre telefónico
* 1 tablilla de prueba (protoboard)
* 1 pinza de punta
* 1 pinza de corte
* Cables banana-caimán (para alimentar el circuito)

1. **Equipo Empleado:**

* Multímetro
* Fuente de alimentación de 5 volts
* Manual de Motorola

1. **Desarrollo experimental y actividades:**

Se desarrollará un programa en VHDL para comprobar si un número es primo o no, por medio de un LED indicador.

Se desarrollará un programa en VHDL para realizar el ASL, usamos una concatenación de forma que se recorriera el bit menos significativo (que en este caso por ser un vector de 8 bits sería el séptimo bit) a la izquierda y se agregará un cero para reemplazarlo y de manera consecutiva se obtenía el ASL.

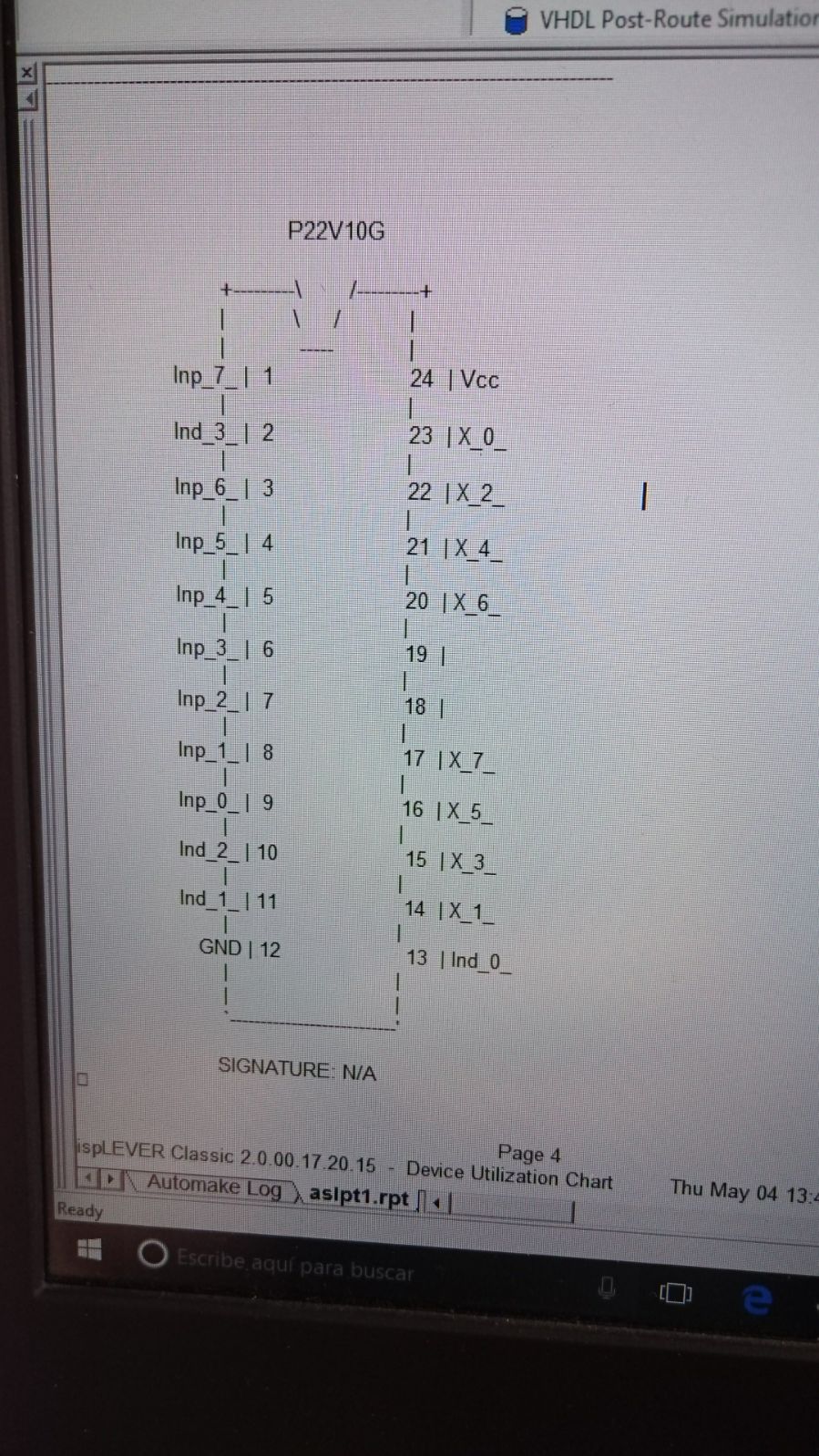
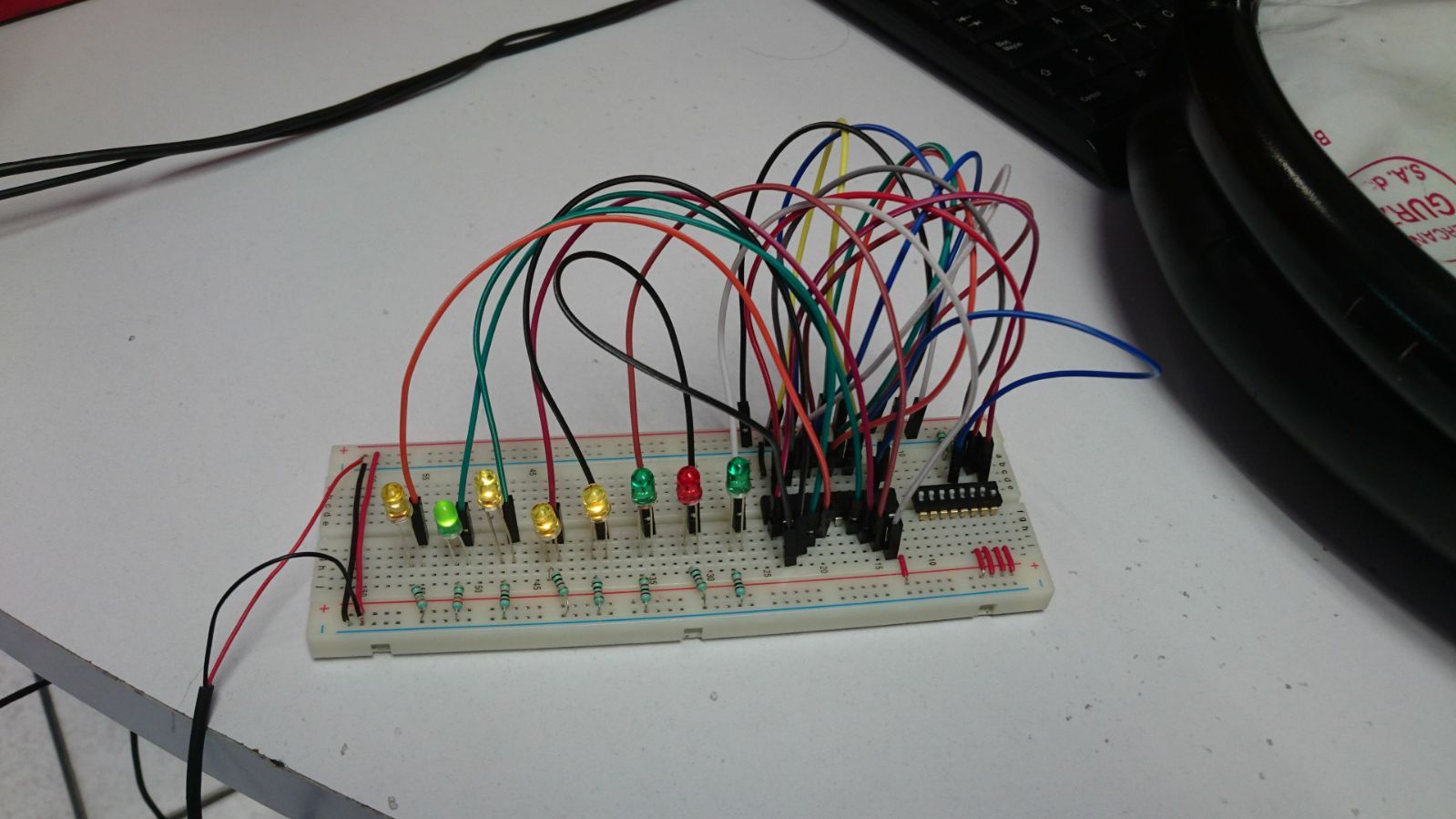


Ilustración 2 Mapa de pines para el ASL, cada “Inp” es el vector de input (entrada) y el “Ind” es el vector (index) que funciona para lograr el ASL.

En la tabla se muestra al circuito funcionando con el ejemplo “00111100” y los corrimientos realizados.

|  |
| --- |
| C:\Users\José Emiliano Pérez\AppData\Local\Microsoft\Windows\INetCache\Content.Word\ASL de 0 con entrada 00111100.jpeg |
| C:\Users\José Emiliano Pérez\AppData\Local\Microsoft\Windows\INetCache\Content.Word\ASL 1 corrimientoi.jpeg |
| C:\Users\José Emiliano Pérez\AppData\Local\Microsoft\Windows\INetCache\Content.Word\ASL 2 Corrimiento.jpeg |
| C:\Users\José Emiliano Pérez\AppData\Local\Microsoft\Windows\INetCache\Content.Word\ASL 3 corrimientos.jpeg |
| C:\Users\José Emiliano Pérez\AppData\Local\Microsoft\Windows\INetCache\Content.Word\ASL 4 corridas.jpeg |
| C:\Users\José Emiliano Pérez\AppData\Local\Microsoft\Windows\INetCache\Content.Word\6 Corridas.jpeg |
| C:\Users\José Emiliano Pérez\AppData\Local\Microsoft\Windows\INetCache\Content.Word\7 o mas corridas.jpeg |

Después de 7 corrimientos el resultado del circuito siempre será 0.

Números primos y números impares: Para los impares hicimos que el circuito comprobara por medio del bit menos significativo si era un ‘1’ o un ‘0’, ya que para todos los impares debe de ser 1.

Para los números primos tuvimos que crear un loop, en el que comprobaba si el número era divisible por otro aparte de 1 y el mismo número, sin embargo, descubrimos que no cabía en una sola GAL así que tuvimos que repartirlo entre dos, de forma que para comprobar si el número era primo o no tuvimos que agregar una compuerta OR en la segunda GAL y programarla para que si alguna de las salidas de las dos GALS era 1 entonces el LED prendería confirmándolo.

|  |  |
| --- | --- |
| C:\Users\José Emiliano Pérez\AppData\Local\Microsoft\Windows\INetCache\Content.Word\Mapa de la primera GAL.JPEG | C:\Users\José Emiliano Pérez\AppData\Local\Microsoft\Windows\INetCache\Content.Word\Mapa de la segunda GAL.JPEG |

En la tabla están los mapas de pines de las dos GALS, los “N22\_3” y “P\_c\_0\_4” son debido a que los resultados ocupan muchas compuertas por lo que crea otra salida innecesaria que no se toma en cuenta, las únicas salidas importantes son “I” en el caso de la primera GAL y “P, I” en el caso de la segunda GAL, además de que la entrada B y C de la segunda GAL son las salidas de la primera y la puerta OR para comprobar el resultado.

En la siguiente tabla se muestran los resultados desde el 0 hasta el 4.

|  |
| --- |
| C:\Users\José Emiliano Pérez\AppData\Local\Microsoft\Windows\INetCache\Content.Word\Primos con 0.jpeg |
| C:\Users\José Emiliano Pérez\AppData\Local\Microsoft\Windows\INetCache\Content.Word\Primos con 1.jpeg |
| C:\Users\José Emiliano Pérez\AppData\Local\Microsoft\Windows\INetCache\Content.Word\Primos con 2.jpeg |
| C:\Users\José Emiliano Pérez\AppData\Local\Microsoft\Windows\INetCache\Content.Word\Primos con 3.jpeg |
| C:\Users\José Emiliano Pérez\AppData\Local\Microsoft\Windows\INetCache\Content.Word\Primos con 4.jpeg |

Como se puede ver, el foco de los números impares (amarillo) sólo prende con el 1 y el 3, el de los primos con el 2 y el 3 pero ninguno prende con el 4 y el 0, de forma que el circuito funciona.

Después probamos con números primos e impares de mayor magnitud tales como el 51 y el 43 y si funcionó.

1. **Observaciones y conclusiones:**

*José Emiliano Pérez Garduño:* Después de haber realizado este proyecto aprendí como se pueden usar los cases de forma eficiente, un loop para poder verificar si un número es primo o no, que si un número en binario tiene como bit menos significativo el 1 es impar, que no debemos usar jumpers para conectar todo el circuito porque algunos pueden venir defectuosos y la capacidad máxima que tiene una GAL22V10

*Alfredo Pérez Quiñonez:* Con el desarrollo del proyecto, conseguimos emplear los conocimientos adquiridos durante el curso sobre VHDL, para poder resolver los dos problemas asignados a nuestro equipo, "Arithmetic Shift Left" y "primos e impares en 8 bits", tuvimos que hacer uso de nuestros conocimientos sobre enunciados secuenciales y concatenación de vectores. El principal obstáculo encontrado al implementar la solución al segundo problema fue la limitante física de la GAL22V10, por lo cual, se utilizó una implementación dividida en dos dispositivos.

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>

<http://www.troubleshooters.com/codecorn/primenumbers/primenumbers.htm>

<http://primes.utm.edu/>

<http://www.mathstools.com/section/main/calculador_factores_primos>

1. **Conclusión de equipo:**

Ambos llegamos a la conclusión de que no siempre podremos programar un circuito para funcionar en una sola GAL y que a veces se requerirán incluso más de 2, por lo que para próximos proyectos lo tomaremos en cuenta.