

TE2002B : Actividad 1.1

Equipo 3:

David Israel Vázquez Leal	A00833121
José Emiliano Flores Martínez	A00833160
Juan Carlos Hernández Ríos	A01740821

Resumen de la plática con el Socio Formador Intel:

Primero, se habló de manera general sobre la historia de la compañía Intel. Fue fundada por Robert Noyce, co-inventor del circuito integrado, Andrew Grove y Gordon Moore, creador de la famosa Ley de Moore. Sin la anterior ley, la humanidad no habría empujado con tanto fervor la innovación en Hardware y Software, específicamente la innovación tecnológica de empujarse los transistores usados en circuitos integrados. Posteriormente, se habló sobre el futuro del mundo y su crecimiento en cuatro áreas tecnológicas importantes: la computación presente en todos lados, la conectividad generalizada, el uso del edge (sensores) en conjunto a la interactividad entre la nube y el edge, y el uso de la inteligencia artificial. Después, se habló sobre el ecosistema de High Tech presente en Guadalajara, siendo Guadalajara Intel una parte importante de ello que permite la innovación y el desarrollo de la tecnología en México. Luego, se habló sobre Intel Guadalajara y el desarrollo de los circuitos integrados que se crean ahí: desarrollo de Silicio, Desarrollo de paquete Si, Diseño de la tarjeta, Desarrollo de Plataforma, Integración y Prueba de Sistema, Integración de Rack y Soporte al cliente; también, se habló sobre los diversos ámbitos en los que se trabajan en sus laboratorios: desarrollo de cómputo en la nube, inteligencia artificial, internet de las cosas, cómputo de alto desempeño, procesamiento cuántico, gráficos, entre otros avances importantes de la industria computacional. Posteriormente, se habló sobre el apoyo de Intel a México, desarrollando múltiples iniciativas con universidades, creando varios laboratorios de investigación en múltiples estados de la república: Monterrey, Baja California, Sonora, Yucatán, entre otros, así como realizando programas como Technovation o Sisters in STEM que impulsen a jóvenes mexicanos. Luego, se habló sobre la cultura laboral en Intel, pidiendo a sus trabajadores que entreguen a tiempo y forma el trabajo que se comprometen a realizar, tener métodos de resolución de problemas, estructurar el trabajo en equipo, saber cuándo pedir ayuda, continuo desarrollo personal, asumir actitudes de liderazgo, entre otros aspectos importantes. Por último, se incitó a los jóvenes a realizar su mejor esfuerzo en la realización del proyecto del bloque, dando libertad absoluta sobre la forma en la que se puede realizar.

Desarrollo de los PLDs

Las PROMs

La primera iteración formal de los PLDs se dio con el uso de la Programmable Read-Only Memory (PROM). La PROM fue creada por Wen Tsing Chow, empleado de American Bosch Arma Corporation, en la segunda mitad de los años 50, para miniaturizar el sistema de guía de misiles ICBM (Air Force Space Command, s.f.). Estos dispositivos permitían obtener salidas deseadas con base a la manipulación física de las entradas, de forma que se utilizaba este método para producir electrónicos de propósito específico. Sin embargo, debido a que las PROM tenían todas sus salidas habilitadas, estaban limitadas en términos

de consumo de poder y hardware, por lo que se buscaron otras formas de programar PLDs (Maxfield, 2011).

PLAs

Las Programmable Logic Arrays (PLAs) fueron diseñadas en 1970 por Texas Instruments (Texas Instruments, 1970). Con el fin de sobrepasar las limitaciones de las PROMs, las PLAs aparecieron para evitar tener que tomar en consideración las salidas y entradas que se iban a procesar mediante el “quemado” de sus conexiones internas no deseadas usando fusibles. Así mismo, debido a su disposición de compuertas AND para tomar las combinaciones de entradas deseadas, así como compuertas OR que condicionan las salidas con circuitos complementarios, se pueden resolver circuitos combinacionales complejos y ahorrar piezas de hardware desperdiciadas como en el caso de las PROMs (Gautam, 2018)

PALs

Las Programmable Array Logics (PALs) consisten en una serie de compuertas AND programables con compuertas OR fijas que se ajustan al propósito del circuito combinacional. A través de las compuertas AND pasa un bus con distintas combinaciones de las entradas con sus respectivos complementos, que se seleccionan para formar las expresiones que se combinarán en compuertas OR y de esa forma generar minitérminos (Gautam, 2018). Fueron desarrolladas por John Birkner y H.T. Chua of Monolithic Memories en 1978 con el fin de proveer de un circuito integrado que tuviera la capacidad de ser programado “eléctricamente” y de forma eficiente (Computer History, s.f) .

GALs

Inventada por Lattice Semiconductor en 1985, las Generic Array Logic (GAL) fueron creadas con el propósito de ser múltiples PALs que tenían la facultad de ser reprogramadas eléctricamente (Maxfield, 2011). Esto ayuda a que sean usadas comúnmente para prototipos, debido a que sus arquitecturas flexibles permiten la implementación de diseños que serían difíciles o imposibles de implementar en dispositivos PAL normales. Sus macroceldas pueden ser programadas en tres modos diferentes: modo registro, en el que los outputs de la GAL pueden ser registrados o combinacionales; modo combinacional, en el que se trata de emular 16L8 PALs; y modo simple, emulando dispositivos simples de PAL en el que se usan 16 inputs para 2 outputs (Yang, 2023).

CPLD

Los CPLD (Complex Programmable Logic Devices) esencialmente son un conjunto de PALs programables para procesar múltiples entradas lógicas y definir sus salidas. Cuentan con una memoria EEPROM, lo que le permite almacenar y ejecutar los programas que se les haya instalado incluso después de un reboot (Digikey, s.f). Así mismo, a comparación de los FPGAs, no tienen tanto poder de procesamiento, por lo que se usan para aplicaciones menos específicas. Aún así, la razón principal por la que los CPLDs siguen en uso es porque a comparación de los FPGAs, su estructura más simple les permite abaratar costos de forma significativa y ocupar menos espacio, y así mismo, los CPLDs no están diseñados para operar a la misma frecuencia que los FPGAs, si no a menores, por lo que su uso principal es para lógica combinacional que no requiere ser secuencial (Roca, 2023).

FPGA

Los FPGAs consisten en un conjunto enorme de bloques lógicos programables que, a diferencia de los ASIC (Application Specific Integrated Circuits) se pueden reprogramar enteramente tras su manufactura. El predecesor directo del FPGA fue creado inicialmente en 1984 por la compañía Altera a través del circuito EP300 que contaba con un sistema EPROM para ser reprogramado. Después, en 1987, Steve Casselman planteó la creación de un sistema con 600,000 compuertas reprogramables gracias al financiamiento del Naval Surface Warfare Center, y con ello nació el FPGA (Hardware Bee, s.f).

1) ¿Cuál es la diferencia entre un PAL, PLA y un GAL?

El PAL consiste en un circuito integrado que cuenta con múltiples compuertas AND de entrada programables y salidas OR fijas cuya salida se puede condicionar para cumplir el propósito del circuito. Por otro lado, los PLAs comparten un diseño parecido a los PALs, con la diferencia de que tanto sus entradas AND como sus salidas OR son programables (GeeksforGeeks, s.f). Los beneficios de usar PALs que PLAs son: muy eficientes y seguras, costo de producción menor, requiere poca potencia y son fáciles de usar (Gautam, 2018).

Por otro lado, las GAL son circuitos integrados que se componen de múltiples PAL, y tienen la particularidad de que se pueden reprogramar eléctricamente a través de HDLs como VHDL o Verilog, lo que permite que sean usados para prototipos (Circuit Reset, s.f).

2) ¿Cómo se implementa una función booleana en un PAL (o PLA) y cómo se implementa en un FPGA?

En un PAL o PLA lo que se hace es una matriz con las combinaciones de las entradas que se busquen, de forma que se tiene N entradas con sus respectivos complementos, conectados a través de un bus directamente a una serie de compuertas AND. Las expresiones obtenidas en estas compuertas permiten expresar minterminos. Por otro lado, una vez se obtienen todos los minterminos, las expresiones toman una estructura similar a las entradas del circuito integrado y se hace una nueva matriz con estos, pero para entrar a una serie de puertas OR. Como resultado, se obtienen maxiterminos con los minterminos, por lo que se pueden expresar las funciones booleanas que sean necesarias siempre y cuando existan suficientes salidas (Gautam, 2018).

Por otro lado, para implementar una función en un FPGA es necesario usar un HDL (High Description Language) que modifica el comportamiento de los bloques programables internos del circuito. Básicamente se cambian las entradas y salidas del circuito electrónicamente con ayuda de un software, de forma que se puede reprogramar cuantas veces sea necesario (Xilinx, s.f).

Link de Genially:

<https://view.genial.ly/64050c2cdf7916001ab3ee3b/interactive-content-11>

Referencias (poner en APA después):

Air Force Space Command. (s.f.). *Mr. Wen Tsing Chow*. afspc.af.mil. <https://www.afspc.af.mil/Portals/3/documents/Pioneers/AFD-100405-081.pdf>

Texas Instruments. (1970). *MOS/LSI from Texas Instruments*. Texas Instruments Incorporated. http://www.bitsavers.org/components/ti/_dataBooks/1970_MOS_LSI_from_Texas_Instruments.pdf

AMD Xilinx. (s. f.). *Programming an FPGA: An Introduction to How It Works*. Xilinx. Recuperado 5 de marzo de 2023, de <https://www.xilinx.com/products/silicon-devices/resources/programming-an-fpga-an-introduction-to-how-it-works.html>

Circuit Reset. (s. f.). *What is Generic Array Logic (GAL)?* Circuit Rest. Recuperado 5 de marzo de 2023, de <https://circuitreset.com/what-is-a-gal-device/>

Computer History. (s. f.). *1978: PAL USER-PROGRAMMABLE LOGIC DEVICES INTRODUCED*. Computer History. Recuperado 5 de marzo de 2023, de <https://www.computerhistory.org/siliconengine/pal-user-programmable-logic-devices-introduced/>

Digikey. (s. f.). *CPLD (dispositivos lógicos programables complejos)*. Recuperado 5 de marzo de 2023, de <https://www.digikey.com.mx/es/products/filter/integradas/cpld-dispositivos-l%C3%B3gicos-programables-complejos/695>

History and Milestones of the FPGA. (s. f.). *HardwareBee*. Recuperado 5 de marzo de 2023, de <https://hardwarebee.com/history-milestones-fpga/#:~:text=In%20the%20mid%2080's%20Altera,blocks%20with%20connections%20between%20them.>

Gautam. (2018). *Programming Array Logic*. geeksforgeeks. Recuperado 5 de marzo de 2023, de <https://www.geeksforgeeks.org/programming-array-logic/?ref=rp>

Gautam. (2018). *Programmable Logic Array*. Recuperado 5 de marzo de 2023, de <https://www.geeksforgeeks.org/programmable-logic-array/?ref=rp>

Gautam. (2018). *Difference between Programmable Logic Array and Programming Array Logic*. GeeksforGeeks. Recuperado 5 de marzo de 2023, de <https://www.geeksforgeeks.org/difference-between-programmable-logic-array-and-programming-array-logic/>

Maxfield, C. (2011, 20 septiembre). *Who made the first PLD? EE Times*. Recuperado 5 de marzo de 2023, de <https://www.eetimes.com/who-made-the-first-pld/>

MOS/LSI from Texas Instruments. (1969, octubre). *Bitsavers*. Recuperado 5 de marzo de 2023, de http://www.bitsavers.org/components/ti/_dataBooks/1970_MOS_LSI_from_Texas_Instruments.pdf

Roca, J. (2023, 13 febrero). *Así son los CPLD, la alternativa de menor coste a los FPGA*. HardZone. Recuperado 5 de marzo de 2023, de <https://hardzone.es/reportajes/que-es/cpld/>

Yang, C. (2023). A Tour of PLDs: Programmable Logic Device (PLD) Handout.