



**BENEMÉRITA UNIVERSIDAD AUTÓNOMA DE PUEBLA**

FACULTAD DE CIENCIAS DE LA ELECTRÓNICA  
MAESTRÍA EN CIENCIAS DE LA ELECTRÓNICA  
OPCIÓN EN AUTOMATIZACIÓN

“ \* ”

# **T E S I S**

Presentada para obtener el título de:  
**Maestro en Ciencias de la Electrónica, Opción en Automatización**

Presenta:

**Lic. Tomás Miguel Vargas García\*\***

CVU:772553

Directores:

Dr. Sergio Vergara Limón (FCE-BUAP) CVU:22133

Dr. Fernando Reyes Cortes (FCE-BUAP) CVU:10830

**Puebla, México**

**Octubre 2018**

\* TRABAJO FINANCIADO POR EL PROYECTO VIEP-BUAP

\*\*BECARIO CONACYT

## **AGRADECIMIENTOS**

A Dios por darme la oportunidad de cursar la maestría y disfrutar de la compañía de multitud de personas, las cuales han sido parte importante en mi desarrollo personal y profesional.

Agradezco a la Facultad de Ciencias de la Electrónica de la Benemérita Universidad Autónoma de Puebla por brindarme la formación profesional con la cual cuento.

Agradezco a mis padres por brindarme su apoyo a lo largo de mi vida, por ayudarme a superar obstáculos y enfrentar nuevos retos.

Agradezco a mi esposa y a mi hijo por apoyarme en las decisiones de mi vida, antes y durante la maestría

A todo el grupo de personas que conforman la Maestría de Ciencias de la Electrónica, Opción Automatización por brindarme la oportunidad de continuar mi formación profesional en sus instalaciones.

A mis asesores de tesis la Dr. Sergio Vergara Limon, Dr. Fernando Reyes Cortes, por el apoyo tanto de conocimientos, moral y asesoría para realizar este trabajo, en especial al Dr. Sergio Vergara Limon por brindarme los conocimientos y la paciencia necesaria para desarrollar esta tesis. A la doctora Ma. Aurora Vargas Treviño por su paciencia, asesorías y apoyo moral en mi desarrollo en la maestría.

A mis sinodales, ya que con sus comentarios preguntas y correcciones contribuyeron de manera significativa en el desarrollo y enriquecimiento de este trabajo.

A CONACyT por el apoyo económico brindado para financiar los gastos personales para terminar la maestría.

A mis compañeros y amigos de la Maestría de Ciencias de la Electrónica, Opción Automatización por la ayuda que brindaron en algún momento, Martín, Dany, Erick Javier, Oscar Isaid, Nelson, Concepción, Luis, Erik, Alejandro, Carlos, Nahum.

A los compañeros de la Generación 2015: Geovanni, Francisco, Gabino, Moya, Rigoberto, Daniel, Hugo y Hermes; por su apoyo y asesoramiento en mi desarrollo dentro de la maestría.

A la Benemérita Universidad Autónoma de Puebla por facilitar sus instalaciones y materiales de estudio.

Al Consejo de Ciencia y Tecnología del Estado de Puebla, por su apoyo con el programa “beca tesis” del cual resulte beneficiado.

## RESUMEN

En este trabajo se realiza el diseño e implementación de un generador de patrones digitales programable a 50 MHz, las características que definen al dispositivo son: paso máximo de reproducción 20ns, 24 bits de salida divididos en 2 canales de 12 bits cada uno, profundidad de memoria de 4 KBytes y un rango de amplitud por salida de  $\pm 5V$ , cuenta con patrones digitales preestablecidos, además de la capacidad de capturar y guardar nuevos patrones digitales editados por el usuario, la comunicación entre la PC y el dispositivo es inalámbrica gracias al uso del módulo WiFi. El diseño del firmware del generador de patrones digitales es realizado en AHDL, la programación de una interfaz de usuario intuitiva se realiza en LabView el cual es un software de instrumentación virtual. La interfaz de usuario permite la asignación de salidas, niveles de voltaje digital; según la tecnología a usar; el paso de reproducción y la captura del patrón digital a reproducir así como la previsualización del patrón digital capturado. Para demostrar la funcionalidad del dispositivo se presentan las capturas realizadas en el analizador lógico de la marca Aligent, a de más de las gráficas del osciloscopio digital de la marca Tektronics para las mediciones de los niveles de voltaje editados por el usuario o seleccionados según la tecnología digital a emular, además de demostrar la frecuencia máxima de reproducción del dispositivo. Finalmente se presenta dos ejemplos de aplicación del generador de patrones digitales emulando el protocolo SPI e i2c.

## **DEDICATORIA**

A mis padres porque en ellos encontré la inspiración de seguir adelante y superarme. A mi esposa e hijo por ser la fuerza de mi vida.

# INDICE

AGRADECIMIENTOS.....	i
RESUMEN .....	ii
DEDICATORIA .....	iii
Introducción .....	1
CAPITULO 1: Descripción del sistema.....	7
1.1 Diagrama general del sistema .....	8
1.2 Tipos de tecnologías digitales .....	12
1.3 Aplicaciones del generador de patrones digitales .....	13
1.4 Conclusiones.....	13
CAPITULO 2: Hardware del generador de patrones digitales.....	14
2.1 Descripción general del hardware .....	16
2.2 Tarjeta de comunicación WiFi módulo RN-XV WiFly .....	16
2.3 Tarjeta embebida DE0-NANO .....	17
2.4 Etapa de acondicionamiento de las salidas digitales .....	19
2.4.1 DAC THS5651A .....	21
2.4.2 Amplificador Operacional.....	25
2.4.3 Filtrado. ....	26
2.5 Diseño del circuito impreso PCB .....	27
2.5.1 Tipos de ruido.....	29
2.6 Descripción del hardware .....	30
2.7 Conclusiones .....	31
CAPITULO: 3 Firmware .....	32
3.1 Multiplicador de frecuencia PLL.....	34
3.2 De serial a paralelo bloque UART_RX2.....	35
3.4 Identificación de comandos .....	37
3.5 Cambio de paso.....	41
3.6 Administrador de memoria.....	45
3.7 Memoria FIFO.....	47

3.8 Firmware para la configuración del voltaje de salida según la tecnología a utilizar.....	49
3.8.1 Dacbuffer Reference .....	50
3.8.2 Bus patrón .....	52
3.8.2 Dac1datos---dac24datos. ....	53
3.9 Conclusiones.....	54
<b>CAPITULO 4: SOFTWARE DE LA INTERFAZ DE USUARIO.....</b>	<b>55</b>
4.1 Asignación de salidas.....	56
4.2 Voltaje de los niveles lógicos de las salidas.....	61
4.2.1 Margen de ruido.....	62
4.2.2 Compatibilidad con las tecnologías digitales .....	63
4.2.1 Por canal.....	66
4.2.1 Individual .....	68
4.2.3 Editar nivel.....	70
4.3 Captura de datos .....	72
4.4 Patrones digitales preestablecidos. ....	76
4.4.1 Alternancia de unos.....	76
4.4.2 Recorriendo el uno de derecha a izquierda. ....	77
4.4.3 Todos uno, todos cero.....	77
4.4.4 Recorriendo el uno de izquierda a derecha .....	78
4.5 Single y cíclico, pasó de tiempo,.....	78
4.6 Gráfica y envío de datos .....	80
4.7 Conclusiones.....	85
<b>CAPITULO 5 Resultados experimentales .....</b>	<b>86</b>
5.1 Resultados visualizados en el analizador lógico Agilent .....	86
5.1.1 Patrón digital AAA555 .....	86
5.1.1 Patrón digital recorriendo 1 Derecha a Izquierda .....	87
5.1.2 Patrón digital recorriendo 1 Izquierda a Derecha .....	89
5.1.2 Patrón digital todo uno, todo cero.....	90
5.1.4 Gráficas a diferentes pasos de tiempo.....	91
5.2 Resultados de la tarjeta acopladora de voltaje, visualizados en el osciloscopio .....	96

5.2.1 Gráficas de la salida para acoplamiento con la tecnología ECL. ....	96
5.2.2 Gráficas de la salida para acoplamiento con la tecnología PECL .....	98
5.2.3 Gráficas de la salida para acoplamiento con la tecnología CMOS .....	99
5.2.4 Gráficas de la salida para acoplamiento con la tecnología TTL.....	101
5.3 Aplicaciones del generador de patrones digitales. ....	102
5.3.1 Protocolo SPI .....	102
5.3.2 Protocolo I2C .....	104
5.4 Conclusiones.....	107
Capítulo 6 Conclusiones Generales .....	108
Apéndice A Participación en congreso .....	110
Apéndice B Artículo .....	111
Apéndice C Manual de usuario .....	117
DESCRIPCIÓN DEL SOFTWARE.....	118
SALIDAS .....	118
VOLTAJE.....	118
DATOS.....	120
GRAFICA .....	121
DESCRIPCIÓN DEL HARDWARE.....	122
INSTRUCCIONES DE USO .....	123
Referencias.....	124

## INDICE DE FIGURAS

Figura 1: Diagrama a bloques de la conexión del generador de patrones con el dispositivo bajo prueba. ....	2
Figura 2: Generadores de patrones digitales en el mercado. ....	2
Figura 1.1: Diagrama general del generador de patrones digitales. ....	7
Figura 1.2: Módulo PC y módulo de la tarjeta wifly. ....	8
Figura 1.3: Señal de reloj y módulo multiplicador de frecuencia. ....	9
Figura 1.4: Módulo convertidor serie-paralelo. ....	9
Figura 1.5: Módulo codificador de comandos y selección de datos. ....	10
Figura 1.6: Módulo administrador de la memoria FIFO. ....	10
Figura 1.7: Módulo divisor de frecuencia. ....	11
Figura 1.8: Memoria FIFO de 8KBytes. ....	11
Figura 1.9: Módulo para el acondicionamiento del nivel de voltaje. ....	11
Figura 1.10: Gráficas de las diferentes tecnologías digitales. ....	12
Figura 2.1: Diagrama electrónico del generador de patrones digitales. ....	15
Figura 2.2: Diagrama general del hardware. ....	16
Figura 2.3: Módulo WiFly RN-171-XV. ....	16
Figura 2.4: Pines de comunicación y características del módulo WiFi RN-XV. ....	17
Figura 2.5: Diagrama de bloques de la tarjeta DE0-NANO. ....	17
Figura 2.6: Pines de la tarjeta DE0-NANO. ....	18
Figura 2.7: Conexiones de la tarjeta DE0-NANO. ....	19
Figura 2.8: Diagrama de acoplamiento para diferentes tecnologías digitales. ....	20
Figura 2.9: Descripción de pines del DAC THS5651A. ....	21
Figura 2.10: Diagrama de tiempos del DAC THS5651A. ....	22
Figura 2.11: Conexión entre el FPGA y el DAC THS5651A. ....	23
Figura 2.12: Caracterización del DAC THS5651A. ....	24
Figura 2.13: Configuración de los pines del OPA659. ....	25
Figura 2.14: Filtro pasa bajas R-C. ....	27
Figura 2.15: Diagrama esquemático del circuito de acoplamiento de señal. ....	28
Figura 2.16: Filtro EMI. ....	29
Figura 2.17: Conexiones del Hardware. ....	30
Figura 2.18: Descripción de las conexiones de la tarjeta acopladora de voltaje. ....	31
Figura 3.1: Descripción breve del funcionamiento del firmware del generador de patrones digitales. ....	32
Figura 3.2: Diagrama de firmware del generador de patrones digitales. ....	33
Figura 3.3: Multiplicador de Frecuencia. ....	34
Figura 3.4: Señales de reloj a la entrada y salida del módulo PLL. ....	34
Figura 3.5: Protocolo de comunicación UART-232. ....	35

Figura 3.6: Tiempo de muestreo del dato recibido por WI-FI.....	35
Figura 3.7: Bloque UART_RX2. ....	36
Figura 3.8: Diagrama de estados del bloque UART_RX2.....	36
Figura 3.9: Ejemplo de los valores convertidos de serie a paralelo por el módulo UART_RX2 en su salida dataOutA.....	37
Figura 3.10: Módulo donde se unen los 5 bytes. ....	37
Figura 3.11: Diagrama a bloques del módulo identificación de comandos. ....	38
Figura 3.12: Diagrama de estados del módulo word32_28_1.....	38
Figura 3.13: Ejemplo de la identificación de comandos. ....	39
Figura 3.14: Diagrama de flujo del estado E0. ....	40
Figura 3.15: Diagrama de flujo del estado E1. ....	41
Figura 3.16: Comando 245: configuración del reloj de la memoria FIFO.....	42
Figura 3.17: Módulo divisor de frecuencia.....	42
Figura 3.18: Comando para cambiar el paso de reproducción de la memoria FIFO.....	43
Figura 3.19: Diagrama de máquina de estados CLKPASOSFIFO. ....	43
Figura 3.20: Salida del bloque CLOCKPASOSFIFO, sin configurar el divisor de frecuencia 100MHz. ....	44
Figura 3.21: Salida del bloque CLOCKPASOSFIFO configurado a una frecuencia de 50 MHz. ....	44
Figura 3.22: Módulo clkfifowifi2. ....	45
Figura 3.23: Secuencia de comandos para la escritura y lectura de la memoria FIFO. ....	45
Figura 3.24: Máquina de estado del bloque clkfifowifi2.....	46
Figura 3.25: Ejemplos de datos a guardar junto con la condición "difdata".....	47
Figura 3.26: Módulo de la memoria FIFO.....	47
Figura 3.27: En FIFO, los valores de los datos se leen de la memoria (b) en el mismo orden en el que se escribieron en la memoria(a). ....	48
Figura 3.28: Se muestra la habilitación de la escritura de la memoria y el pulso de reloj de la memoria FIFO para guardar el dato presente en la entrada de la memoria FIFO.....	48
Figura 3.29: Lectura y escritura de la memoria FIFO para mantener los datos en la memoria.....	49
Figura 3.30: Diagrama a bloques del firmware del acoplamiento de voltaje. ....	50
Figura 3.31: Bloque para recibir la palabra de 10 bits donde viene la dirección, el nivel alto y bajo de la salida asignada.....	51
Figura 3.32: Diagrama a bloques del módulo Dacbuffer Reference. ....	51
Figura 3.33: Diagrama de máquina de estados del módulo Dacbuffer_Reference.....	52
Figura 3.34: Módulo P1atterndata.....	52
Figura 3.35: Módulo DacXdatos. ....	53
Figura 3.36: Diagrama de máquina de estados del módulo DacXdatos. ....	53
Figura 3.37: Gráfica de las salidas del módulo DacXdatos. ....	54
Figura 4.1: Descripción general del funcionamiento del software. ....	55
Figura 4.2: Gráfica de un patrón digital para configurar un DAC por $I^2 C$ . ....	56
Figura 4.3: Secuencia de la asignación de salidas. ....	56

Figura 4.4: Descripción de la pestaña de asignación de salidas.....	57
Figura 4.5: Proceso de identificación de error en la asignación de salidas, del canal 1. ....	58
Figura 4.6: Descripción de los colores del indicador de error.....	59
Figura 4.7: Diagrama de flujo de los datos a enviar y graficar según la habilitación de las salidas..	60
Figura 4.8: Ejemplo de error en la asignación de salidas. ....	60
Figura 4.9: Niveles lógicos de entrada-salida y márgenes de ruido para puertas TTL estándar.....	61
Figura 4.10: Niveles de voltaje de las salidas del generador de patrones digitales.....	63
Figura 4.11: Estructura del comando 230. ....	64
Figura 4.12: Opciones de configuración de los niveles de voltaje de las salidas. ....	64
Figura 4.13: Diagrama de flujo de la habilitación de la tarjeta de acoplamiento de voltaje y configuración.....	65
Figura 4.14: Descripción de la pestaña por canal de la configuración de voltaje. ....	66
Figura 4.15: Diagrama de flujo de la selección de niveles de voltaje por canal 1 para el canal 2 es el mismo procedimiento. ....	67
Figura 4.16: Descripción de la pestaña "individual" de la configuración de voltaje. ....	68
Figura 4.17: Diagrama de flujo de la configuración de salidas de manera individual.....	69
Figura 4.18: Pestaña de la opción "Editar Nivel".....	70
Figura 4.19: Diagrama de flujo que describe el funcionamiento de la pestaña "Editar nivel". ....	71
Figura 4.20: Pestaña "Datos" de la interfaz gráfica. ....	72
Figura 4.21: Acciones a realizar por los botones: limpiar datos, leer, guardar patrón.....	73
Figura 4.22: Diagrama a bloques para guardar un nuevo patrón digital. ....	73
Figura 4.23: Diagrama a bloques para leer un patrón digital previamente guardado.....	74
Figura 4.24: Descripción de la pestaña datos. ....	74
Figura 4.25: Diagrama de flujo de la gráfica que representa el patrón digital capturado.....	75
Figura 4.26: Patrón digital capturado en los dos arreglos. ....	76
Figura 4.27: Gráfica del patrón digital AAA555.....	76
Figura 4.28: Patrón digital y gráfica para recorrer el uno de derecha a izquierda. ....	77
Figura 4.29: Patrón digital y gráfica donde todas las salidas alternan en uno y cero.....	77
Figura 4.30: Patrón digital y gráfica donde el uno se va recorriendo desde el bit 11 hacia el bit 0. ....	78
Figura 4.31: Botones single, cíclico y caja de paso de tiempo. ....	78
Figura 4.32: Función de los botones cíclico y single.....	79
Figura 4.33: Gráfica del patrón digital todos uno, todos cero con un paso de tiempo de 4. ....	79
Figura 4.34: Diagrama de flujo del envío del paso de cada renglón del patrón digital. ....	80
Figura 4.35: Secuencia de comandos enviados al FPGA desde el software del generador de patrones digitales.....	80
Figura 4.36: Secuencia del envío de datos al firmware del generador de patrones digitales. ....	81
Figura 4.37: Tercera Pantalla de la interfaz gráfica sin patrón digital.....	81
Figura 4.38: Gráfica de un patrón digital de 5 líneas de salida. ....	82
Figura 4.39: Secuencia de la creación de la gráfica y envío de los datos por WiFi. ....	83
Figura 4.40: Explicación de cómo se grafican los datos del patrón digital. ....	83

Figura 4.41: Diagrama de flujo para la lectura del arreglo del patrón digital y gráfica del patrón digital en Labview.....	84
Figura 5.1: a) Patrón digital AAA555, b) Gráfica del patrón en el software del generador de patrones digitales.....	86
Figura 5.2: Gráfica del patrón digital AAA55 en analizador lógico Agilent. ....	87
Figura 5.3: a) Datos, b) gráfica del patrón digital recorriendo uno D-I.....	88
Figura 5.4: Gráfica del patrón digital recorrer uno D-I en analizador lógico Agilent. ....	88
Figura 5.5: (a) Datos, (b) gráfica del patrón digital recorriendo 1 I-D.....	89
Figura 5.6: Gráfica del patrón digital recorrer uno D-I visualizado en el analizador lógico de Agilent. ....	90
Figura 5.7: (a) Datos, (b) gráfica del patrón digital todos uno, todos cero.....	90
Figura 5.8: Gráfica del patrón digital todos uno, todos cero, capturado en el analizador lógico de la marca Aligent. ....	91
Figura 5.9: Configuración del tipo de reproducción y paso del patrón digital.....	92
Figura 5.10: Gráfica del patrón todo uno, todo cero con un paso de 100ns. ....	92
Figura 5.11: Configuración del paso a 500ns. ....	92
Figura 5.12: Patrón digital todo uno, todo cero con un paso de 500ns.....	93
Figura 5.13: Paso de tiempo configurado a 1us.....	93
Figura 5.14: Patrón digital con un paso de 1us.....	94
Figura 5.15: Paso de tiempo configurado a 5us.....	94
Figura 5.16: Patrón digital con un paso de 5us.....	94
Figura 5.17: Configuración del paso de tiempo a 1ms.....	95
Figura 5.18: Patrón digital con un paso de 1ms.....	95
Figura 5.19: Configuración del paso de tiempo a 2ms.....	95
Figura 5.20: Patrón digital con un paso de 2ms.....	95
Figura 5.21: Gráfica de la tecnología ECL. -0.8: 1 Lógico, -1.8: 0 Lógico. a) 50 MHz, b) 25 MHz, c) 10MHz. ....	96
Figura 5.22: Circuito ECL básico. ....	97
Figura 5.23: Gráfica de la señal PECL a) 50MHz, b) 25MHz, c)5 MHz. ....	98
Figura 5.24: Gráfica de la señal LVPECL a) 50MHz, b) 25MHz,c) 5 MHz. ....	99
Figura 5.25: Gráfica de una señal CMOS a) 50 MHz, b) 25MHz, c) 5MHz.....	100
Figura 5.26: Gráfica de una señal TTL a) 50MHz, b) 25MHz, c) 10MHz. ....	101
Figura 5.27: Descripción de los pines del DAC MCP4921.....	102
Figura 5.28: Diagrama de tiempos del DAC MCP4921.....	102
Figura 5.29: patrón digital para configurar el DAC MCP4921.....	103
Figura 5.30: Protocolo de comunicación SPI.....	104
Figura 5.31: Patrón digital visualizado en el analizador lógico de Quartus II. ....	104
Figura 5.32: Visualización del protocolo SPI en el analizador lógico de Agilent. ....	104
Figura 5.33: Protocolo de comunicación I2C.....	105

Figura 5.34: Definición de las condiciones de inicio y final del protocolo i2c.....	105
Figura 5.35: Diagrama de tiempos del protocolo I2C. ....	105
Figura 5.36: Protocolo i2c a la salida del generador de patrones digitales. ....	106
Figura 5.37: Visualización del protocolo SPI en el analizador lógico de Agilent. ....	106
Figura 5.38: Periodo de la señal de reloj del protocolo i2c, frecuencia de 66.6666MHz. ....	107
Figura 5.39: Periodo de la señal de reloj del protocolo i2c, frecuencia de 100kHz. ....	107
Manual figura 1.1 Pestañas del software del generador de patrones digitales. ....	117
Manual figura 1.2: Pestaña SALIDAS. ....	118
Manual figura 1.3: Pestaña VOLTAJE- POR CANAL. ....	119
Manual figura 1.4: Pestaña VOLTAJE-INDIVIDUAL - EDITAR NIVEL.....	119
Manual figura 1.5: Descripción de la pestaña DATOS.....	120
Manual figura 1.6: Botones de configuración del paso y tipo de reproducción. ....	121
Manual figura 1.7: Pestaña de Grafica.....	121
Manual figura 1.8: Conexiones del Hardware.....	122
Manual figura 1.9: Descripción de las conexiones de la tarjeta acopladora de voltaje.....	122

## INDICE DE TABLAS

Tabla 1: Tabla comparativa de los generadores de patrones digitales.....	4
Tabla 1.1: Tipos de señales digitales. ....	12
Tabla 2.1: Modos de la interfaz de entrada. ....	21
Tabla 2.2: Descripción de los tiempos del diagrama de tiempo. ....	22
Tabla 2.3: Números en decimal a convertir por el DAC THS5651A.....	24
Tabla 2.4: Ganancias propuestas por el fabricante respecto al ancho de banda. ....	25
Tabla 3.1: Tabla de comandos.....	39
Tabla 4.1: Niveles de voltaje de las salidas del generador de patrones digitales. ....	63

## Introducción

La automatización es un área estratégica en el desarrollo de un país ya que permite que la producción de bienes se realice en menor tiempo y en mayor cantidad respecto a procesos manuales. Además, la precisión de las maquinas muchas veces llega a ser mejor que lo que se pudiera lograr manualmente, lo anterior elimina las fallas humanas, incrementa la calidad y la velocidad de producción, esto reduce los costos y mejorar la competitividad de las empresas respecto a sus pares en otros países.

En el desarrollo de un producto o tecnología es necesario llevar acabo investigación en el campo donde se quiere llevar a cabo el desarrollo del producto o tecnología, con la finalidad de obtener un producto, proceso o tecnología competente con la de otros países.

La aplicación de conocimientos y desarrollo tecnológico en el diseño de instrumentos para emulación de sistemas, a través de la naturaleza multidisciplinaria de la automatización, permite una relación directa y aplicaciones en el área del desarrollo tecnológico de un país. Asimismo, dota al especialista de una herramienta que le facilita aplicar sus conocimientos en bien de la sociedad. Los instrumentos de un laboratorio de robótica o electrónica digital, cuando se encuentra automatizados con la tecnología y elementos teóricos, permiten al especialista ampliar sus conocimientos, obtener mejores diagnósticos, técnicas de diseño, óptimas y eficientes para el diseño de circuitos integrados, protocolos de comunicación, diagnostico de fallas, lo que produce como consecuencia un rápido proceso de diseño e innovación, logrando los resultados esperados en menor tiempo y con mejor desempeño, de esta forma las expectativas de desarrollo e innovación tecnológica de un país son mejoradas.

Un generador de patrones digitales es exactamente lo que su nombre implica: un generador de patrones digitales usado como un estimulador para realizar pruebas digitales [10]. El generador de patrones digitales puede proporcionar un patrón digital repetible, conocido o con errores; esta característica es una de las virtudes más sobresalientes del generador de patrones digitales; ya que a menudo es imposible crear un error predecible exactamente en el momento y en el lugar donde se necesita, utilizando solo el circuito en sí. La respuesta del dispositivo bajo prueba (DUT) revela su capacidad de manejar los datos que quedan fuera del funcionamiento normal del dispositivo.

Los generadores de patrones, también conocidos como generadores de datos o generadores de sincronización de datos, típicamente proporcionan 8, 16 o incluso más flujos de pulsos digitales sincronizados como señal de estímulo para buses de computadora, elementos de telecomunicaciones digitales y más [10].

Los generadores de patrones digitales, pueden usar un patrón creado por el usuario o capturado, agregando degradaciones o errores donde sea necesario para aplicaciones de prueba especiales

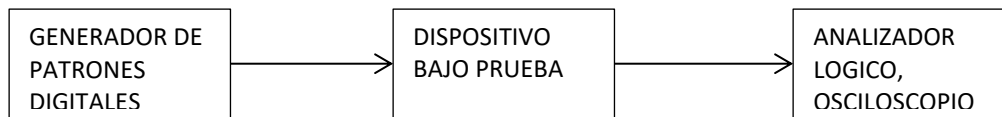
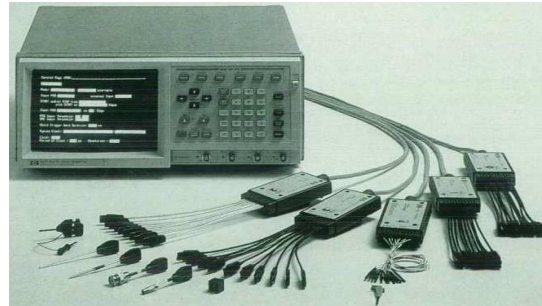


Figura 1: Diagrama a bloques de la conexión del generador de patrones con el dispositivo bajo prueba.



A) PI 5800.



B) HP8175A (ECL, TTL, CMOS).



C) DG 2020<sup>a</sup>.



D) Modulo Wave Gen Xpress.



E) IO-3208.



F) PG3A.

Figura 2: Generadores de patrones digitales en el mercado.

## **PI-5800**

El generador de patrones digitales PI-5800 se muestra en la figura 2 A). No requiere de habilidades especiales de programación. La selección del formato de salida de RZ (retorno a cero), NRZ (no retorno a cero) alta impedancia, 50  $\Omega$  y verdadero o complementario; proporcionan flexibilidad para satisfacer cualquier necesidad de prueba. El usuario tiene la capacidad de actualizar patrones sin detenerse a reprogramar. Hasta 99% de la salida de datos de un canal se puede cambiar síncronamente mientras la unidad está en el modo RUN. Los modos de bucle y anidamiento permiten un flujo de datos prácticamente sin fin. Hasta 256 sub-patrones se pueden repetir dentro del programa principal para producir patrones de más de 250 mil millones de largo [1].

## **HP 8175A**

El HP 8175<sup>a</sup>, se muestra en la figura 2 B), como generador de señales es capaz de proveer hasta tres tipos de estímulos diferentes dependiendo de la configuración seleccionada. Puede ser configurado para pulsaciones simples como generador de datos en paralelo de veinticuatro canales, o como generador de datos en serie de dos canales.

Los patrones de datos se pueden generar en 24 canales, cada uno de 1Kbits de profundidad, a 50 Mbits. Dos máquinas pueden trabajar juntas en una configuración de maestro esclavo para generar hasta 48 canales. Para superar la limitación de profundidad, dos mejoras ayudan al cliente a minimizar el número de ubicaciones de memoria utilizadas. Una es la expansión de la memoria virtual - un patrón que se va a utilizar repetidamente necesita ser almacenado sólo una vez en la máquina y puede ser incluido en el flujo de datos de salida tan a menudo como sea necesario. La otra es la duración del patrón programable cada patrón consiste en datos y parámetros de duración, por lo que las secuencias de ráfagas rápidas y largas señales de estado estacionario se pueden generar utilizando sólo unos pocos lugares de memoria.

La interacción con el dispositivo bajo prueba es facilitada por una entrada de activación, que puede indicar inicio, parada, continuación de ciclo, activación o desactivación de salidas o ramificación a otros ciclos de datos. La ramificación se puede implementar, por ejemplo, cuando un DUT requiere rutinas de inicialización. Los pines de disparo pueden asignarse para monitorear estados lógicos en varios puntos de la DUT. Una opción de sincronización fina proporciona un retardo programable en cuatro canales. El retardo puede variar de 20 ns a 40 ns con una resolución de 100 ps [2].

## **DG2020A**

El generador de patrones digitales DG2020A, ver figura 2 C) provee a los diseñadores con herramientas de alto performance para evaluar semiconductores digitales y circuitos lógicos. Cualquiera que fuera el proceso de diseño, caracterización, debug, validación o verificación. La siguiente tabla muestra las especificaciones del DG2020A [3]. EL DG2020A incluye secuencia avanzada de flujo de datos proviendo al diseñador una fuente flexible de datos digitales. Este generador de patrones puede importar datos desde un osciloscopio TDS Tektronix gracias a su software DG-Link.

### **Wave Gen Xpress 8MB - 100 MHz 16-bit**

El modulo Wave Gen Xpress, se muestra en la figura 2 D), es un dispositivo de salidas digitales para la generación de vectores de datos de 16 bits. Este rápido generador de señales digitales permite un funcionamiento hasta de 100MHz con precisión de ciclo. Tal funcionalidad es esencial para estimular las entradas de dispositivos electrónicos y tarjetas.

Como una unidad compacta con un bufer de memoria grande de 8 MB controlado desde la PC por USB, es el complemento ideal para el FPGA u otra tarjeta de prototipo de computación embebida. WG Xpress Digital Pattern Generator apunta a pruebas funcionales en placas de prototipo, circuitos integrados y otros sistemas integrados digitales [4].

### **IO-3208, Logic Analyzer, Pattern Generator**

El IO-3208 se muestra en la figura 2 E). Todos los IO-3200 funcionan como Analizador Lógico y Generador de Patrones. Se comunican y funcionan con USB. El analizador lógico puede muestrear hasta 200 MSa / s, y el generador de patrones puede operar a hasta 200 MSa / s. Cuentan con un buffer que es de hasta 2 Mpt de largo. Tienen 32 canales de analizador lógico y hasta 8 canales de generador de patrones.

Cada canal del Generador de patrones también es un canal del Analizador lógico, que puede alternar entre el modo de Entrada y el de Salida durante una sola captura. Esto le permite estimular su circuito y registrar los resultados en un solo pin. Es como tu propio mini probador de cama de alfileres. Los potentes asistentes de generación de datos lo ayudan a crear contadores, relojes, protocolos de comunicación SPI e I2C, etc. El software también le permite cargar sus propios archivos de datos [5].

### **PG3A**

El generador de patrones digitales PG3A, ver figura 2 F), es una herramienta de propósito general que puede ser usada para estimular las tarjetas electrónicas. Con el PG3 se tiene flexibilidad y expansión con diferentes factores de forma y una amplia selección de sondas para una amplia gama de aplicaciones.

Este generador de patrones digitales se puede utilizar para emulación y estimulación de periféricos /ASIC, prueba de nivel de protocolo, verificación de espera, prueba de producción, prueba de señal mixta y estímulo digital general. El PG3 se puede acoplar con un analizador lógico Tektronix y/o un osciloscopio digital Tektronix para proporcionar un sistema completo de pruebas [6].

Tabla 1: Tabla comparativa de los generadores de patrones digitales.

Generador de patrones digitales	Velocidad del dato	Profundidad del patron	Tiempo de subida/bajada (20% to 80%)	No. Canales	Precio
---------------------------------	--------------------	------------------------	--------------------------------------	-------------	--------

DGPMCEA	100Mb/s	4KB	3ns at $5V_{p-p}$	24	
PI 5800	50 Mb/s	4 K/Ch.	6ns at $5V_{p-p}$	16.	
HP 8175A	50 Mb/s	1 K/Ch.	3ns at $5V_{p-p}$	24	\$340
DG2020A	200 Mb/s	64 K/Ch	1.4ns at $5V_{p-p}$	12,24 o 36.	\$1250
Wave Gen Xpress	100 Mb/s	8Mb	2ns at $5V_{p-p}$	16	\$512.42
IO-3208	200 Mb/s	256kpt/Ch.	2.7ns at $5V_{p-p}$	8	\$750
PG3A	300 Mb/s	32 Mb	2ns at $5V_{p-p}$	64	\$7,270

El tema de la automatización es un objetivo estratégico para nuestro país y por ende de vital importancia para el estado de Puebla. Actualmente, el equipo electrónico asociado a la innovación tecnológica es tecnología de importación. La dependencia tecnológica y el elevado costo del equipo hacen complicada y difícil la disponibilidad de equipar laboratorios de electrónica y robótica. Por lo tanto, no es un proceso trivial llevar equipo de laboratorio de calidad a la sociedad.

El diseño de tarjetas de adquisición y control tienen un alto nivel de complejidad, por lo que en su mayoría son desarrollados por compañías transnacionales en las cuales un equipo multidisciplinario realiza los diseños, estos son de gran valor tanto para la compañía que los realiza como para su país. La tarjeta que se propone en este trabajo de tesis es de arquitectura abierta. Dentro de esta tarjeta se desarrollara un firmware que realice las funciones de un instrumento de medición que en este caso es un generador de patrones digitales. La tecnología avanza a grandes pasos, por lo que la investigación de dispositivos de última generación se vuelve de vital importancia.

En el diseño de circuitos y sistemas digitales se requieren realizar pruebas con diferentes datos a diferentes frecuencias, con el propósito de verificar que el circuito integrado o sistema digital cumpla con las especificaciones del diseño. Así como también, se requieren realizar las mismas pruebas con el objetivo de conocer el funcionamiento de un nuevo sistema. Para lograr esto, se requiere de un generador de datos digitales.

Los beneficios que proporciona un generador de datos en el diseño digital son muy grandes, debido a la gran variedad de aplicaciones en la instrumentación orientada a la robótica, automatización, sistemas de adquisición, etcétera.

El objetivo general y los objetivos particulares del tema de tesis se mencionan a continuación:

#### **OBJETIVO GENERAL**

“Diseñar e implementar un generador de patrones digitales de 2 canales de 12 bits c/u, programables a 50MHz “.

#### **OBJETIVOS PARTICULARES**

- Revisar los antecedentes del tema.
- Realizar el diseño del generador de patrones digitales.
- Diseñar la tarjeta de adquisición.
- Desarrollar el firmware necesario para el generador de patrones digitales.
- Implementar la interfaz.
- Diseñar el software del instrumento amigable al usuario.
- Realizar pruebas del sistema diseñado.
- Publicación de resultados.
- Escritura de Tesis

En el capítulo 1 se realiza una breve descripción del sistema, haciendo uso del diagrama general del sistema, así como de los tipos de señales digitales y las aplicaciones básicas de un generador de patrones digitales.

En el capítulo 2 se mencionan los componentes que forman el hardware del generador de patrones digitales y una descripción breve del funcionamiento de cada uno de ellos. El cual está formado principalmente por una tarjeta embebida FPGA, un módulo WiFi, convertidor digital analógico (DAC) y la etapa de acondicionamiento de señal.

En el capítulo 3 se presenta el diseño del firmware del generador de patrones digitales a implementar en el FPGA, se aborda la explicación de cada uno de los bloques usados en el firmware, describiendo su funcionamiento con la ayuda de diagramas de flujo, diagramas a bloques y principalmente máquinas de estado.

En el capítulo 4 se presenta y menciona la interfaz de usuario del generador de patrones digitales; el software de la interfaz se desarrolló mediante la herramienta de instrumentación virtual LabView; en cada subcapítulo se desarrolla la descripción de las opciones de configuración del generador de patrones digitales, la descripción de su funcionamiento haciendo uso de diagramas de flujo y diagramas de bloques.

En el capítulo 5 se abordan los resultados obtenidos del diseño e implementación del generador de patrones digitales, el cual mediante la integración del hardware, firmware y software, se tiene un instrumento capaz de generar señales eléctricas discretas a diferentes niveles de voltaje.

Finalmente en el capítulo 6 se dan las conclusiones generales. En los apéndices se mencionan, la participación en el congreso, el artículo y por último el manual de usuario del dispositivo.

# CAPITULO 1: Descripción del sistema

En el diseño de circuitos y sistemas digitales se requieren realizar pruebas con diferentes datos a diferentes frecuencias, con el propósito de verificar que el circuito integrado o sistema digital cumpla con las especificaciones del diseño. Además, se requiere realizar las mismas pruebas con el objetivo de conocer el funcionamiento de nuevos sistemas que son introducidos al mercado. Estas pruebas se logran hacer con un generador de patrones digitales.

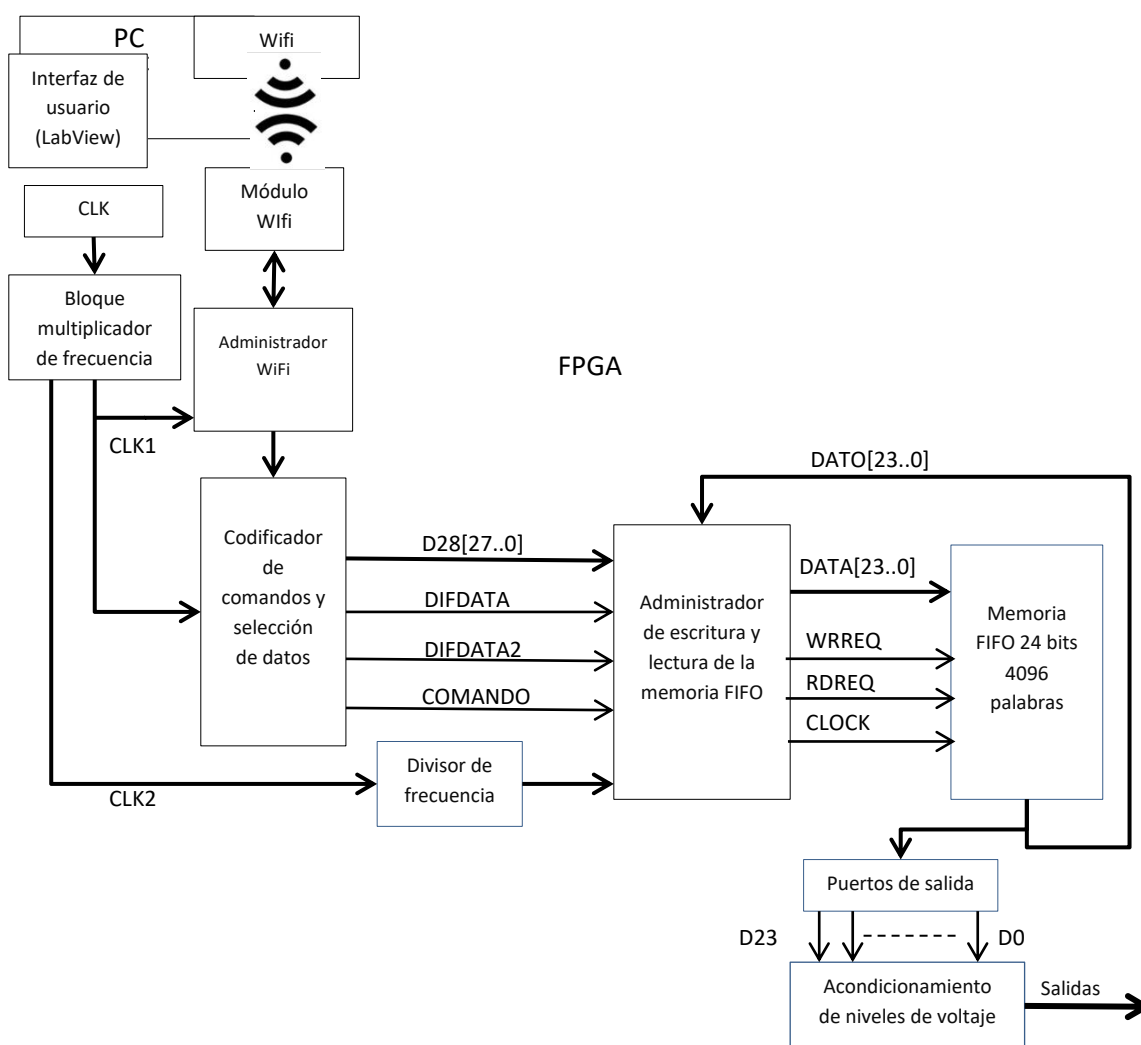


Figura 1.1: Diagrama general del generador de patrones digitales.

Estos aparatos generan una serie de patrones digitales (datos) los cuales son aplicados al sistema que se requiere probar. En cada caso los patrones generados deben satisfacer las condiciones de la prueba a efectuar. Como parte de esta aplicación se desarrollará un software interactivo para la programación del generador de datos digital. Es decir, un ambiente en donde el usuario se sienta cómodo y pueda interactuar con el instrumento.

La figura 1.1 muestra la estructura general del generador de patrones digitales, mostrando la conexión entre la PC y el FPGA, así como los bloques del firmware del generador de patrones.

### 1.1 Diagrama general del sistema

En el módulo del PC se lleva a cabo el desarrollo e implementación de la interfaz de usuario del generador de patrones digitales, la cual se desarrolló en el software LabVIEW y hace uso del puerto WiFi de la computadora, para mandar los datos del patrón digital a la tarjeta del generador de patrones digitales, figura 1.2.

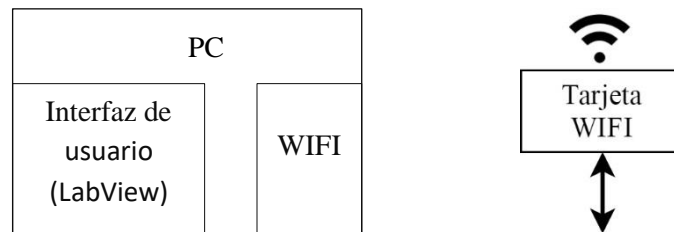


Figura 1.2: Módulo PC y módulo de la tarjeta wifly

La captura del patrón digital se realiza en dos arreglos, en los cuales se ingresan los números binarios de 12 bits en cada uno. Cada arreglo representa un canal de salida del generador de patrones digitales y cada bit del arreglo es una salida del generador de patrones digitales.

En la selección del tipo de reproducción (única o repetitiva) se muestran los botones, de los cuales se debe seleccionar uno de los dos, indicando el tipo de reproducción, donde si es única, entonces el patrón digital solo se reproduce una vez, pero si es repetitiva entonces el patrón se repite indefinidamente, hasta que el usuario lo detenga.

La frecuencia de reproducción es configurable desde la interfaz del usuario teniendo la opción de modificarla en un intervalo de 0.1Hz a 50 MHz.

Habiendo capturado el patrón digital, seleccionado el tipo de reproducción y configurado la frecuencia; se procede a graficar el patrón digital y el envío de los datos capturados, por el puerto WiFi de la computadora hacia el módulo WiFi de la tarjeta del generador de patrones digitales.

En la tarjeta del generador de patrones digitales, como componentes principales, tenemos la tarjeta WiFi y el FPGA. La tarjeta WiFi recibe los datos enviados por el puerto WiFi de la computadora y los muestra en formato serie, en su salida Rx, los cuales son enviados al FPGA ya que cuentan con el nivel de voltaje adecuado (3.3V) para la conexión con el FPGA. El reloj de la tarjeta del FPGA es de 100Mhz, el cual es necesario para la sincronización y funcionamiento de la tarjeta.

Al ser el FPGA un campo de arreglos de compuertas programables; por sí solo no es posible que se comporte como un generador de patrones digitales; necesita un firmware que le diga cómo debe de funcionar. Dentro del FPGA tenemos el bloque “multiplicador de frecuencia,” el cual se encarga de multiplicar la frecuencia, en un factor de dos, del reloj de la tarjeta del generador de patrones digitales.

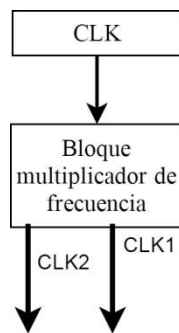


Figura 1.3: Señal de reloj y módulo multiplicador de frecuencia.

Los 200 MHz de salida del módulo multiplicador de frecuencia, figura 1.3, son utilizados en el módulo divisor de frecuencia, el cual se encarga de crear la frecuencia de reproducción de la memoria FIFO, que ha sido configurada en la interfaz de usuario.

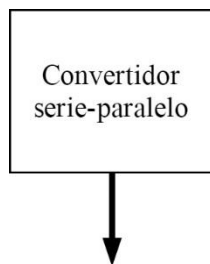


Figura 1.4: Módulo convertidor serie-paralelo.

El bloque “convertidor serie-paralelo”, figura 1.4, recibe los datos de la tarjeta WiFi en formato serie y los convierte a paralelo, para su posterior manipulación.

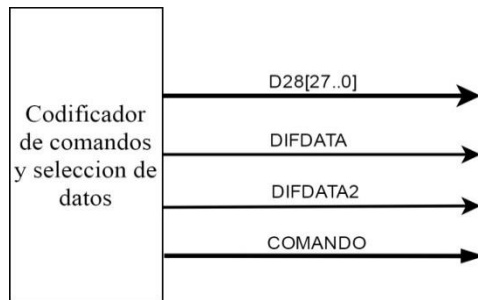


Figura 1.5: Módulo codificador de comandos y selección de datos.

En el bloque “Codificador de comandos y selección de datos”, figura 1.5, se encarga de verificar que el byte presente en sus entradas represente uno de los comandos asignados al sistema, para posteriormente unirlos con los siguientes 4 bytes y, finalmente, dependiendo del comando recibido; proceder a seleccionar los datos recibidos, ya que en los 4 bytes recibidos se encuentra D28, DIFDATA y DIFDATA2. Las salidas COMANDO, D28 y DIFDATA; son enviadas tanto al divisor de frecuencia como al bloque de control de lectura y escritura de la memoria FIFO.

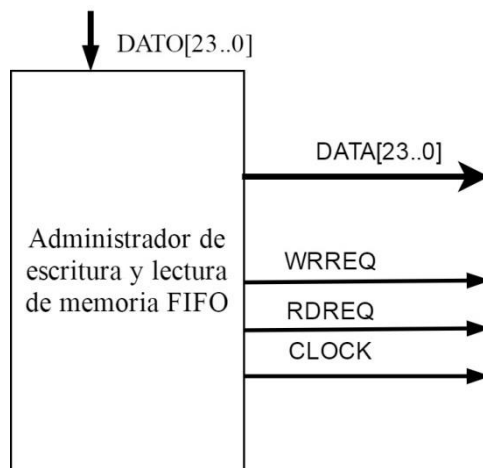


Figura 1.6: Módulo administrador de la memoria FIFO.

El “Administrador de escritura y lectura de la memoria FIFO”, figura 1.6, dependiendo del comando recibido, procede a borrar la memoria, escribir la memoria o realizar la lectura y escritura de los datos almacenados en la memoria FIFO; con la intención de mantener la información en la memoria FIFO y tener la posibilidad de leer la misma información las veces necesarias.

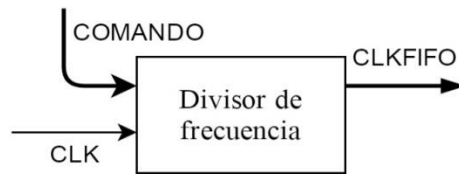


Figura 1.7: Módulo divisor de frecuencia.

El bloque “divisor de frecuencia”, figura 1.7, se encarga de reproducir la señal de reloj a la frecuencia configurada en la interfaz de usuario y así reproducir los datos almacenados en la memoria FIFO a la frecuencia deseada por el usuario.

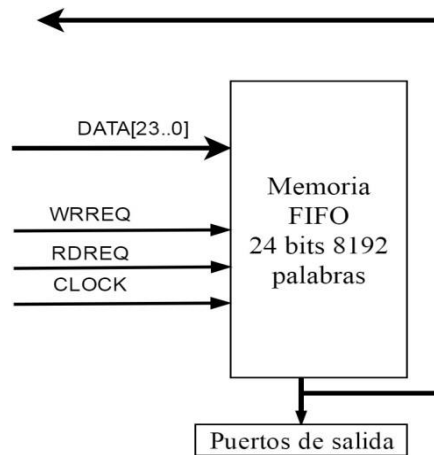


Figura 1.8: Memoria FIFO de 8KBytes.

La “memoria FIFO”, figura 1.8, se encarga de almacenar el patrón digital; que ha sido enviado desde la interfaz de usuario; para su posterior reproducción.

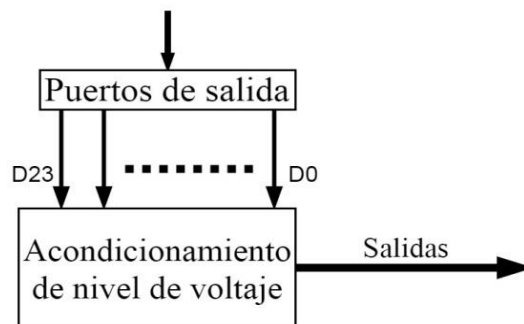


Figura 1.9: Módulo para el acondicionamiento del nivel de voltaje.

Finalmente tenemos el bloque de “acoplamiento de nivel de voltaje”, figura 1.9. Debido a que existen diferentes tipos de tecnologías digitales como TTL, CMOS, ECL y para tener una amplia funcionalidad es necesario acoplar el nivel de voltaje de salida del FPGA a estas tecnologías y haciendo que el sistema sea compatible y aumente su funcionalidad.

## 1.2 Tipos de tecnologías digitales

Las señales digitales tienen diferentes niveles de voltaje para representar un 1 o un 0 lógico, el cual depende de la tecnología del circuito digital; como se muestra en la tabla 1. Los diferentes niveles de voltaje se muestran en la figura 10 [7] -[9],[27],[28].

Tabla 1.1: Tipos de señales digitales.

TECNOLOGÍA	VOLTAJE
<b>a. TTL</b> Lógica de Transistor a Transistor	1:5V 0:0V
<b>b. LVCMOS</b> Semiconductor de metal oxido de bajo voltaje	1:3.3 0:0-0.8V
<b>c. PECL</b> Lógica positiva de emisor acoplado.	1: $V_{CC}-0.8$ 0: $V_{CC}-1.8$
<b>d. ECL</b> Lógica de emisor acoplado	1:-0.8 0:-1.8
<b>e. LVDS</b> <i>Señal diferencial de bajo voltaje</i>	$\Delta V = V_+ - V_-$ donde $\Delta V [150mV, 500mV]$

Gráficas de tecnologías digitales.

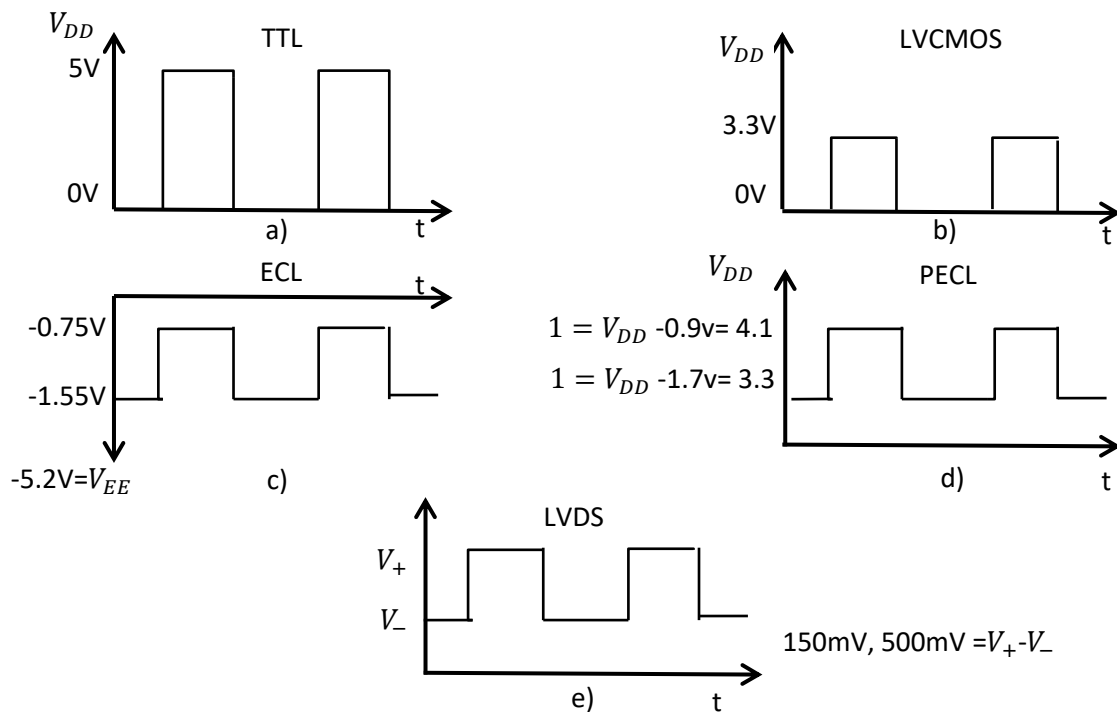


Figura 1.10: Gráficas de las diferentes tecnologías digitales.

### 1.3 Aplicaciones del generador de patrones digitales

El generador de patrones digitales tiene cientos de diferentes aplicaciones pero en el contexto de mediciones electrónicas se tienen tres diferentes categorías: verificación, caracterización y prueba de margen de estrés. [10]

- Verificación

Prueba digital de módulos de transmisores y receptores. En el desarrollo y diseño de equipos inalámbricos, el hardware de transmisores y receptores deben de ser simulados para verificar conformidad con emergentes y propiedades estándares inalámbricos. Verificación de la ausencia de cortos en buses de datos. Comprobación de crosstalk en circuito impreso. En el desarrollo e implementación de protocolos de comunicación.

- Caracterización

En el desarrollo de nuevos convertidores digital-analógicos (DAC) y convertidores analógico-digitales (ADC) pues deben ser exhaustivamente testados para determinar sus límites de linealidad, monotonía y distorsión. En el diseño de circuitos impresos, para observar su capacidad de transmisión de datos sin tener distorsión o atenuación alguna.

- Prueba de estrés/margen

Estrés de receptores de comunicación. Los generadores de señales ayudan al diseñador a buscar errores en el funcionamiento de sistemas digitales desde dispositivos hasta sistemas completos, lo cual produce un gran ahorro de tiempo. También nos ayuda a reproducir rápidamente los diagramas de tiempos para caracterizar los dispositivos. Creación de errores o violaciones de tiempos para observar el comportamiento del sistema digital así como su recuperación.

- Emulación

Otra prueba consiste en la emulación de sensores o tarjetas digitales, para engañar al control digital y observar su comportamiento en ciertas situaciones del sistema o planta a controlar. Emulación de la respuesta de un control digital para una planta a controlar. En el caso de los robots rotrari; si se le colocara un convertidor digital analógico, para mandar la señal a los drivers del robot; se podría mandar una secuencia de números que representen las diferentes posiciones del efector final. En el caso automotriz podemos emular ciertas señales digitales provenientes de algunos sensores o subsistemas como: sensor de cuerpo de aceleración, sensor de revoluciones, árbol de levas, control abs, sistema de verificación de luces exteriores.

### 1.4 Conclusiones

Para poder desarrollar un dispositivo o sistema es necesario definir el funcionamiento del mismo, investigar si existe alguno igual o parecido en el mercado; en dado caso que existiera, investigar sus características y funcionamiento, para luego indagar si lo propuesto por el diseñador es correcto o hay que hacer correcciones, así como proponer alguna característica que lo haga atractivo al público al que está dirigido, así como hacerle mejoras o alguna diferencia respecto a los existentes en el mercado. En el siguiente capítulo se describe el hardware del generador.

## CAPITULO 2: Hardware del generador de patrones digitales

En este capítulo se describen las etapas que conforman la instrumentación electrónica del generador patrones digitales. Los principales componentes son la interfaz empleada para la comunicación WiFi, la tarjeta embebida FPGA y la etapa de acoplamiento de voltaje, con el objetivo de obtener un patrón de señales digitales con el nivel de voltaje adecuado.

El diagrama electrónico consta de un módulo WiFi (Wifly), una tarjeta de desarrollo comercial de DEO-NANO; de la marca TERAasic; con el FPGA Cyclone IV EP4CE22F17C6 de la compañía ALTERA [ ], al cual se le programa el firmware del generador de patrones digitales, este firmware consta de una memoria FIFO donde se almacenan los datos del patrón digital, los cuales serán enviados por la computadora para la reproducción del patrón digital elegido o creado por el usuario.

Otra característica del generador de patrones digitales es la capacidad de adaptar el nivel de voltaje de los datos digitales para que estos se acoplen al nivel de voltaje de la tecnología usada en el sistema a verificar, caracterizar o someter a prueba de margen de estrés. Para la realización de este circuito de acoplamiento de señal se realiza la caracterización de convertidor analógico digital (DAC) y de esta manera conocer el intervalo, resolución y los valores necesarios para el acoplamiento de señal según sea el tipo de tecnología a aplicar por el usuario.

Como fuente de alimentación se propone una fuente simétrica de 5V y -5V, la cual se va a encargar de alimentar a los operacionales y al DAC del circuito de acondicionamiento de señal.

Otro aspecto importante es el diseño del circuito impreso, pues las pistas de las 24 señales del generador de patrones digitales deben de cumplir con las características de una guía de onda para que al momento de propagar la señal a la frecuencia elegida por el usuario, esta se propague correctamente, a través del circuito impreso y llegue al dispositivo bajo prueba sin alguna distorsión o pérdida de la señal.

Esta es la razón por la cual se lleva un análisis de las pistas del circuito impreso para que estas cumplan con la característica de guía de onda de microcinta y así propagar la señal sin pérdidas o atenuaciones. En el diseño de las pistas se considera a propagar una señal a 200MHz, la cual es 4 veces mayor a la frecuencia máxima del generador de patrones digitales y con ello se pretende no tener problemas a una frecuencia de 50MHz. La figura 2.1 muestra el diagrama eléctrico del generador de patrones digitales.

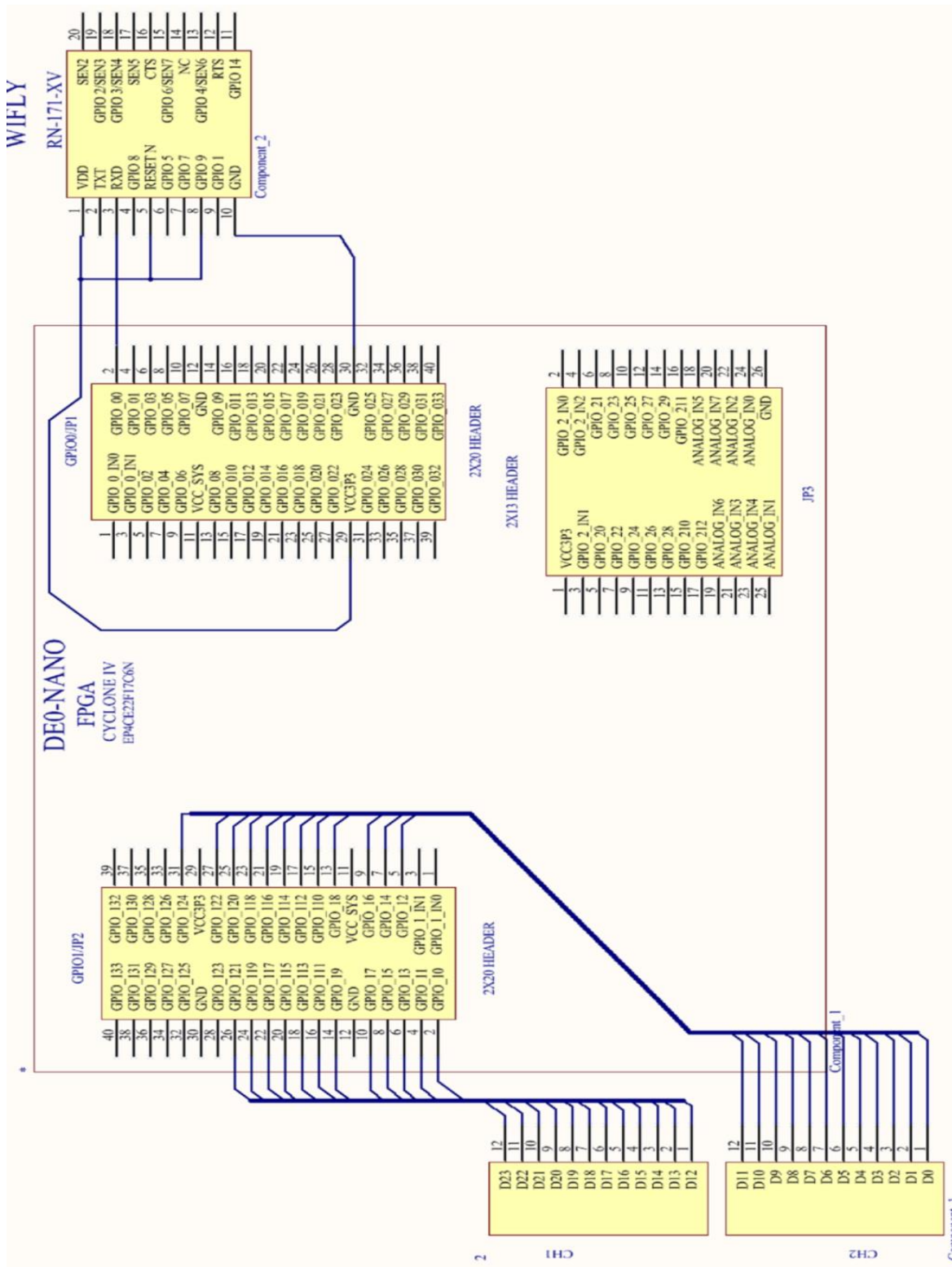


Figura 2.1: Diagrama electrónico del generador de patrones digitales.

## 2.1 Descripción general del hardware

El hardware consta de diferentes etapas, en primer lugar, la tarjeta WiFi recibe los datos enviados por el puerto WiFi de la computadora. Estos datos son enviados a la tarjeta DEO-NANO; dicha tarjeta cuenta con un FPGA, la cual brinda amplias ventajas en el diseño del generador patrones digitales, cumpliendo las funciones de divisor de frecuencia, almacenamiento y reproducción de los datos digitales. Finalmente se tiene la etapa de acoplamiento de nivel de voltaje, en donde se emplean un conjunto de operacionales de alta velocidad para ajustar el nivel de voltaje de las señales digitales a la frecuencia seleccionada por el usuario. La figura 2.2 muestra las etapas que forman el hardware del generador de patrones.

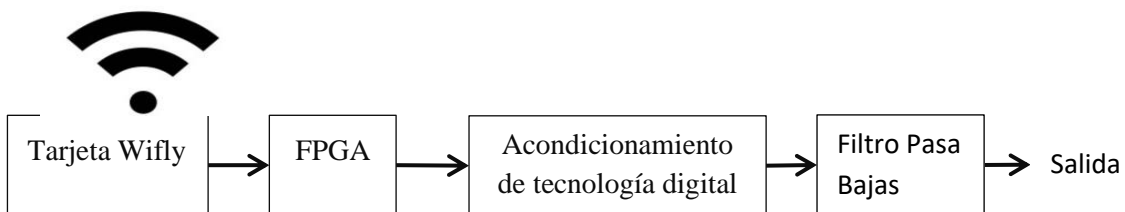


Figura 2.2: Diagrama general del hardware.

## 2.2 Tarjeta de comunicación módulo RN-XV

El módulo WiFi RN-171-XV, figura 2.3, de la empresa Roving Networks es la interfaz de comunicación WiFi empleada para la recepción inalámbrica de los datos del patrón digital, capturados en la PC. Esta tarjeta viene adaptada para ser compatible con el módulo XBee. Este diseño facilita su uso, ya que el RN-171, por sí sólo, es de montaje superficial y su método de soldadura es delicado, además de requerir el diseño de una PCB para su montaje. También cuenta con una antena integrada a la PCB, LEDs indicadores y una distribución de pines que, aunque no corresponde al paso de pines comúnmente usado (como por ejemplo una protoboard), se adquirió una PCB Xbee Breakout de Sparkfun que adapta el paso entre pines de XBee al estándar de 2.54 mm. [11],[22]. Los pines del módulo WiFly se describen en las figura 2.4.

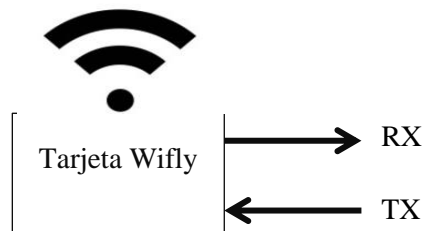
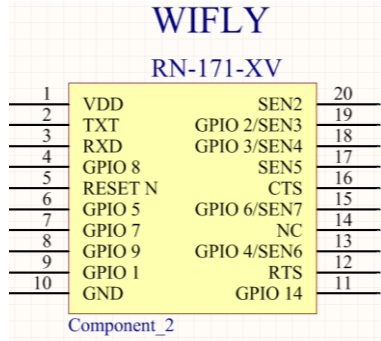


Figura 2.3: Módulo WiFly RN-171-XV.



Características	
Alimentación	3.0 a 3.7 V
Frecuencia	2.4GHz
Protocolo de Red	TCP/IP
Protocolo de Datos	UART

Figura 2.4: Pines de comunicación y características del módulo WiFi RN-XV.

### 2.3 Tarjeta embebida DE0-NANO

Los Arreglos de Compuertas Programables por Campo (FPGA's por sus siglas en inglés) son dispositivos que están compuestos por una matriz elementos lógicos programables y tienen una alta densidad de estos elementos. La programación de dichos dispositivos define la manera en que funciona cada bloque lógico. Los FPGA's a diferencia de los microcontroladores y microprocesadores no son secuenciales, sino que son circuitos combinacionales, ya que sus salidas dependen de entradas que provienen de otros elementos lógicos y tampoco son de propósito general, sino específico.

La programación se realiza en el software Quartus II, ya que es el sistema de desarrollo más reciente de Altera, el cual ofrece características más avanzadas y soporta los nuevos dispositivos PLD, tales como la familia Cyclone de FPGAs, que se encuentran en muchas tarjetas educativas recientes. La estructura general de la tarjeta DE0-NANO se muestra en la figura 2.5.

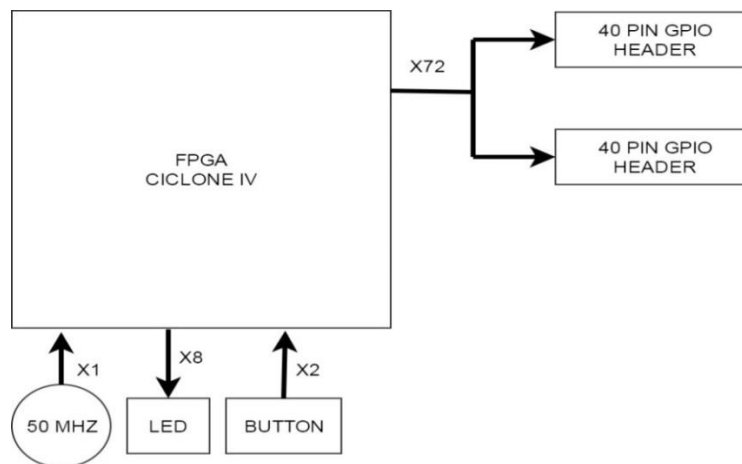


Figura 2.5: Diagrama de bloques de la tarjeta DE0-NANO.

DE0-NANO FPGA CYCLONE IV/ EP4CE22F17C6N

GPIO-1						GPIO-2						GPIO-0						JP1					
40	J14	IO	IO	J13	39	Signal	FPGA	Descirpcion	1	A8	IO	IO	D3	2	1	A8	IO	IO	D3	2			
38	K15	IO	IO	J16	37	name	Pin		3	B8	IO	IO	C3	4	3	B8	IO	IO	C3	4			
36	L13	IO	IO	M10	35	No.			5	A2	IO	IO	A3	6	5	A2	IO	IO	A3	6			
34	N14	IO	IO	L14	33	CLK	R8	Clock	7	B3	IO	IO	B4	8	7	B3	IO	IO	B4	8			
32	P14	IO	IO	N15	31	Key0	J15	Button	9	A4	IO	IO	B5	10	9	A4	IO	IO	B5	10			
30	GND			VCC 3P3	29	Key1	E1	Button	11	VCC_SYS			GND	12	11	VCC_SYS			GND	12			
28	N16	IO	IO	R14	27	SW0	M1	Switch	13	A5	IO	IO	D5	14	13	A5	IO	IO	D5	14			
26	P16	IO	IO	P15	25	SW1	T8	Switch	15	B6	IO	IO	A6	16	15	B6	IO	IO	A6	16			
24	L15	IO	IO	R16	23	SW2	B9	Switch	17	B7	IO	IO	D6	18	17	B7	IO	IO	D6	18			
22	K16	IO	IO	L16	21	SW3	M15	Switch	19	A7	IO	IO	C6	20	19	A7	IO	IO	C6	20			
20	N11	IO	IO	N9	19	LED0	A15	LED	21	C8	IO	IO	E6	22	21	C8	IO	IO	E6	22			
18	P9	IO	IO	N12	17	LED1	A13	LED	23	E7	IO	IO	D8	24	23	E7	IO	IO	D8	24			
16	R10	IO	IO	P11	15	LED2	B13	LED	25	E8	IO	IO	F8	26	25	E8	IO	IO	F8	26			
14	R11	IO	IO	T10	13	LED3	A11	LED	27	F9	IO	IO	E9	28	27	F9	IO	IO	E9	28			
12	GND			VCC_SYS	11	LED4	D1	LED	29	VCC 3P3			GND	30	29	VCC 3P3			GND	30			
10	T11	IO	IO	R12	9	LED5	F3	LED	31	C9	IO	IO	D9	32	31	C9	IO	IO	D9	32			
8	T12	IO	IO	R13	7	LED6	B1	LED	33	E11	IO	IO	E10	34	33	E11	IO	IO	E10	34			
6	T13	IO	IO	T14	5		L3	LED	35	C11	IO	IO	B11	36	35	C11	IO	IO	B11	36			
4	T15	IO	IO	R9	3	GPIO-2						37	A12	IO	IO	D11	38	37	A12	IO	IO	D11	38
2	F13	IO	IO	T9	1	26	GND	AN_1	25	39	D12	IO	IO	E12	40	39	D12	IO	IO	E12	40		
						24	AN_0	AN_4	23														
						22	AN_2	AN_4	21														
						20	AN_7	AN_6	19														
						18	AN_5	IO	G15	17													
						16	G16	IO	F14	15													
						14	F16	IO	F15	13													
						12	D14	IO	D15	11													
						10	D16	IO	C15	9													
						8	C16	IO	C14	7													
						6	B16	IO	A14	5													
						4	M16	IO	E16	3													
						2	E15	IO	VCC3P3	1													

Figura 2.6: Pines de la tarjeta DE0-NANO.

La organización de los pines de la tarjeta DE0-NANO se muestra en la figura 2.6.

La tarjeta DE0-NANO de la empresa Terasic; aunque sus características son básicas, cuenta con una gran capacidad para el desarrollo proyectos; sus características principales son [12] [13]:

- FPGA Altera Cyclone IV EP4CE22F17C6N
- 153 pines de entrada/salida.
- USB-Blaster en la tarjeta para su programación.
- Dos filas de 40 pines de propósito general (GPIO), pin de 5 V, 2 pines de 3.3 V y cuatro pines de GND.
- 32 MB SDRAM.
- 2 KB I2C EEPROM.
- 8 leds de color verdes.
- Pushbuttons.

- Dip switch.
- Sensor acelerómetro ADI ADXL345 con resolución de 13 bits.
- ADC NS ADC128S022 12 bits, 8 canales.
- Cristal 50 MHz.
- Fuente de alimentación: puerto tipo USB (5 V), un pin de 5V por cada fila de GPIO, 2 pines externos de 3.6 V-5.7 V.

La figura 2.7 muestra los componentes y conexiones de la tarjeta DE0-NANO.

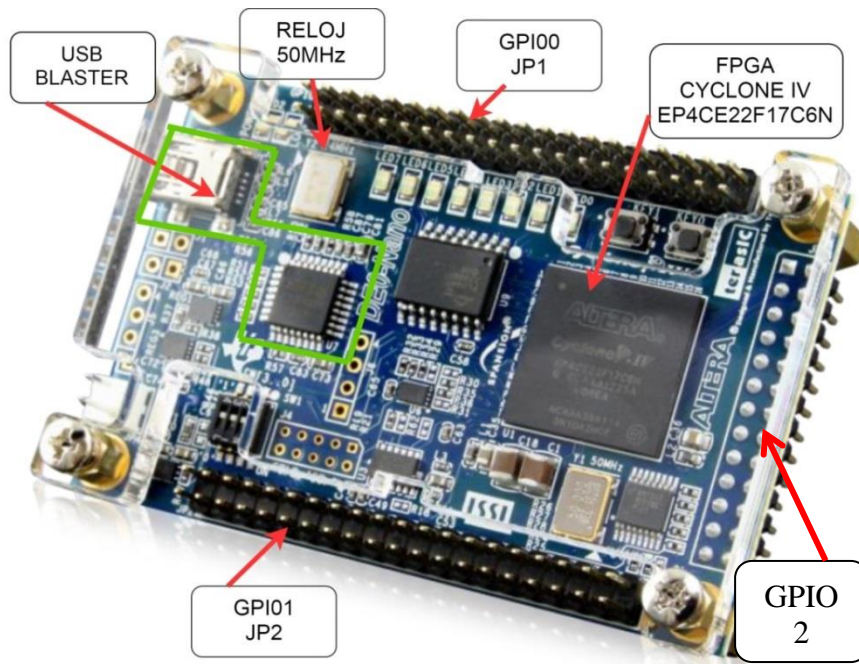


Figura 2.7: Conexiones de la tarjeta DE0-NANO.

## 2.4 Etapa de acondicionamiento de las salidas digitales

El diagrama de la figura 2.8 es la parte encargada de acoplar el nivel de voltaje aportado por el FPGA, el cual es de 3.3V al voltaje apropiado de alguna de las tecnologías digitales antes mencionadas. De esta manera el generador de patrones digitales es compatible con diferentes tecnologías y podemos hacer uso de el para probar dispositivos que trabajen con alguna o algunas tecnologías digitales. La ecuación (2.1) del circuito acoplador de voltaje se muestra a continuación:

$$V_{out} = -\left(\frac{R_f}{R_2} V_{in1} + \frac{R_f}{R_3} V_{in2}\right) \quad (2.1)$$

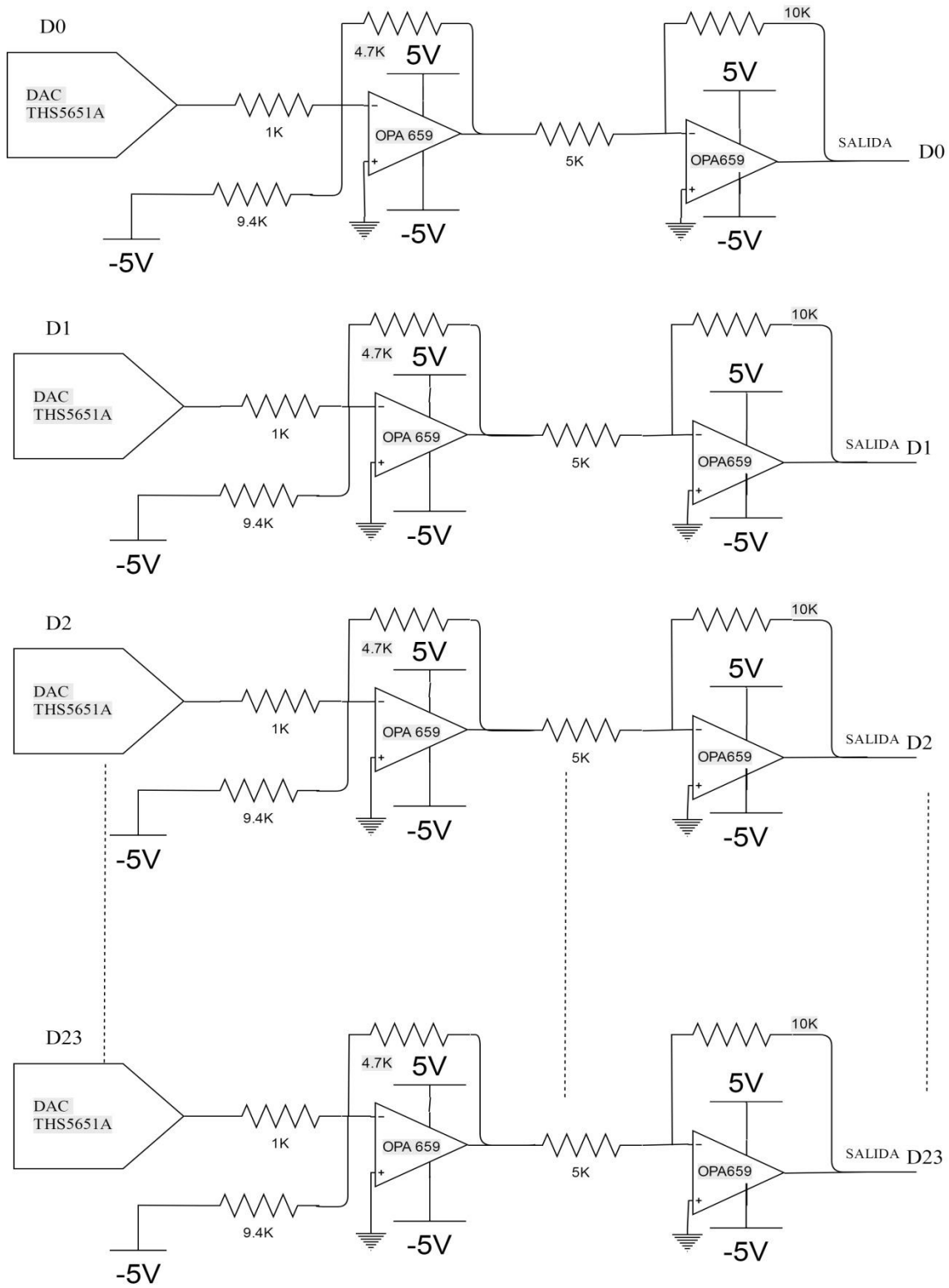


Figura 2.8: Diagrama de acoplamiento para diferentes tecnologías digitales.

### 2.4.1 DAC THS5651A

El THS5651A es un convertidor digital analógico con una resolución de 10 bits, específicamente optimizado para transmisión de datos en sistemas de comunicación de manera alámbrica e inalámbrica. Este opera con un voltaje de alimentación de 3V a 5.5V, haciendo posible la compatibilidad con tecnología CMOS (de 3V y 5V). A escala completa este provee una corriente diferencial de 20 mA. La corriente de salida puede ser aplicada directamente a la carga, sin un búfer externo adicional [18].

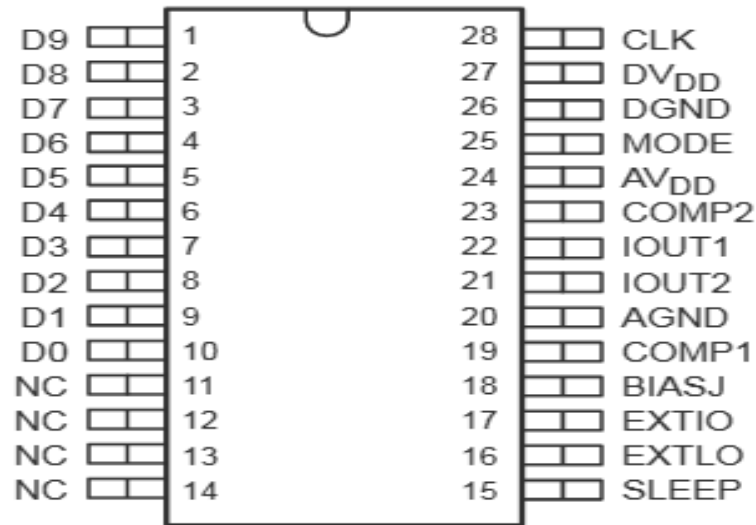


Figura 2.9: Descripción de pines del DAC THS5651A.

La figura 2.9 muestra el diagrama de pines del integrado en encapsulado SOIC. El pin 25 “MODE” configura el tipo de dato a recibir en sus entradas digitales; (véase tabla 2.1); pues si MODE=1 el dato recibido es complemento a dos, si MODE=0 se recibe los datos directamente en formato binario.

Tabla 2.1: Modos de la interfaz de entrada.

FUNCIÓN / MODE	MODE 0	MODE 1
	PIN MODE CONECTADO A DGND	PIN MODE CONECTADO A $DV_{DD}$
FORMATO DE ENTRADA	BINARIO	COMPLEMENTO A DOS

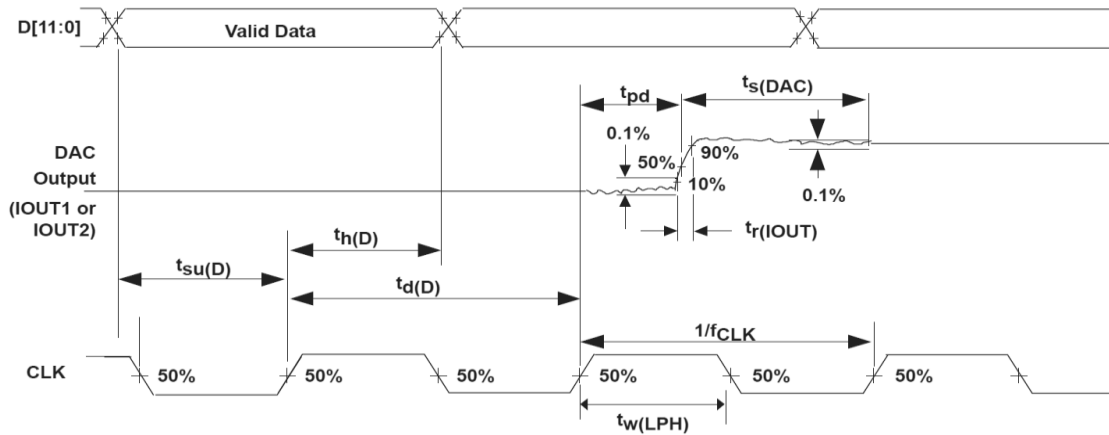


Figura 2.10: Diagrama de tiempos del DAC THS5651A.

La figura 2.10 muestra el diagrama de tiempos a cumplir para que el DAC realice la conversión digital analógica.

Tabla 2.2: Descripción de los tiempos del diagrama de tiempo.

PARAMETRO	MIN	TIPICO	MAX	UNIDAD
TIEMPO				
$t_{su}(D)$ tiempo de configuración de entrada	1			ns
$t_h(D)$ tiempo presente de la entrada	1			ns
$t_w(LPH)$ tiempo de espera de la entrada	4			ns
$t_d(D)$ tiempo de retardo digital			1	ns
$t_r(IOUT)$ tiempo de subida de la salida de 10% al 90%		2		ns
$t_f(IOUT)$ tiempo de bajada de la salida de 90% al 10%		2		ns
$t_s(DAC)$ tiempo de estabilización de la salida a 0.1%		35		ns
$t_{pd}$ retardo de propagación de la salida a 0.1%		2		ns

La tabla 2.2 describe los tiempos a cumplir del diagrama de tiempo del DAC THS5651A [18].

La función de transferencia se expresa en la siguiente ecuación:

$$IOUT1 = IOUT_{FS} * \frac{PALABRA}{4096} \quad (2.2),$$

donde  $IOUT_{FS}$  es la corriente a escala completa, PALABRA es la representación decimal de la palabra a la entrada del DAC. Las corrientes de salida son manejadas por una resistencia de carga  $R_{LOAD}$ , esto se traduce a  $VOUT1$  y  $VOUT2$ , en las terminales IOUT1 e IOUT2 según sea el caso.

$$V_{OUT1} = I_{OUT1} * R_{LOAD} = I_{OUT_{FS}} * \frac{PALABRA}{4096} * R_{LOAD} \quad (2.3)$$

#### 2.4.1.1 Referencia de operación

La corriente máxima de salida se ajusta por la aplicación de una resistencia externa. La corriente bias a través de la resistencia está definida por la referencia de voltaje bandgap y el control de amplificación. La corriente a escala completa es igual a treinta y dos veces la corriente bias,

$$I_{OUT_{FS}} = 32 * I_{BIAS} = \frac{32 * V_{EXTIO}}{R_{BIAS}} \quad (2.4)$$

donde  $V_{EXTIO}$  es el voltaje en la terminal EXTIO. La referencia de voltaje bandgap entrega un voltaje de 1.2V. Esta referencia se activa cuando la terminal EXTLO es conectada a GND, de esta manera el DAC opera con la referencia interna.

#### 2.4.1.2 Conexión del DAC con el FPGA

Debido a que el DAC usa 10 bits y una señal de reloj para mostrar el valor de la conversión a la salida, se decidió por usar los 5 bits más significativos, disminuyendo el número de pines necesarios realizar la conversión de digital a analógico. Esta configuración nos reduce la resolución de salida a 40 mV por cada incremento en estos 5 bits. Lo cual no es un problema para alcanzar los valores deseados TTL, CMOS, ECL, PECL. En total se utilizan 6 bits por salida; utilizando esta configuración, para 24 salidas se necesitarían 144 pines del FPGA.

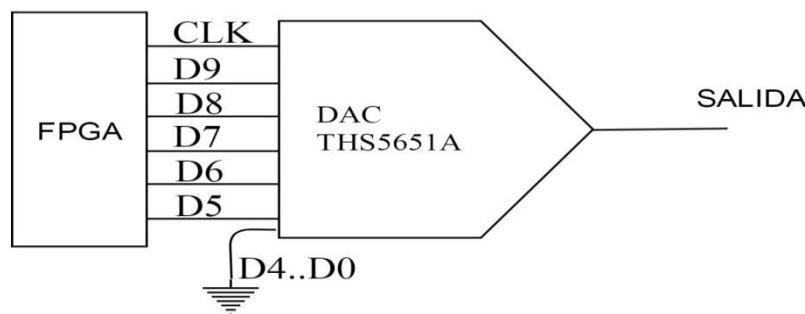


Figura 2.11: Conexión entre el FPGA y el DAC THS5651A.

En la figura 2.11 se muestran los bits usados en el DAC, estos son los más significativos. A continuación en la figura 2.12 se muestra la gráfica de la caracterización del DAC usando los cinco bits más significativos.

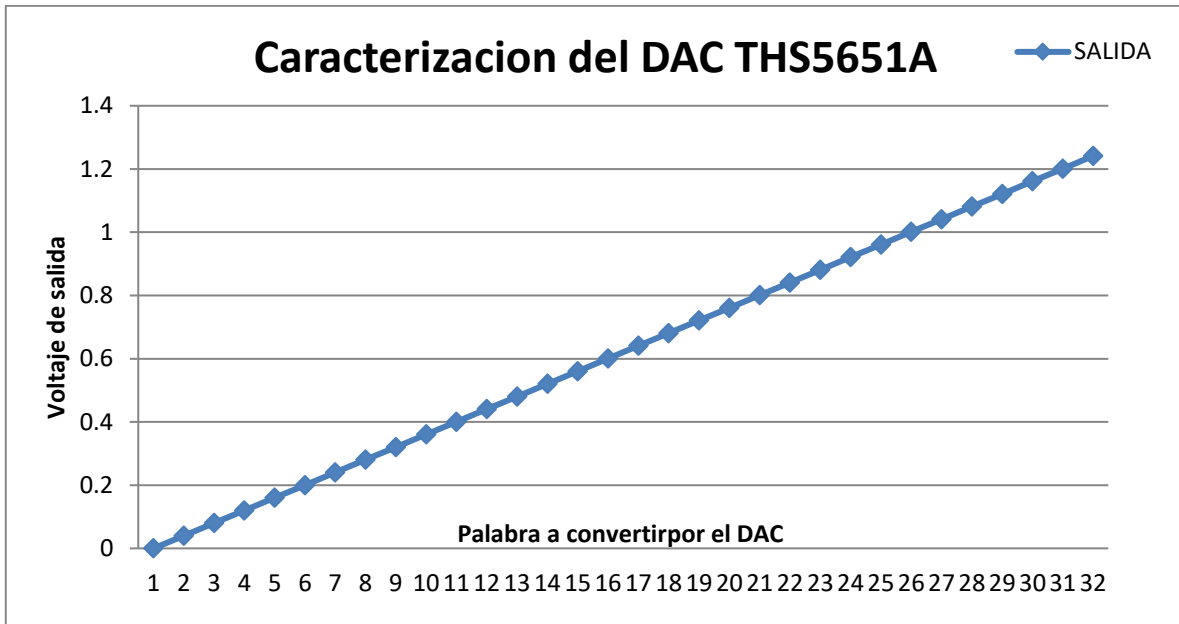


Figura 2.12: Caracterización del DAC THS5651A.

Aunque la resolución del DAC sea configurada a 5 bits se observa que bien podemos alcanzar diferentes niveles de voltaje a su salida. Para acoplar esta salida a los diferentes niveles de voltaje dependiendo de la tecnología a utilizar, se utilizan amplificadores operacionales.

Tabla 2.3: Números en decimal a convertir por el DAC THS5651A.

Numero	Conversión	Numero	Conversión	Numero	Conversión
0	0	13	0.5205	26	1.0410
1	0.0400	14	0.5605	27	1.0810
2	0.0800	15	0.6005	28	1.1210
3	0.1201	16	0.6406	29	1.1611
4	0.1601	17	0.6806	30	1.2011
5	0.2001	18	0.7207	31	1.2412
6	0.2402	19	0.7607		
7	0.2802	20	0.8007		
8	0.3203	21	0.8408		
9	0.3603	22	0.8808		
10	0.4003	23	0.9208		
11	0.4404	24	0.9609		
12	0.4804	25	1.0009		

La tabla 2.3 muestra los valores de salida de voltaje respecto a la palabra digital en la entrada del convertidor D/A. La ecuación de salida del DAC usando los 5 bits más significativos; donde 32 es el valor de queda de la siguiente manera

$$V_{OUT1} = I_{OUT_{FS}} * \frac{PALABRA}{4096} * R_{LOAD} * 32 \quad (2.5)$$

### 2.4.2 Amplificador Operacional

El amplificador operacional de la figura 2.13, que se propuso en la configuración de inversor, es el opa 659, el cual tiene un ancho de banda de 650 MHz y permite un voltaje de suministro de  $\pm 6.5V$ .

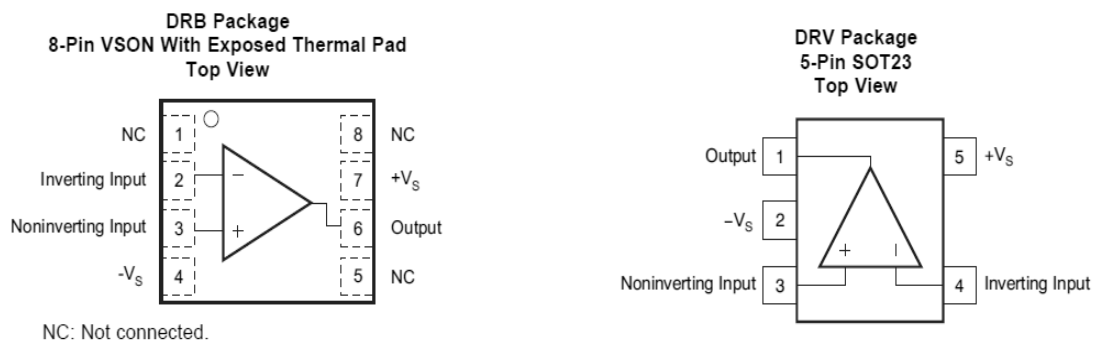


Figura 2.13: Configuración de los pines del OPA659.

El OPA659 combina una banda ancha, ganancia unitaria estable, amplificador operacional de realimentación de voltaje con etapa de entrada JFET para ofrecer un rango dinámico de amplificación ultra alto, para el almacenamiento en búfer de alta impedancia en aplicaciones de adquisición de datos [19].

El ancho de banda del operacional varía respecto a la ganancia configurada en el operacional, como se muestra en la tabla 2.4.

Tabla 2.4: Ganancias propuestas por el fabricante respecto al ancho de banda.

PARÁMETRO	CONDICIONES DE PRUEBA	MIN	TÍPICO	MAX	UNIDAD
ANCHO DE BANDA PARA PEQUEÑA SEÑAL	$V_O = 200mV_{PP}, G = 1 V/V$		650		MHz
	$V_O = 200mV_{PP}, G = 2 V/V$		335		MHz
	$V_O = 200mV_{PP}, G = 5 V/V$		75		MHz
	$V_O = 200mV_{PP}, G = 10 V/V$		35		MHz

Por lo anterior el primer operacional tiene una ganancia de 4.7 y el segundo una ganancia de 2, permitiendo que una frecuencia de 50MHz pueda ser amplificada.

### 2.4.3 Filtrado

A la señal proporcionada por el convertidor digital analógico THS5651A se le suman señales de ruido debido a las interferencias electromagnéticas. Por ello es necesario filtrar la señal mediante un filtro pasa bajas con la intención de reducir las interferencias superpuestas a la señal; generalmente los filtros que trabajan a frecuencias mayores a 1MHz suelen ser pasivos [20].

La función de transferencia de un filtro pasa bajas está dada como la relación tensión de salida, tensión de entrada. La magnitud de la ganancia se presenta en la siguiente ecuación:

$$G(\omega) = \frac{\left(\frac{1}{\omega C}\right)}{\sqrt{R^2 + \left(\frac{1}{\omega C}\right)^2}} = \frac{1}{\sqrt{1 + (\omega RC)^2}} \quad (2.6)$$

Donde  $G(\omega)$  es la magnitud de la ganancia y  $\omega$  es la frecuencia angular. La frecuencia de corte del filtro es aquella a la cual la ganancia del circuito toma el valor  $1/\sqrt{2}$  del valor máximo; cuando la ganancia se ha reducido al 70% y la tensión de salida tiene una amplitud 0.707 veces la de la señal de entrada [20], [21].

La ganancia a la frecuencia de corte es igual a

$$G_c = \frac{1}{\sqrt{2}} \times G_{Vmax} = 0.707 \times G_{Vmax} \quad (2.7)$$

$$G_c = \frac{1}{\sqrt{2}} = \frac{1}{\sqrt{1 + (\omega_c RC)^2}} \quad (2.8)$$

$$2 = 1 + (\omega_c RC)^2 \quad (2.9)$$

$$\omega_c = \frac{1}{RC} \quad (2.10)$$

Por lo tanto, la frecuencia de corte se obtiene por medio de la siguiente ecuación:

$$f_c = \frac{1}{2\pi \times R \times C} \quad (2.11)$$

Dónde:  $f_c$  es la frecuencia de corte,  $R$  es la resistencia y  $C$  es la capacitancia. Debido a que el ancho de banda del generador de patrones digitales es de 0.1Hz-50MHz 0.1 Hz – 50MHz, se decidió diseñar un filtro pasa bajas pasivo RC para filtrar la señal obtenida del convertidor D/A, ver Figura 2.14

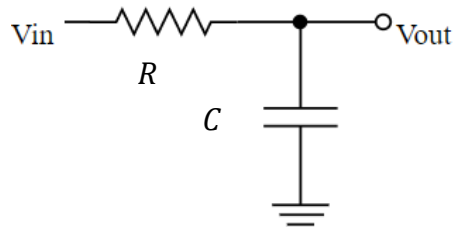


Figura 2.14: Filtro pasa bajas R-C.

El cálculo del filtro pasa bajas R-C, de la figura 2.14, se realiza en función de la frecuencia de corte; por lo que proponiendo una resistencia y conociendo la frecuencia de corte se logra calcular el valor del capacitor. Para este caso específico se propuso una frecuencia de corte de 100 MHz, considerando que el ancho de banda del generador de onda arbitraria es de 0 – 50 MHz, así mismo una resistencia R con un valor de 50Ω, por lo tanto, el valor del capacitor se obtiene con la siguiente ecuación.

$$C = \frac{1}{2\pi * 50 \Omega * 100 \text{ MHz}} = 33 \text{ pF} \quad (2.12)$$

## 2.5 Diseño del circuito impreso PCB

El circuito impreso se define como un modo de conexión de los elementos o componentes electrónicos por medio de pistas de cobre, normalmente adheridas a un soporte aislante rígido o flexible [14]. La placa o tarjeta de circuito impreso (PCB, por sus siglas en inglés) se compone de un soporte físico aislante para la colocación, soldadura de los componentes y el trazado de las pistas conductoras de cobre. El soporte base tiene que ser muy buen aislante eléctrico y muy resistente al fuego. Los materiales más utilizados son: fibra de vidrio, politetrafluoretileno, PTFE-fibra de vidrio, PTFE-fibra de cerámica; siendo una superficie plana de un espesor variable y normalmente de forma rectangular o cuadrada; está constituida por un material base o sustrato de tipo laminado rígido o flexible, termoplásticos, resina epoxidica, resina de silicona, resina melaminica. Los tipos de PCB que actualmente se fabrican son [15]-[17]:

- Monocapa o simple capa.
- Bicapa o doble cara.
- Multicapa o más de dos caras.
- Flexible - multicapa.
- Rígido-flexible multicapa.
- Tridimensional o MCB.

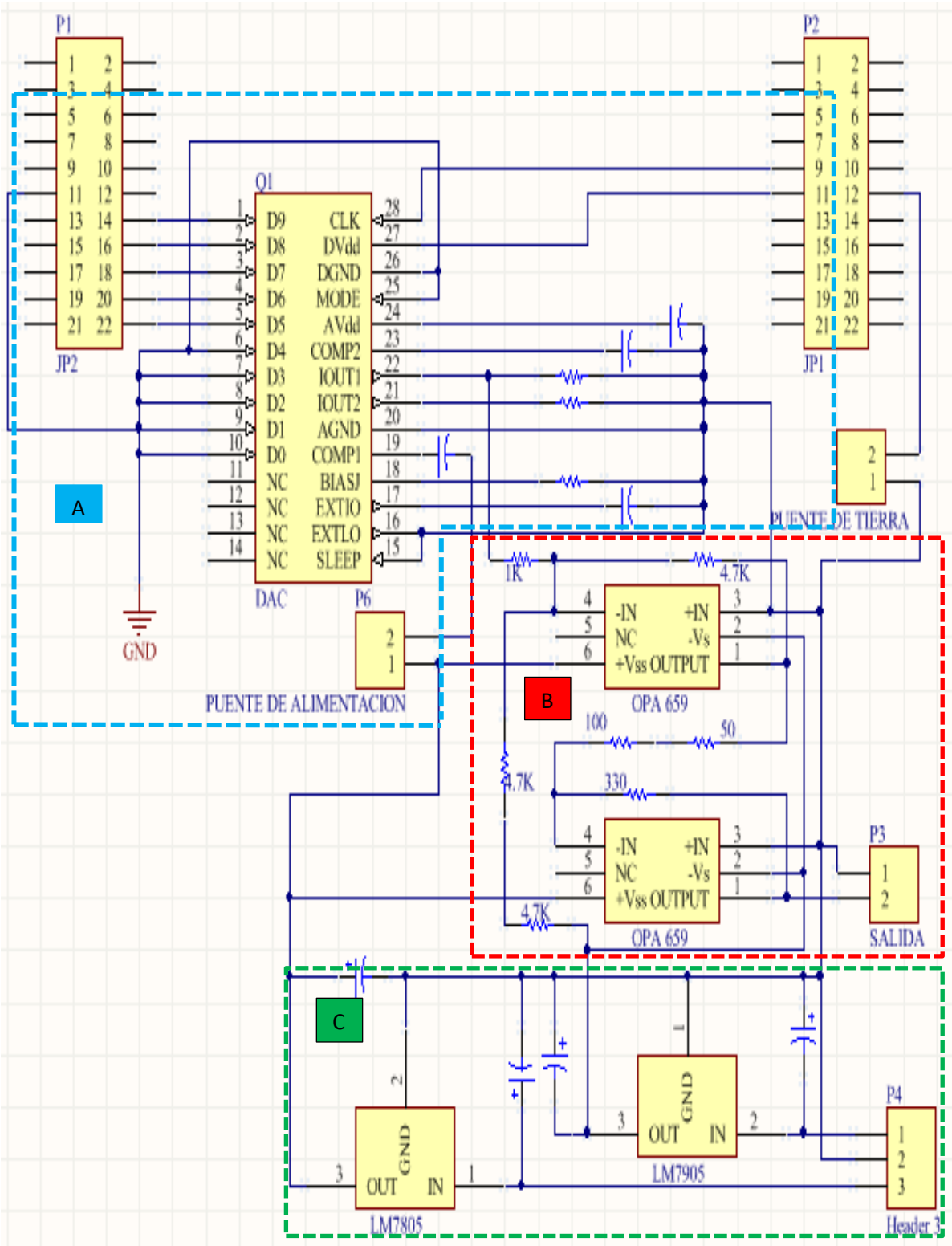


Figura 2.15: Diagrama esquemático del circuito de acoplamiento de señal.

En la sección A de la figura 2.15 se muestran las conexiones del convertidor digital analógico THS5651A; en la sección B se observa la etapa de acoplamiento de señal, realizada con amplificadores operacionales OPA659; en la sección C se muestra la fuente simétrica de  $\pm 5V$  en donde los pines del conector P4 representan las siguientes terminales de conexión:

- 1 : -12V
- 2 : GND
- 3 : +12V

Para el diseño del circuito impreso es importante considerar el ruido eléctrico, el cual afecta el funcionamiento de la electrónica del circuito impreso [15] -[17]. Los diferentes tipos de ruido se mencionan a continuación.

### 2.5.1 Tipos de ruido

- Ruido externo: Se produce debido a interruptores externos al circuito: arranque de motores, debido al arco que se forma al abrir el relé.
- Ruido en las líneas de alimentación de los circuitos integrados.
- Cross-talk: Ruido inducido en una línea por la señal de una línea cercana [15] [17].
- Ruido debido a la propia corriente de la señal a causa de impedancias por las que pasa esa corriente, malas soldaduras [26].

Con el objetivo de eliminar las perturbaciones ajenas al sistema es necesario utilizar diferentes estrategias, tales como que el chasis del equipo sea conductor y vaya conectado a tierra [29]. Es indispensable que la tierra del circuito impreso vaya conectada a una placa específicamente destinada a tierra. Si es necesario, es de utilidad dividir los planos de tierra en tierra digital y analógica. Una parte sensible de un sistema digital es aquel actuado por un flanco de subida o de bajada, por lo que es recomendable proteger estos elementos. En el caso del diseño del circuito, la señal de reloj del convertidor tiene que ir protegida, esto se logra poniendo postes a tierra alrededor de la pista.

Otra estrategia para reducir las interferencias electromagnéticas (EMI, por sus siglas en inglés) es a partir de filtros EMI, que consiste en un arreglo de capacitores distribuidos por toda la superficie de la placa, en puntos estratégicos, de tal forma que el rizo se va a tierra y es absorbido por los filtros, ver Figura 2.16.

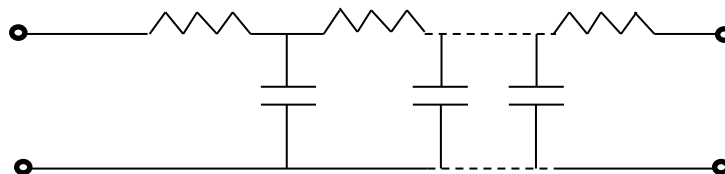


Figura 2.16: Filtro EMI.

Los filtros EMI, como el de la figura 2.16, reducen el ruido proveniente de las fuentes de alimentación del circuito.

## 2.6 Descripción del hardware

A continuación se describen las conexiones de las tarjetas que forman el generador de patrones digitales.

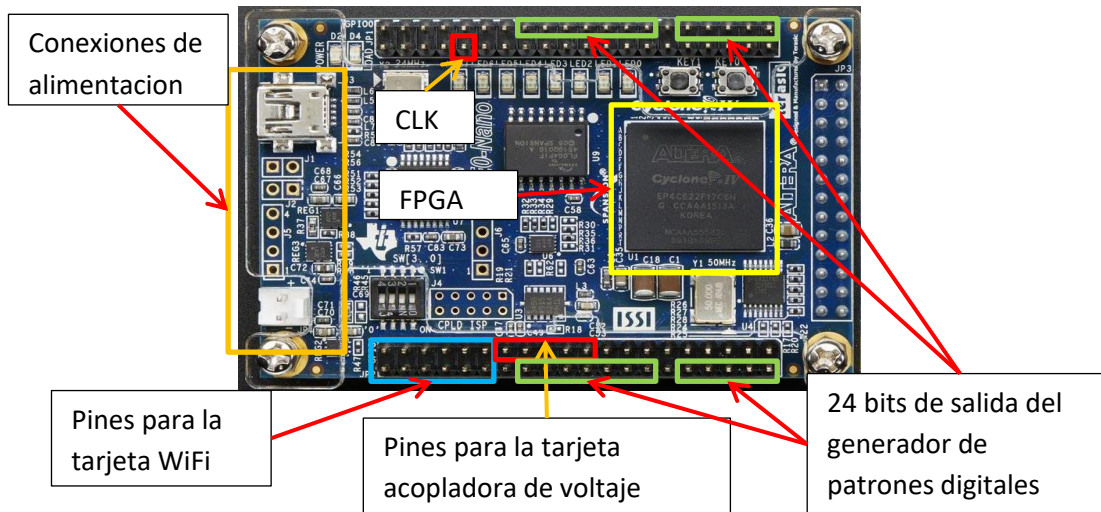


Figura 2.17: Conexiones del Hardware.

En la figura 2.17 se encierran y se describen las conexiones y los pines a usar de la tarjeta DEO\_NANO, para la tarjeta acopladora de voltaje y los 24 bit de salida de tecnología CMOS, también se muestra el encapsulado del FPGA CYCLONE IV.

Para identificar cada una de las conexiones se utilizan diferentes colores: el marco de color naranja encierra las dos posibles formas de alimentar la tarjeta, ya sea por un puerto mini USB tipo B o un conector molex de 2 pines; el marco de color azul encierra los pines usados por el módulo WiFi; los marcos de color rojo encierra los pines que se utilizan para la tarjeta acopladora de voltaje; los marcos de color verde encierran los 24 pines de salida del generador de patrones digitales, los cuales tienen un nivel fijo de tecnología CMOS; por último en marco de color amarillo se encierra el FPGA CYCLONE IV en el cual se graba el firmware del generador de patrones digitales.

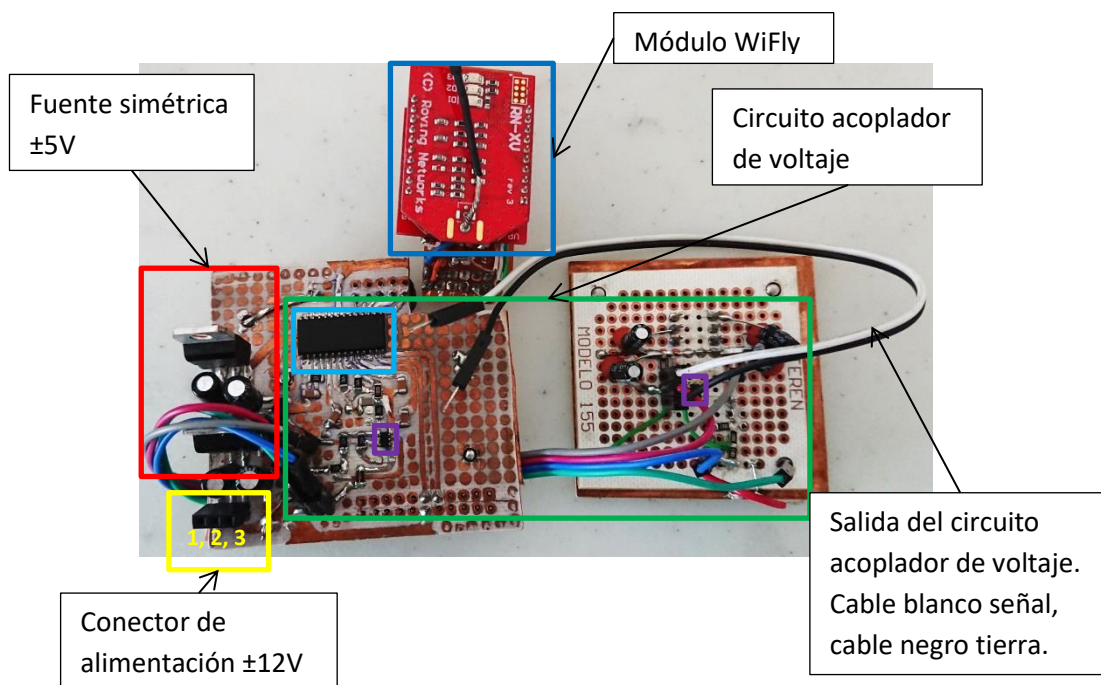


Figura 2.18: Descripción de las conexiones de la tarjeta acopladora de voltaje.

De igual forma en la figura 2.18 se muestran las etapas de la tarjeta acopladora de voltaje, para indicar cada etapa de la tarjeta, cada una de ellas es encerrada en un marco de diferente color:

en el marco de color amarillo se muestra el conector de la fuente de alimentación de  $\pm 12V$  para la fuente simétrica de  $\pm 5V$  encerrada en el marco de color rojo; en el marco de color azul se muestra el módulo WiFi; usado en la comunicación del FPGA con una pc; dentro del marco verde, se muestran los componentes del circuito acoplador de voltaje; con un marco de color azul se señala el convertidor digital analógico y finalmente con color morado el amplificador operacional.

## 2.7 Conclusiones

El hardware es fundamental para el desarrollo del generador de patrones digitales, pues de él depende la posibilidad de reproducir el patrón digital a 50 MHz o más; también de este depende que el sistema pueda reproducir los diferentes niveles de voltaje de las tecnologías digitales. Siendo de suma importancia la selección de los dispositivos electrónicos a usar, así como el diseño de la conexión entre ellos.

## CAPITULO: 3 Firmware

El firmware del FPGA de manera general se menciona en la figura 3.1, se encarga de convertir los datos, en serie enviados por la tarjeta WiFi a datos en paralelo antes de crear el bus de 40 bits se discrimina si la información recibida a través de la tarjeta RN-XV es válida para el firmware del FPGA aquí es donde entra en juego el identificador de comando, con el cual se habilita la creación del bus de 40 bits para su posterior manipulación. Debido a que el primer byte enviado es el comando, al ser un comando válido se crea el bus de 40 bits; donde el comando ocupa el byte más significativo. Si el comando recibido es 241 se habilita la escritura de la memoria FIFO, con el cual se debe de escribir el dato enviado junto con el comando en la memoria FIFO y verificar que no se repita a menos que el siguiente dato recibido sea el mismo. Si el comando es 245 se configura la frecuencia del reloj del módulo que controla la lectura de la memoria FIFO, es decir, solo se recibe el número que representa la frecuencia deseada y no se ejecuta hasta que se reciba el comando 244, el cual ordena la lectura de la memoria FIFO.

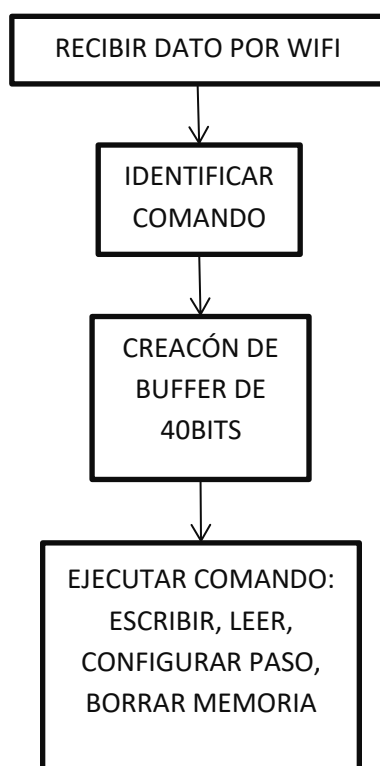


Figura 3.1: Descripción breve del funcionamiento del firmware del generador de patrones digitales.

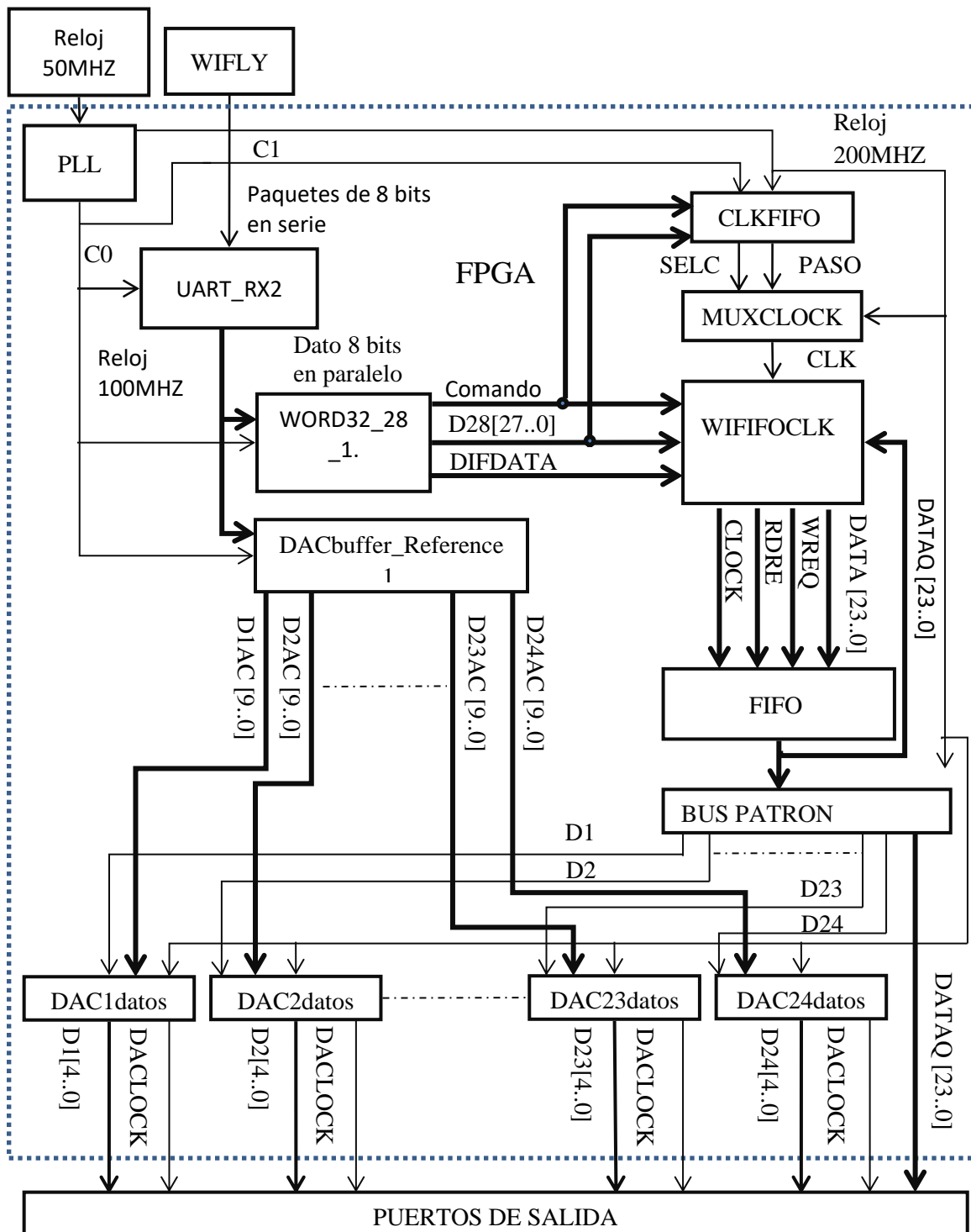


Figura 3.2: Diagrama de firmware del generador de patrones digitales.

En la Figura 3.2 se muestran los bloques del firmware del generador de patrones digitales. Cada uno de los bloques realiza una función específica, a continuación se describe el funcionamiento de cada uno de estos bloques.

### 3.1 Multiplicador de frecuencia PLL

Como primer bloque se menciona el multiplicador de frecuencias, el cual es una mega-función diseñada por Altera. En este bloque se crean las frecuencias de reloj necesarias para el generador de patrones digitales.

La figura 3.3 es un PLL del cual se obtienen dos frecuencias de reloj; una de 100 MHz y otra de 200MHz; con base a la frecuencia del reloj de la tarjeta que es de 50 MHz.

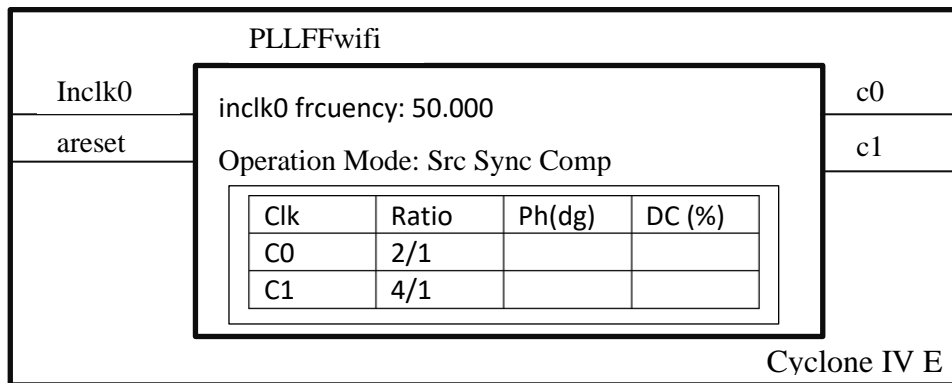


Figura 3.3: Multiplicador de frecuencia.

La salida C0 es una frecuencia usada como señal de reloj para recibir los datos del módulo WiFi, posteriormente armar el bus de 40 bits y mostrar los datos según el comando recibido. C1 es la señal de reloj del bloque CLKFIFO, el cual modifica el paso de reproducción de la memoria FIFO.

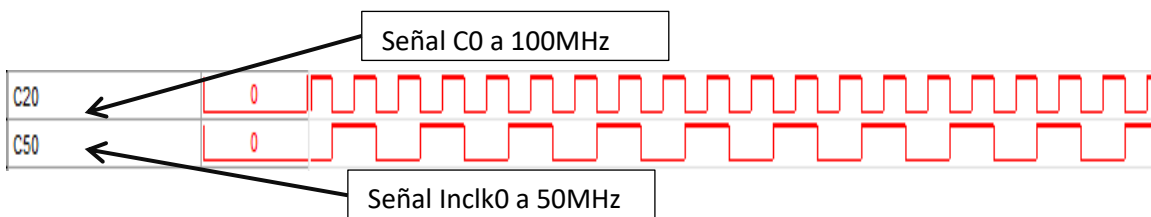


Figura 3.4: Señales de reloj a la entrada y salida del módulo PLL.

En la figura 3.4 se muestran las diferentes señales de reloj usadas en el firmware del generador de patrones digitales; la señal “clock” es la señal de entrada al módulo PLL, esta señal es provista por un cristal de cuarzo de 50 MHz. Como C1 es la señal de 200 MHz se utilizó como señal de reloj para el analizador lógico, es por ello que no se visualiza

### 3.2 De serial a paralelo bloque UART\_RX2

El Transmisor Recibidor Asíncrono Universal (UART, por sus siglas en inglés), permite establecer una comunicación serial entre dos dispositivos. Ambos dispositivos deben ser configurados para que funcionen a un flujo de datos predeterminado, a esta tasa de transferencia se le conoce como baudrate o tasa de bits por segundo [22] [24]. La comunicación básica se muestra en la figura 3.5.

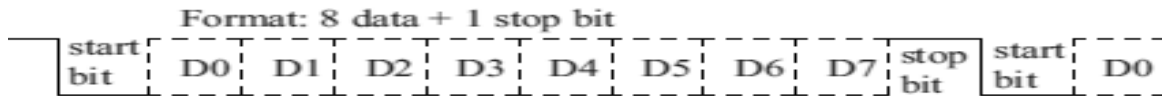


Figura 3.5: Protocolo de comunicación UART-232.

Cuando la transmisión cambia de “1” a “0” y a este bit se le conoce como bit de inicio. Los bits de información son transferidos del menos significativo al más significativo usualmente son 8 bits y por último el bit de finalizado.

La velocidad de transferencia está dada por:

$$\text{bits por segundo (bps)} = \frac{1}{\text{tiempo por bit}} \quad (3.1)$$

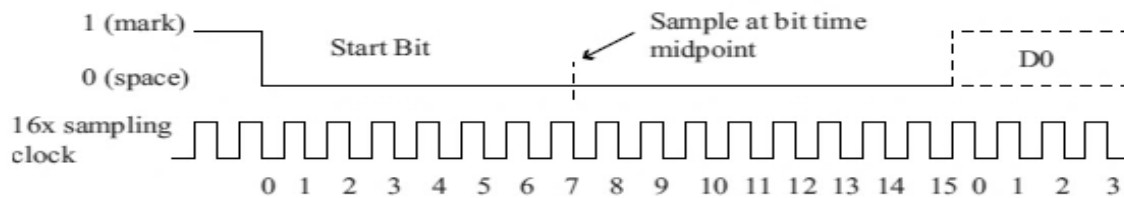


Figura 3.6: Tiempo de muestreo del dato recibido por WI-FI.

La figura 3.6 muestra la frecuencia de muestreo del pin RX con el cual se evitan lecturas erróneas [23],[25].

El total de bits por segundo es el inverso del tiempo que se toma en transferir un bit. Para garantizar una transferencia adecuada, el tiempo de muestreo debe ser 16 o 64 veces mayor que el de transferencia; por si existiera algún desfase [23].

El siguiente bloque UART\_RX2, recibe los datos del módulo RN-XV para convertirlos de serie a paralelo, para poder discernir si los datos recibidos son válidos o no, para su posterior concatenación en un bus de 40 bits.



se cumple se pasa al estado S1 donde se lleva un conteo hasta 10 y se verifica que la entrada sea GND, si esto se cumple pasa al estado S2 se realiza otro conteo hasta 98 y pasa al estado S3, se realiza otro contador a 215 y pasa al estado S4 donde según la bandera FLAGB pasará del estado S5 al estado, S13 donde se van capturando los 8 bits del byte enviado por la interfaz de usuario y entre cada bit recibido pasa por S3 y cuando llega a S13 pasa al estado S14, donde se lleva otro conteo hasta 108 y de ahí regresa al estado inicial S0.

Cada vez que se reciben 8 bits y se verifica que el nivel de UART\_RX es alto (VCC), entonces se procede a mostrar el arreglo de 8 bits en paralelo para el siguiente bloque y se activa la bandera dataRdy. De esta manera confirmamos que los 8 bits recibidos son válidos para enviar al siguiente bloque. Este proceso se muestra en la figura 3.9.

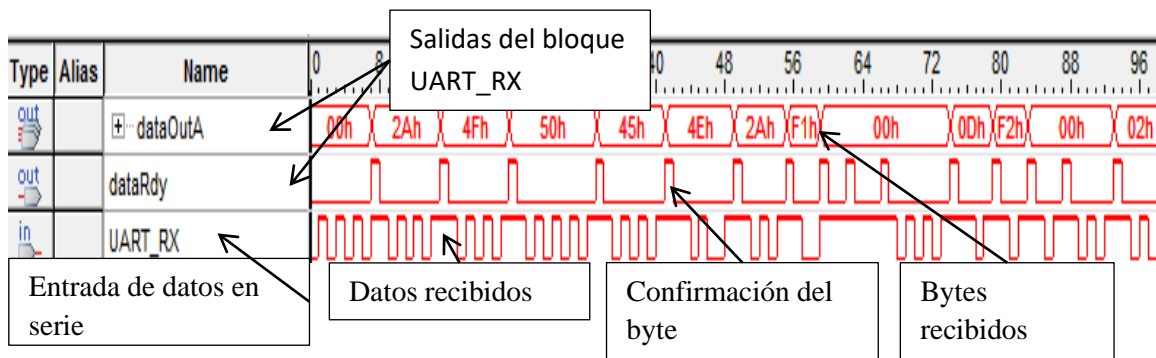


Figura 3.9: Ejemplo de los valores convertidos de serie a paralelo por el módulo UART\_RX2 en su salida dataOutA.

### 3.4 Identificación de comandos

Una vez que se han convertido los datos de serie a paralelo, es necesario determinar si se trata de un comando válido para después realizar la creación del bus de 40 bits, de esto se encarga el bloque word32\_28\_1, figura 3.10.

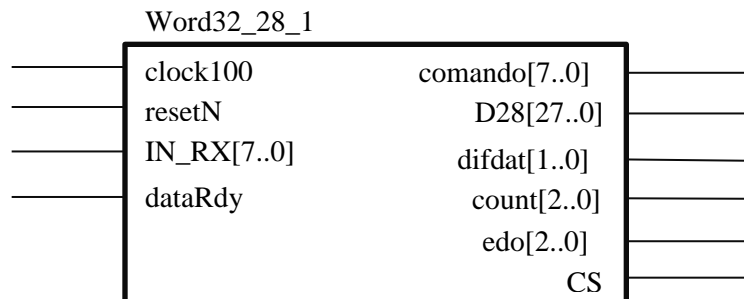


Figura 3.10: Módulo donde se unen los 5 bytes.

La interfaz de usuario envía 5 bytes (40 bits) para que el firmware del generador de patrones digitales los interprete según la programación. Por lo tanto el módulo de identificación de comandos recibe cinco paquetes de 8 bits cada uno, siendo el comando el primer paquete. Solamente cuando el primer byte recibido es un comando válido, entonces pasa a recibir los siguientes 4 bytes. En la Figura 3.11 se menciona la secuencia de los bytes recibidos.

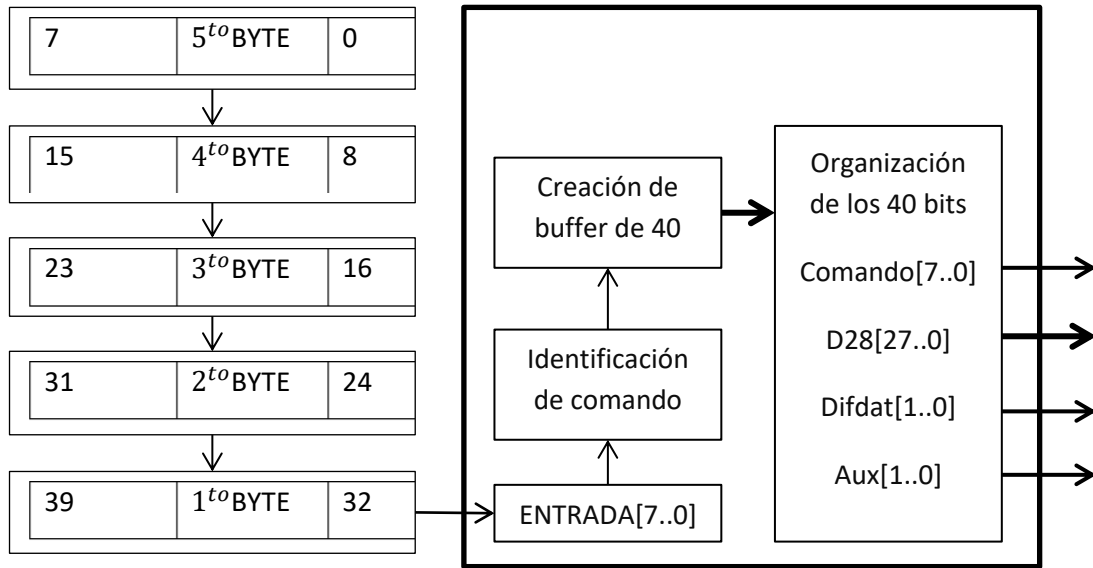


Figura 3.11: Diagrama a bloques del módulo identificación de comandos.

En la izquierda de la figura 3.12 se muestran los 5 bytes recibidos por el módulo "UART\_RX", los cuales son enviados al módulo "identificador de comandos" donde primero se verificará que el primer byte recibido sea un comando para proceder a juntar los 5 bytes y realizar la acción del comando recibido.

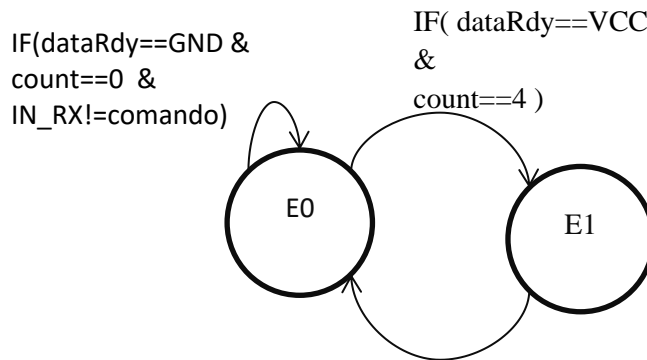


Figura 3.12: Diagrama de estados del módulo word32\_28\_1.

En el estado E0 de la figura 3.12, primero se identifica que el primer byte recibido sea un comando válido, para posteriormente anexar los siguientes cuatro bytes con el byte de comando en una palabra de 40 bits. En el estado E1, estos 40 bits son organizados según el comando recibido y mostrados en las siguientes salidas, “comando. D28[27..0], difdata[1..0]”.

Tabla 3.1: Tabla de comandos.

COMANDO	DESCRIPCIÓN
241,F1	Habilita la escritura de los datos recibidos en la memoria FIFO
242,F2	Borra la memoria FIFO
243,F3	Indica que se terminaron de enviar los datos
244,F4	Habilita la reproducción de la memoria FIFO
245,F5	Habilita la configuración de la frecuencia de reproducción.

La tabla 3.1 hace mención de los comandos usados en el firmware del generador de patrones digitales y de las acciones que indican cada comando en el firmware del generador de patrones.

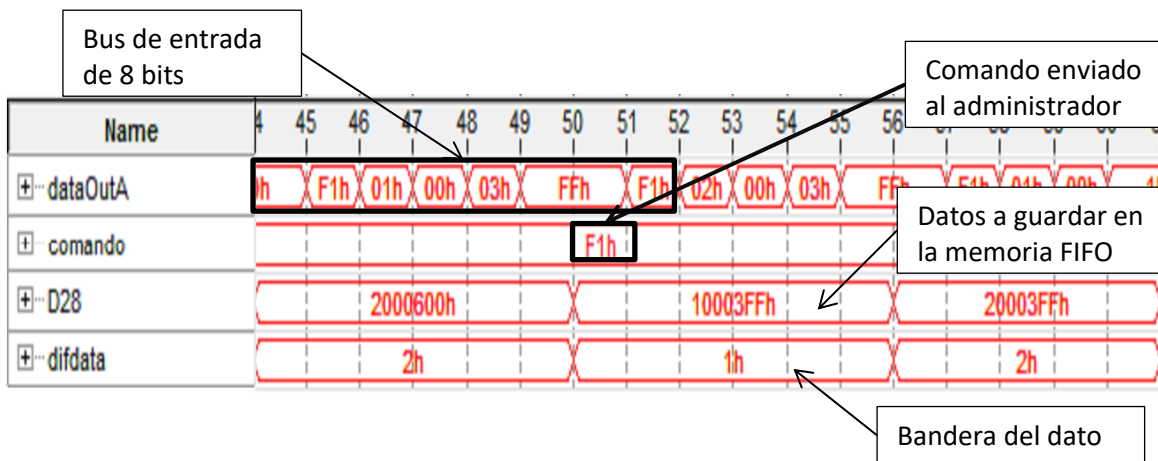


Figura 3.13: Ejemplo de la identificación de comandos.

La figura 3.13 muestra el diagrama de tiempo de las entradas y salidas del bloque identificador de comandos el cual es visualizado en el analizador lógico de altera. Se muestra el comando “F1 (241)” el cual indica que los datos en D28 deben guardarse en la memoria FIFO. Para no repetir el dato en la memoria FIFO se tienen dos bits que indican que el dato presente debe guardarse en la memoria, los cuales están presentes en el bus difdata y solamente cuando cambia entre 1 y 2 es cuando el dato de D28 debe guardarse en la memoria FIFO; de esta manera se evita repetir el dato en la memoria FIFO, pues sin esta condición, en lo que cambia el dato de D28 la memoria FIFO ya guardo más de una vez este dato y no se respetaría la secuencia del patrón digital ingresado por el usuario.

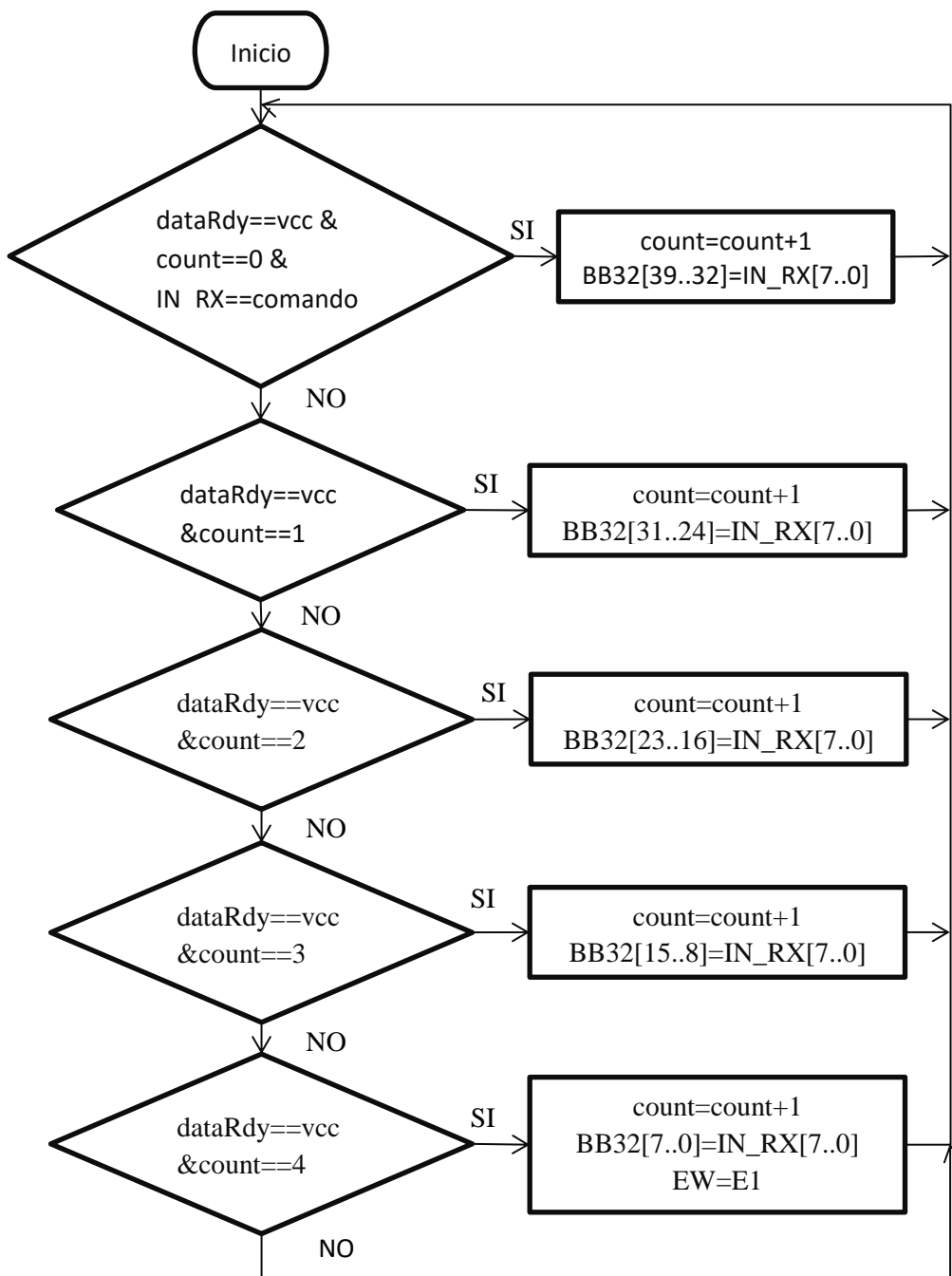


Figura 3.14: Diagrama de flujo del estado E0.

En el estado E0; explicado por el diagrama de flujo la figura 3.14 se crea el bus de 40 bits. Después de esto se pasa al estado E1 donde, según el comando, la información es organizada y enviada al siguiente bloque.

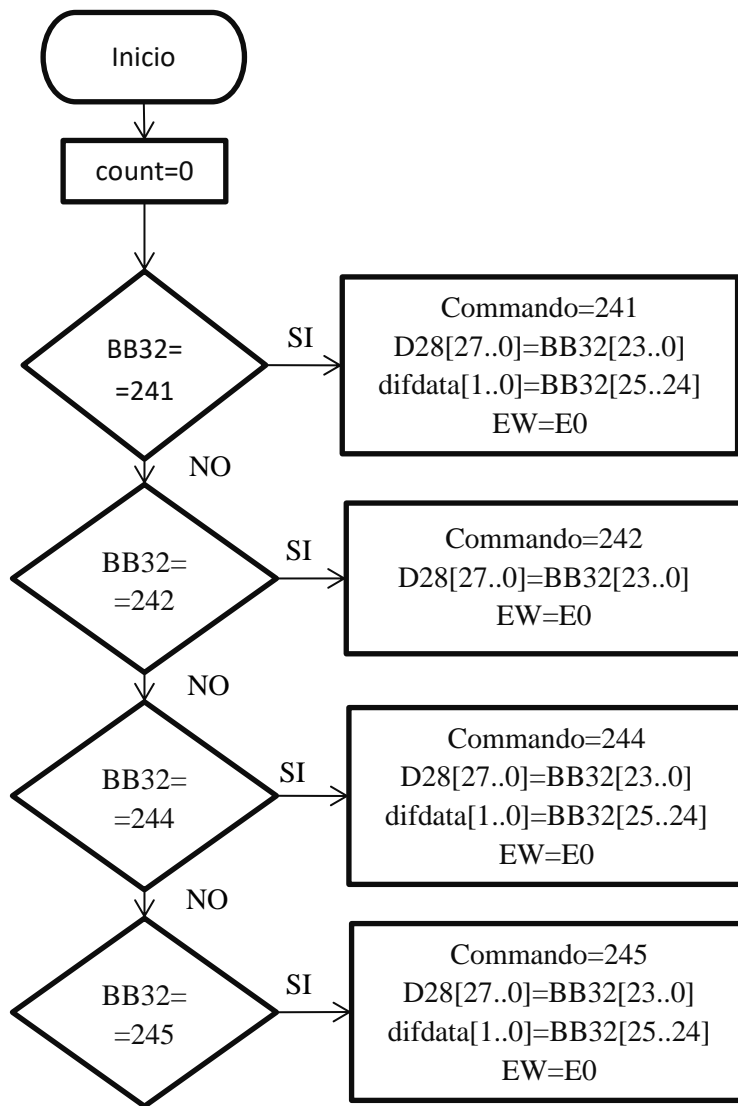


Figura 3.15: Diagrama de flujo del estado E1.

Una vez creado el bus de 40 bits el firmware del bloque identificador de comandos pasa al siguiente estado E1, donde según el comando recibido el bus de 40 bis es dividido como se muestra en el diagrama de flujo de la figura 3.15 y regresa al estado E0.

### 3.5 Cambio de paso

Para poder variar el paso de reproducción de la memoria, se propone el bloque CLKFIFO; funciona como divisor de una frecuencia principal (200 MHz), esta es dividida por un contador interno el cual es configurado desde la interfaz del usuario para que la memoria trabaje a la frecuencia indicada por el usuario. Junto con el comando 245, se envía el número de cuentas del reloj principal de este

módulo y así se determina la frecuencia de reproducción del patrón digital almacenado en la memoria FIFO.

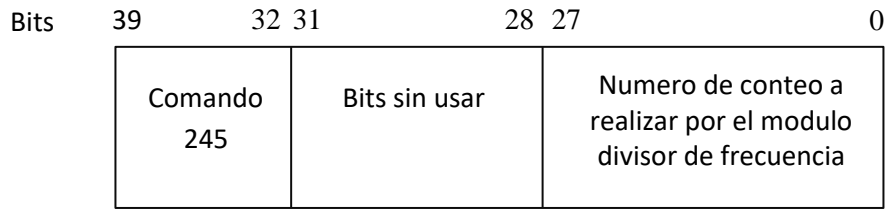


Figura 3.16: Comando 245: configuración del reloj de la memoria FIFO.

En la figura 3.16 se menciona la estructura del bus de 40 bits para el comando 245, donde el primer byte muestra el comando, los siguientes 4 bits no se usan y los restantes 28 bits representan el número a contar para variar el paso de la reproducción del patrón digital. La figura 3.17 describe las conexiones del bloque CLOCKPASOFIFO.

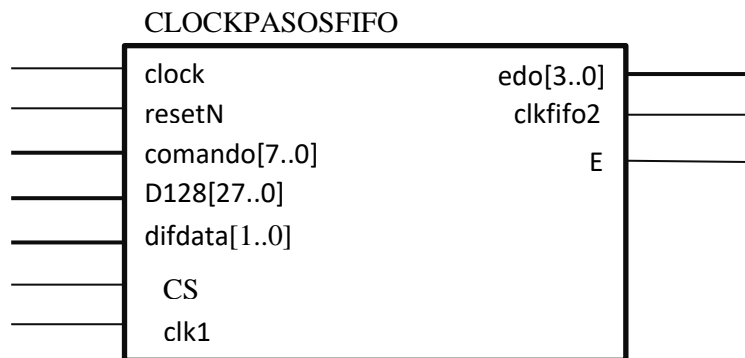


Figura 3.17: Módulo divisor de frecuencia.

Debido a que el módulo UART\_RX, el cual recibe los datos seriales de la tarjeta RN-XV; trabaja a una frecuencia de 100MHz y el módulo clkfifowifi2 que controla la memoria FIFO, tiene que recibir y escribir los datos a esta frecuencia, por lo tanto el módulo CLOCKPASOSFIFO debe dejar pasar la frecuencia de reloj de 100 MHz, razón por la cual este módulo trabaja al doble de esta frecuencia, de ahí el uso de la segunda frecuencia del PLL a 200 MHz. Una vez que se ha escrito la memoria con los datos del patrón digital, se procede a configurar la frecuencia de la salida del módulo CLOCKPASOSFIFO y con ello la frecuencia de trabajo del módulo clkfifowifi2, posteriormente al recibir el comando de reproducción de la memoria FIFO el módulo CLOCKPASOSFIFO procede a dividir la frecuencia de su reloj a como fue configurado.

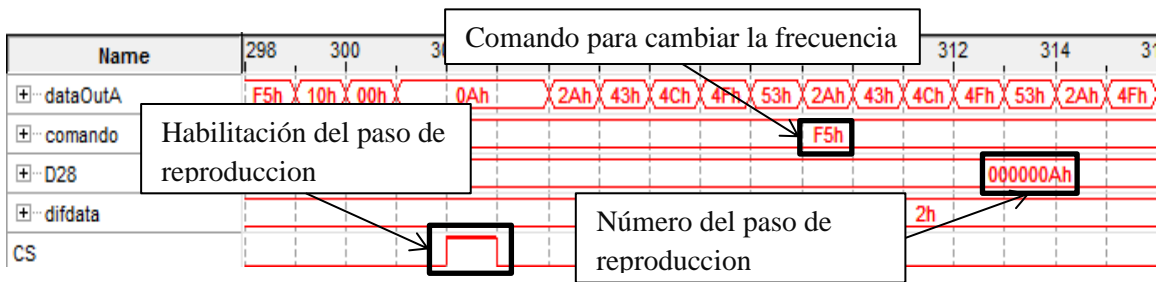


Figura 3.18: Comando para cambiar el paso de reproducción de la memoria FIFO.

La figura 3.18 muestra el comando 245 (F5 hex) con el cual se configura el reloj de la memoria FIFO, para asignar el número de conteo para la señal de reloj, se utiliza la señal "CS", la cual indica en qué momento el bloque divisor de frecuencia debe de capturar el número a contar.

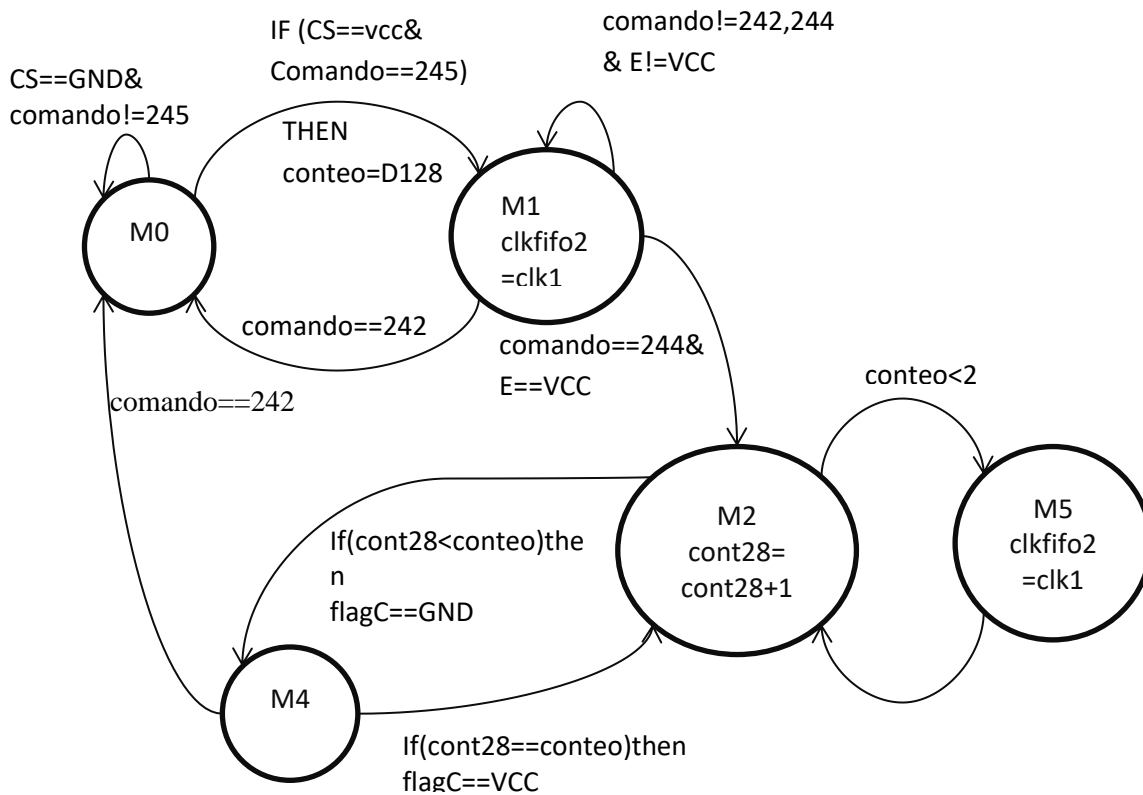


Figura 3.19: Diagrama de máquina de estados CLKPASOSFIFO.

En la máquina de estados de la figura 3.19 se muestra que la condición para capturar el número a contar es que la entrada “CS” esté en un nivel alto y pasa al siguiente estado M1 donde espera el comando F4 para iniciar el paso de reproducción asignado por el usuario, pero mientras deja pasar el reloj de 100 MHz para que se guarden los datos del patrón digital. Cuando recibe el comando F4 inicia el conteo para el reloj del bloque administrador de la memoria FIFO. Si el paso asignado es de 10 ns, entonces pasa al estado M5, pero si el paso asignado es de 20 ns o mayor entonces la señal de reloj se crea entre los estados M2 y M4 y permanece en esta secuencia hasta que se reciba el comando F2 y entonces regresa al estado M0.

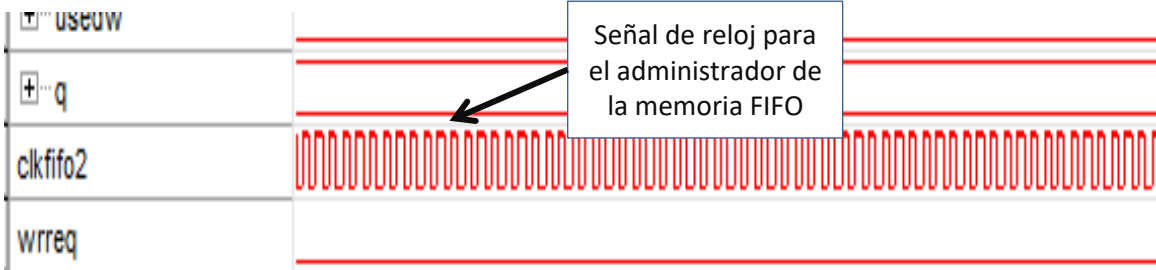


Figura 3.20: Salida del bloque CLOCKPASOSFIFO, sin configurar el divisor de frecuencia 100MHz.

Las figura 3.20 muestra la señal de reloj para el módulo administrador de la memoria FIFO con el paso configurado a 10ns.

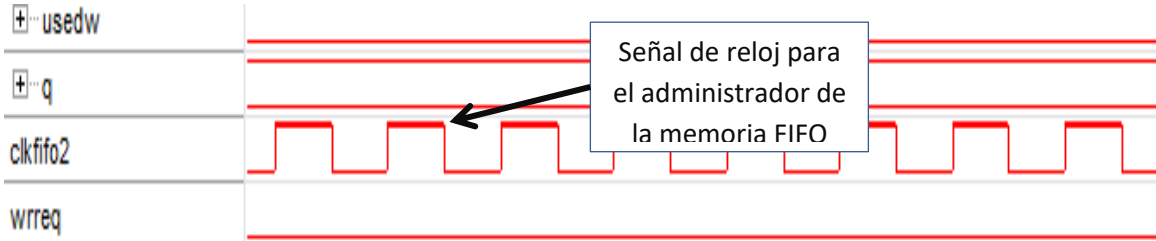


Figura 3.21: Salida del bloque CLOCKPASOSFIFO configurado a una frecuencia de 50 MHz.

En la figura 3.21 se observa la señal de reloj del módulo administrador de la memoria FIFO para un paso de 20ns y así los datos de la memoria FIFO se reproduzcan a 50 MHz.

### 3.6 Administrador de memoria

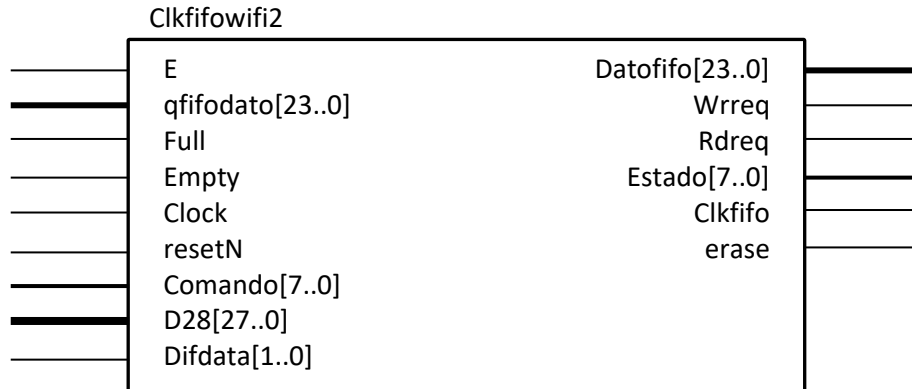


Figura 3.22: Módulo clkfifowifi2.

El bloque Clkfifowifi2, ver figura 3.22, se encarga de administrar la memoria FIFO, es decir si el comando recibido indica que los datos recibidos deben de ser guardados en la memoria, genera las condiciones necesarias para escribir el dato en la memoria FIFO; ahora bien si el comando recibido indica la lectura de la memoria, se generan las condiciones necesarias para leer la memoria y volver a escribir el dato leído para mantener la información en la memoria y pueda ser leída nuevamente sin ningún problema.

El diagrama a bloques de la figura 3.23 explica el funcionamiento de este bloque.

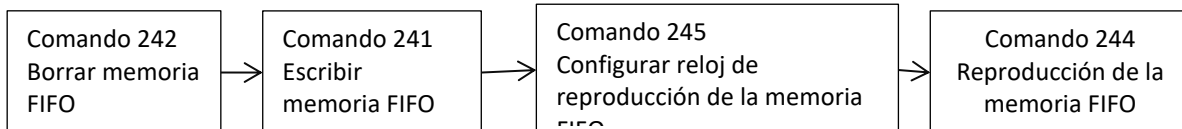


Figura 3.23: Secuencia de comandos para la escritura y lectura de la memoria FIFO.

Debido a que la frecuencia de 100MHz es mayor a la frecuencia de envío del puerto WIFI, es necesario indicar en qué momento se deben de guardar los datos y así no repetir el dato a menos que se indique que se tiene que guardar el mismo dato. Para ello se utiliza la entrada difdata[1..0]; esta entrada es utilizada para indicar que el dato presente en la entrada del módulo, es un dato que debe guardarse en la entrada de la memoria FIFO. Sin esta condición el dato presente podría llenar la memoria FIFO antes de que se reciba el siguiente dato.

Los dos bits de la bandera “difdata” pasan por una función XOR y solamente cuando el resultado es verdadero, procede a guardar el dato presente en “qfifodato[23..0]”. Esto se cumple solamente cuando los bits del valor anterior de difdata, comparados con los bits del actual, son diferentes. Con este procedimiento se evita la repetición de los datos recibidos.

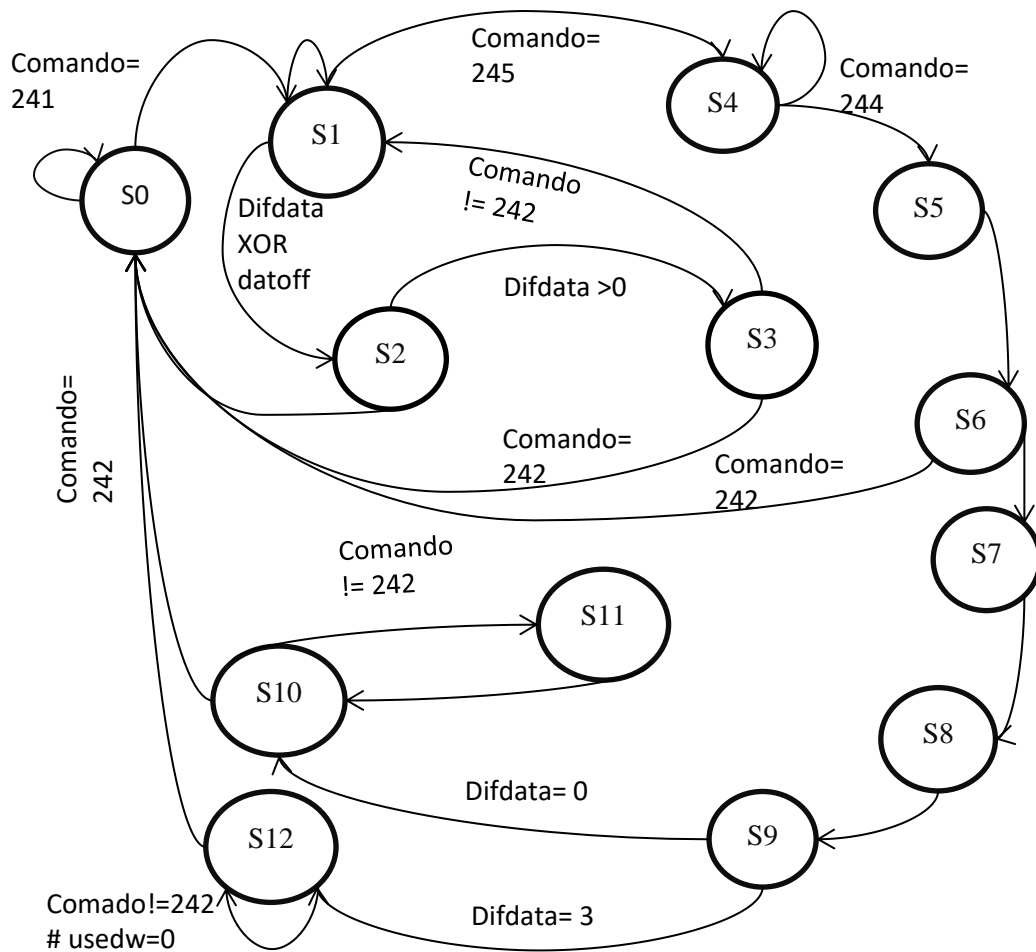


Figura 3.24: Máquina de estado del bloque clkfifowifi2.

La figura 3.24 muestra la máquina de estados del bloque administrador de la memoria FIFO llamado Clkfifowifi2, el cual se encarga de escribir en la memoria los datos del patrón digital recibidos por el puerto WiFi, que están acompañados con el comando 241 (F1 hex); de esto se encargan los estados S1, S2, S3. Cuando se recibe el comando 245 (F5 hex) con la configuración del conteo el bloque pasa al estado S4 para esperar el comando 244 (F4 hex) y pasar a la siguiente secuencia de estados desde S5 hasta S10 y S11, donde estos dos últimos se encargan de generar la señal de reloj de la memoria FIFO y cumplir con el paso asignado por el usuario. Difdata indica el tipo de reproducción, para “cíclico” el valor es 0 y para “una vez” el valor es 3.

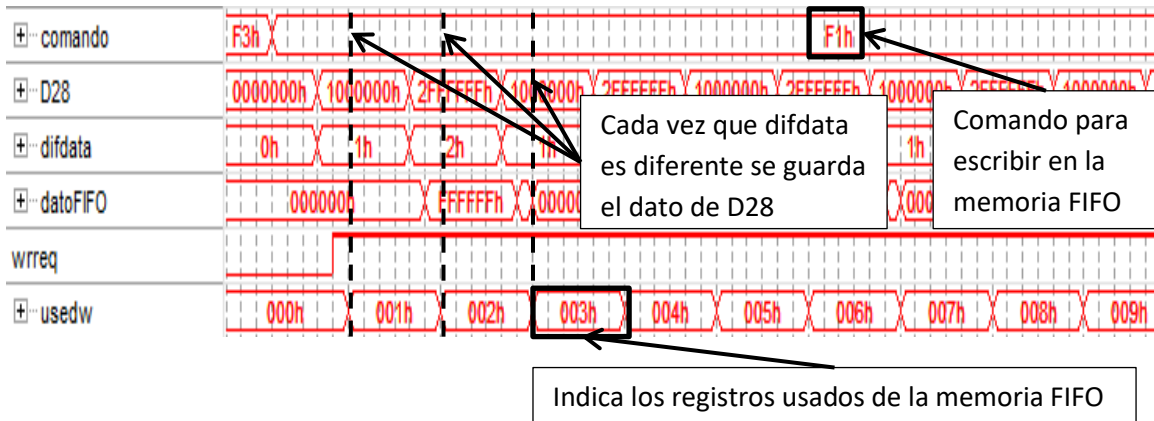


Figura 3.25: Ejemplos de datos a guardar junto con la condición "difdata".

En la figura 3.25 explica la condición de escritura del dato recibido por el bloque UART\_RX, la cual consiste en la función XOR, aplicada al dato presente en el bus "difdata" y el dato anterior del mismo. Solo cuando son diferentes entonces se procede a guardar el dato en la memoria FIFO.

### 3.7 Memoria FIFO

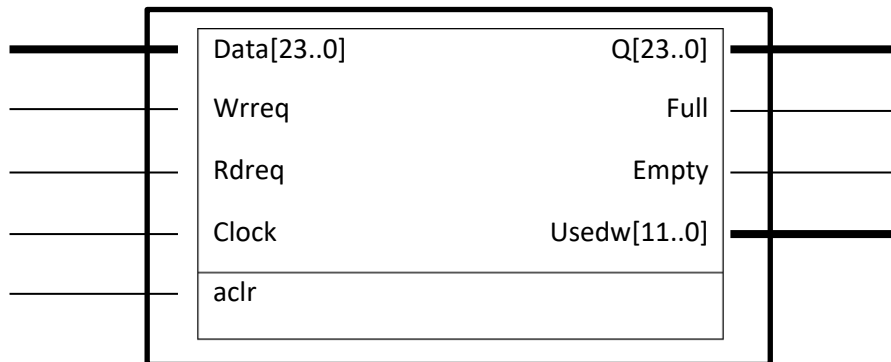


Figura 3.26: Módulo de la memoria FIFO.

El bloque FIFOwifi, figura 3.26, es la memoria FIFO de 8K de 24 bits de palabra donde se guardan los datos del patrón digital, a esta memoria se le habilito la entrada CLR asíncrona, en cualquier momento en el que tenga un nivel alto, todo el contenido de la memoria es borrado y el contador de registros usados pasa a cero. A continuación se muestra una imagen que describe el comportamiento de la memoria FIFO [7].

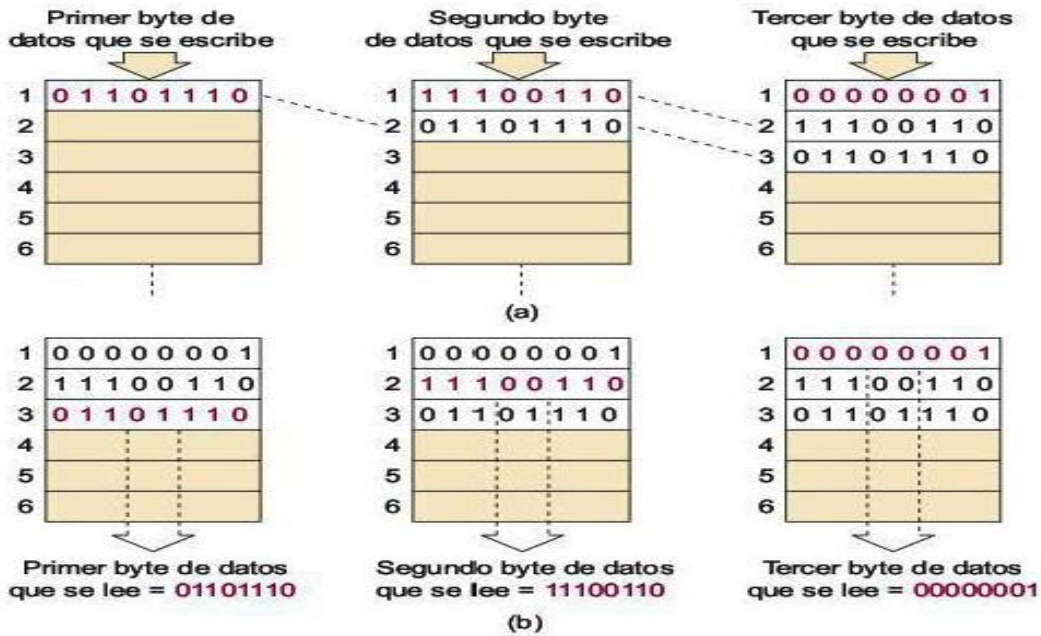


Figura 3.27: En FIFO, los valores de los datos se leen de la memoria (b) en el mismo orden en el que se escribieron en la memoria(a).

La figura 3.27 describe el funcionamiento de escritura y lectura de la memoria FIFO

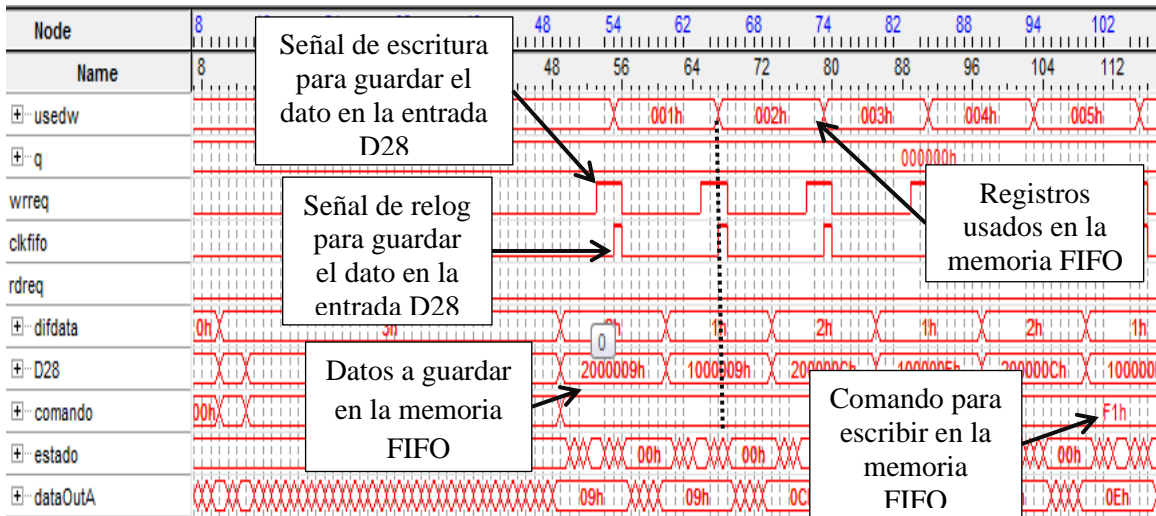


Figura 3.28: Se muestra la habilitación de la escritura de la memoria y el pulso de reloj de la memoria FIFO para guardar el dato presente en la entrada de la memoria FIFO.

En la figura 3.28 se muestra el procedimiento para escribir los datos en la memoria FIFO, primero se habilita la señal de escritura `wrreq`, el dato se guarda en el flanco ascendente de la señal de reloj de la memoria FIFO.

La figura 3.29 muestra la lectura y escritura de la memoria FIFO, pues si la memoria no fuera escrita mientras se lee, ya no podría ser leída de nuevo. Si su apuntador llegara a la posición cero, indicaría que la memoria se encuentra vacía, haciendo imposible la lectura de los datos. Por esta razón es necesario escribir la memoria conforme se lee, esto se logra habilitando la lectura de la memoria, se pierde el primer ciclo de reloj para posteriormente habilitar la escritura y finalmente se dejan habilitados la lectura y escritura de la memoria FIFO mientras se reproduce el patrón digital.

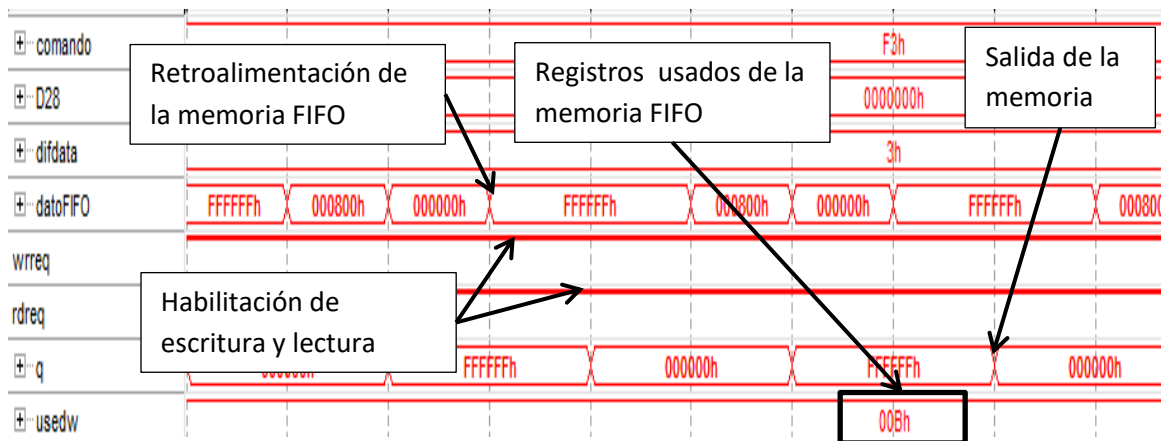


Figura 3.29: Lectura y escritura de la memoria FIFO para mantener los datos en la memoria.

Si solo se leyera la memoria FIFO esta pasaría al estado de memoria vacía y para leerla de nuevo habría que escribir otra vez los datos. Esta manera de trabajar no es eficiente para el generador de patrones digitales, pues no sería posible una lectura cíclica de los datos de la memoria FIFO. Se tiene la necesidad de que una vez leído el dato de la memoria este tiene que escribirse de nuevo en la memoria, manteniendo la posibilidad de leer la memoria FIFO cuantas veces sea necesario, para ello una vez iniciada la reproducción de la FIFO se pierde el primer ciclo de reloj, habilitando la escritura en el siguiente ciclo, manteniendo activos la lectura y escritura de la memoria FIFO hasta que el usuario detenga la reproducción cíclica.

### 3.8 Firmware para la configuración del voltaje de salida según la tecnología a utilizar.

Con la intención de hacer al dispositivo más versátil, se realiza el firmware donde se capturan los datos y la dirección del DAC a utilizar, junto con el circuito electrónico analógico encargado de mostrar los diferentes niveles lógicos según la necesidad del usuario. Para ello se hace uso de los siguientes bloques: DACbuffer\_Reference, Bus patrón, dac1datos,..., dac24datos.

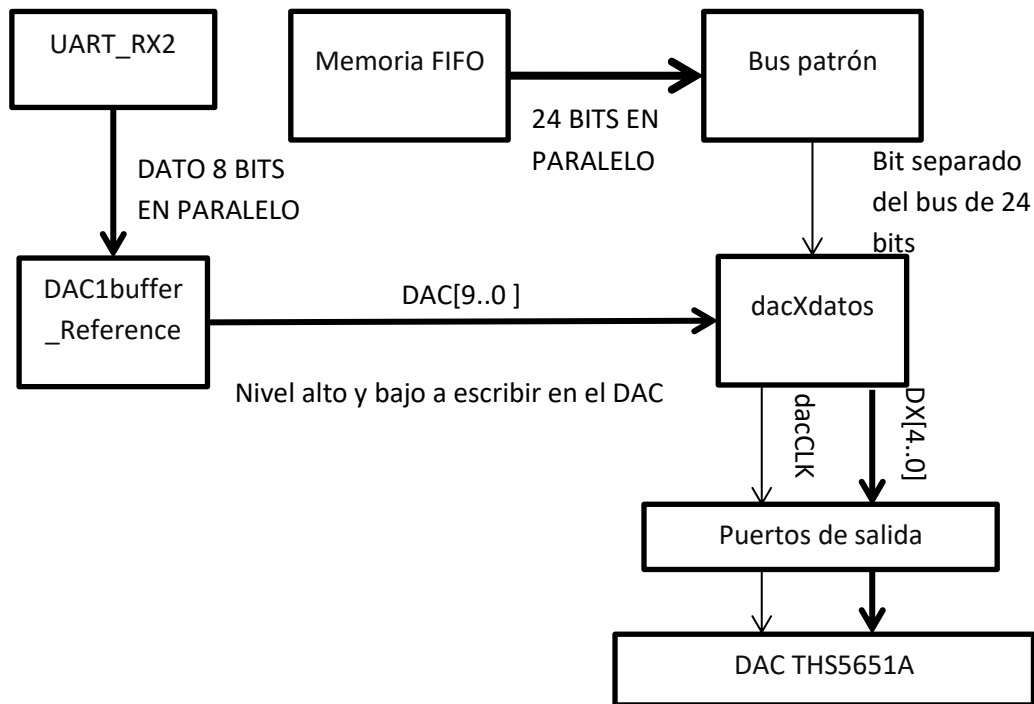


Figura 3.30: Diagrama a bloques del firmware del acoplamiento de voltaje.

El diagrama de bloques de la figura 3.30 muestra la parte del firmware que se encarga de configurar el nivel de voltaje para cada salida del patrón digital, es importante señalar que esta parte del firmware trabaja de forma paralela con el administrador de la memoria y en conjunto con la salida de la memoria FIFO.

Con este firmware y el circuito acoplador de voltaje, el usuario puede trabajar con diferentes señales digitales como son: TTL, CMOS, ECL y PECL. Además, puede elegir un voltaje personalizado para representar los niveles lógicos. Con todo esto se pueden realizar diferentes pruebas a dispositivos digitales, ayudando a entender su funcionamiento y a realizar caracterizaciones en menos tiempo, optimizando el tiempo del usuario y en consecuencia su producción.

### 3.8.1 Dacbuffer Reference

Este módulo se encarga de recibir el comando 230 (F6 hex) y proceder a obtener los 10 bits que representan el nivel alto y nivel bajo de cada salida del generador de patrones digitales, además de recibir la dirección a la que pertenecen los datos recibidos. Al final muestra los datos recibidos en la dirección asignada para que el módulo **dacXdatos** lo reciba y los mande al DAC según sea el nivel deseado.

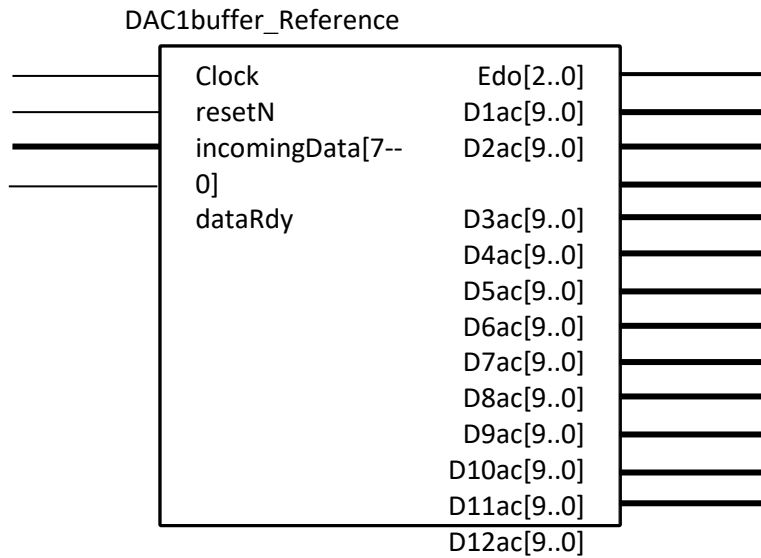


Figura 3.31: Bloque para recibir la palabra de 10 bits donde viene la dirección, el nivel alto y bajo de la salida asignada.

En este módulo, figura 31, se llevan a cabo tres acciones, se construye un buffer de 40 bits, después se determina si el comando recibido es 230 (E6 hex), para finalmente proceder a obtener la dirección donde van a ser escritos los 10 bits a enviar al bloque dacXdatos conectado en la dirección recibida.

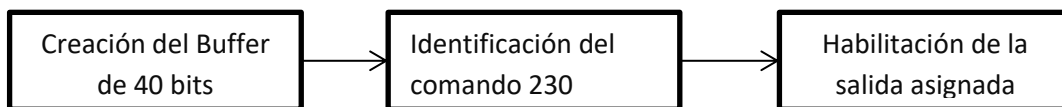


Figura 3.32: Diagrama a bloques del módulo Dacbuffer Reference.

La figura 3.32 muestra la secuencia para la configuración del nivel de voltaje de cada salida, así cada salida puede tener un nivel de voltaje independiente.

La máquina de estados de este bloque, ver figura 3.33, se describe a continuación: En el estado S0 se escribe al DAC la palabra 0x15 para fijar la salida de la tarjeta acopladora de voltaje en 0V, en el estado S1 se crea un bus de 40 bits. El estado S2 revisa que el comando recibido sea 230 (E6 hex), pasando al estado S3 donde se obtiene la salida asignada y sus niveles de voltaje. Por último en el Estado S4 se escribe a la salida asignada, las palabras que representan el nivel bajo y alto la salida.

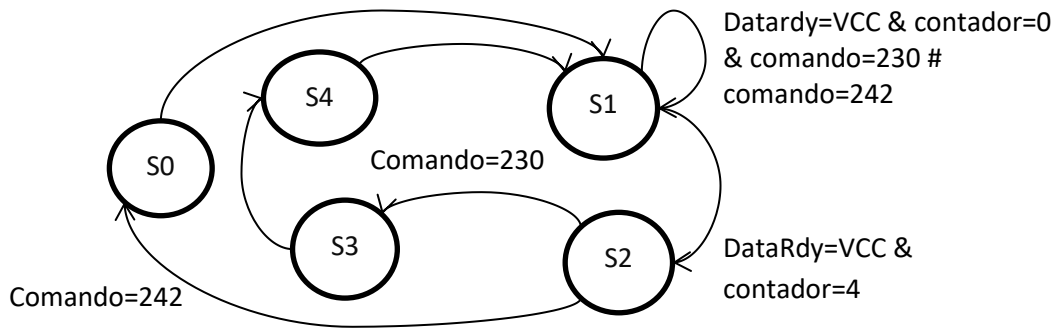


Figura 3.33: Diagrama de máquina de estados del módulo Dacbuffer\_Reference.

### 3.8.2 Bus patrón

En este bloque descompone el bus de datos que sale de la memoria FIFO y muestra el valor de cada bit del bus en una salida individual. Esta salida se envía al bloque Dacxdatos, indica la palabra que se escribe a la entrada del DAC.

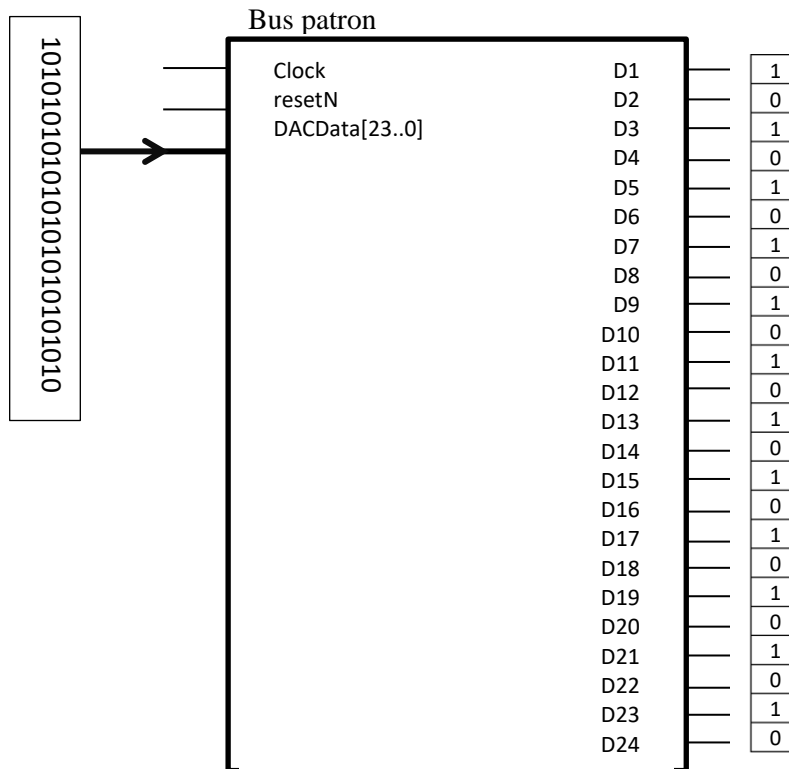


Figura 3.34: Módulo P1atterndata.

Cada salida de este módulo, ver figura 3.34, se conecta a la entrada “gen” del módulo “DacX1datos” correspondiente, esta señal es utilizada para determinar la palabra, a escribir al DAC, que

corresponda al nivel lógico de esta señal. Cada señal puede tener su propio nivel de voltaje lógico independiente de las demás salidas.

### 3.8.2 Dac1datos---dac24datos.

El DAC TSH5651A además de los datos a su entrada, necesita de una señal de reloj para indicarle que cargue esos datos y proceda a realizar la conversión.

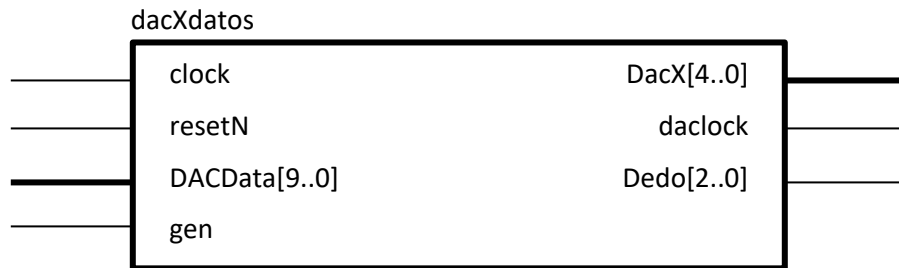


Figura 3.35: Módulo DacXdatos.

El módulo de la figura 3.35 recibe en su entrada, DACdata[9..0], los 10 bits que representan los dos niveles lógicos de una señal digital, los cinco bits más significativos pertenecen al nivel alto y los cinco menos significativos pertenecen al nivel bajo. El valor de la entrada "gen", es el encargado de cambiar el valor de la salida DacX[4..0] y en cada cambio se genera un pulso de reloj con el que se indica al DAC THS5651A que tiene un dato nuevo a convertir en su entrada.

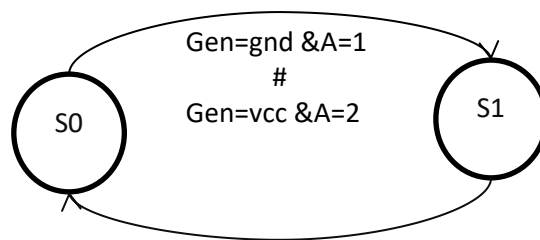


Figura 3.36: Diagrama de máquina de estados del módulo DacXdatos.

La máquina de estados del bloque DacXdatos, se muestra en la figura 3.36, en el estado S0 se escribe la palabra a convertir por el DAC y de ahí en el estado S1 se manda la señal de reloj al DAC para que realice la conversión del dato presente en las entradas del DAC y regresa al estado S0.

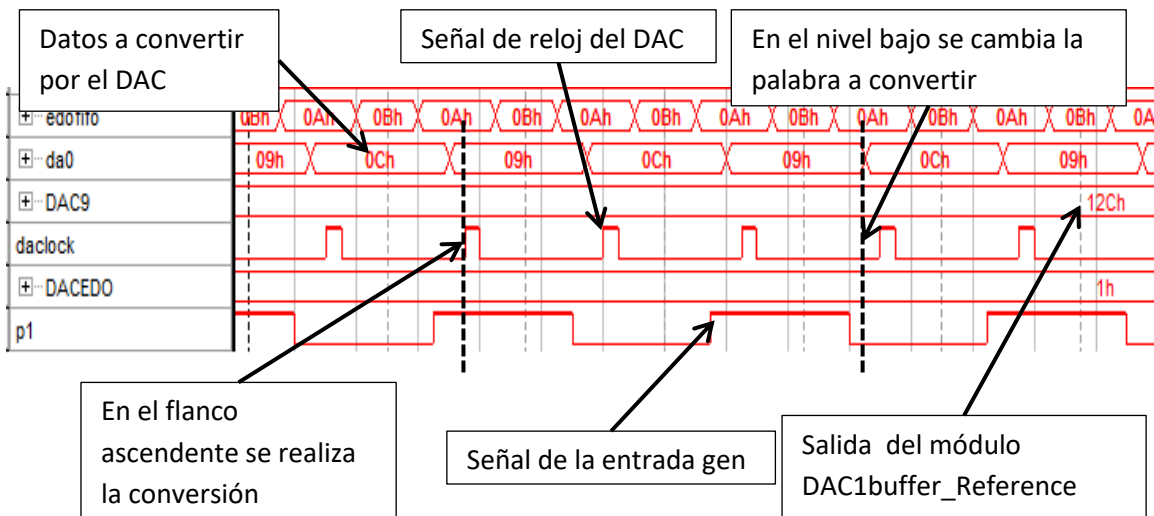


Figura 3.37: Gráfica de las salidas del módulo DacXdatos.

La condición para mostrar estos bits a su salida, está sujeta al nivel lógico de la entrada “gen”, ver figura 3.37, la cual recibe el valor que tiene ese bit en el patrón digital. Después de cada cambio el módulo genera la señal de reloj a la vez que mantiene la palabra que representa el nivel lógico de la entrada “gen”, haciendo posible que el DAC cambie el nivel de voltaje de su salida, en base a la palabra que ha sido escrita en su entrada de datos a convertir.

De esta manera se pueden alcanzar diferentes niveles de voltaje, los cuales representan diferentes tecnologías digitales haciendo más versátil al generador de patrones digitales, pues se amplía el campo de aplicaciones, debido a que no solo trabaja con la tecnología TTL o CMOS si no también con la tecnología ECL y PECL. Además, el usuario puede configurar el nivel de voltaje deseado; en un rango de -5V a 5V con una resolución de 0.31V; según la necesidad que llegase a tener.

### 3.9 Conclusiones

En la actualidad la mayoría de los dispositivos digitales requieren de un firmware para su funcionamiento. Este le dice que hacer en las situaciones que allá considerado el diseñador, todo dispositivo que se quiera describir como inteligente necesita de esa programación que le ayude a tomar decisiones, en base al algoritmo grabado en su memoria. El correcto planteamiento del firmware determinara la calidad y eficacia del dispositivo diseñado.

## CAPITULO 4: Software de la interfaz de usuario

En este capítulo se describe el software de la interfaz de usuario del generador de patrones digitales. La interfaz de usuario está diseñada para que su uso sea intuitivo y no sea necesario un manual extenso para describir el uso y funcionamiento del generador de patrones digitales.

Para facilitar la captura del patrón digital se desarrolló el siguiente software. En el cual primero se muestra la pestaña de CONFIGURACION, adentro de esta pestaña se tienen las pestañas SALIDAS, VOLTAJE y DATOS. En la pestaña SALIDAS se captura el nombre del bus de datos, se asignan los bits de salida. En la pestaña VOLTAJE, el usuario selecciona la tecnología digital con la que quiere trabajar, o de manera manual asigna el voltaje que represente los niveles lógicos alto y bajo. En la pestaña DATOS se captura el patrón digital o se lee de un archivo generado anteriormente, se configura la frecuencia deseada de reproducción y se selecciona el tipo de reproducción SINGLE (UNA) o CICLICA. Por último para enviar los datos e iniciar la reproducción del patrón digital, se selecciona la pestaña GRÁFICAS.

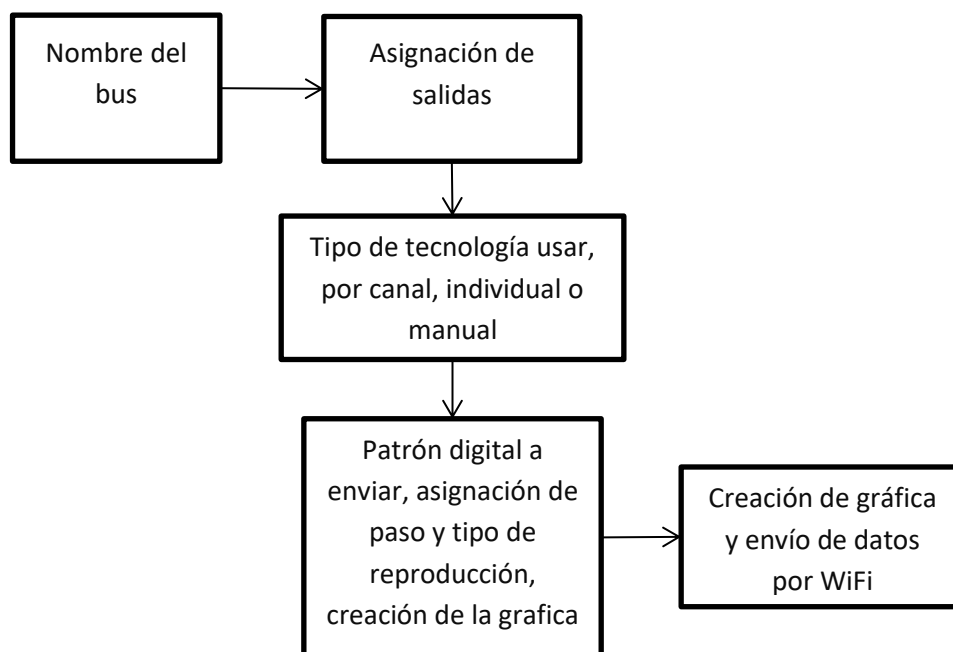


Figura 4.1: Descripción general del funcionamiento del software.

El diagrama a bloques de la figura 4.1 explica a grandes rasgos el funcionamiento del software del generador de patrones digitales.



La figura 4.4 describe la ventana de salidas, en donde el usuario habilita y asigna las salidas que va a utilizar.



Figura 4.4: Descripción de la pestaña de asignación de salidas.

En el proceso de asignación de salidas, el usuario podría realizar la asignación de una salida a dos etiquetas diferentes, lo que nos llevaría a una situación de “error”. Una salida no puede ser asignada a dos etiquetas al mismo tiempo, para ello se realiza un escaneo para determinar si cada posible salida solo pertenece a una sola etiqueta, en caso de que llegara a detectar lo contrario se muestra una señal de “error” encendiendo un indicador de color rojo, indicando al usuario de que debe revisar su asignación de salidas, pues ha repetido una o más salidas en diferentes etiquetas. Cuando el usuario haya realizado las correcciones necesarias, el indicador cambiara al color verde indicando que la asignación de salidas es correcta.

Para poder realizar el escaneo de las salidas asignadas, primero se identifica cuantos buses o etiquetas han sido creados, posteriormente se lee cada columna de la matriz de salidas y se suman los valores lógicos de cada renglón de la columna, es decir cuando un renglón es seleccionado tiene un valor de “1” y cuando no es seleccionado obtiene un valor de “0”. Se suman todos los elementos

de cada columna y si en alguna de ellas el valor de la suma es mayor a 1, indica que esa salida tiene más de una etiqueta asignada y por lo tanto el color de la barra de error cambia de verde a rojo.

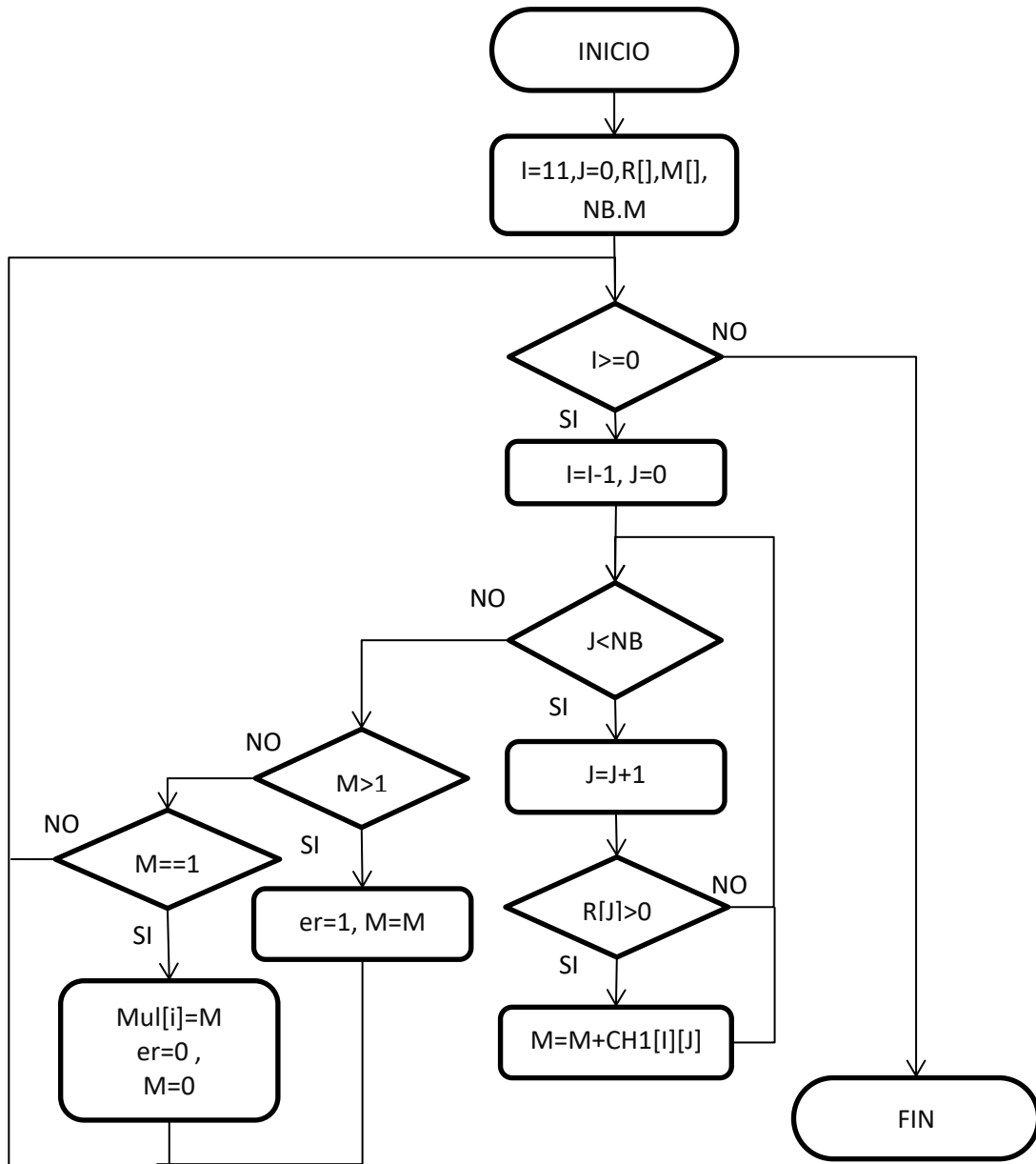


Figura 4.5: Proceso de identificación de error en la asignación de salidas, del canal 1.

Las variables del diagrama de flujo de la figura 4.5 son: NB: número de buses; J: renglón, i: columna, M: suma de los renglones de la columna[i], er: indicador de error, Mul[i] vector que contiene las salidas habilitadas, R[]: guarda el número de caracteres de cada bus creado; esto se realizó para no

tomar en cuenta a las secciones que no fueron nombradas y se encuentran en medio de las que si lo están .

El proceso de asignación de salidas y verificación de error se muestra en el diagrama de flujo de la figura 4.5, en el cual se explica que el arreglo del canal 1 se escanea columna por columna y en cada habilitación se asigna el número 1 y se suma con los demás renglones asignados en la misma columna, por lo tanto si la suma es mayor a 1 indica que una salida ha sido asignada a más de una etiqueta y se indica el error de la asignación de salidas, ver figura 4.6.

Este proceso se repite 12 veces en cada canal y se forman dos vectores horizontales los cuales contienen la habilitación de las salidas de los dos canales.



El color rojo indica que hay un error en la asignación de salidas



El color verde indica que no hay algún problema en la asignación de salidas

Figura 4.6: Descripción de los colores del indicador de error.

En dado caso, que el usuario quisiera no atender la señal de error, al pasar a la pestaña de datos, podrá capturar o editar el patrón digital, pero no podrá visualizarlo ni enviarlo. Para ello será necesario que corrija los errores que tenga en la asignación de salidas hasta que el indicador de error cambie a color verde, avisando que no hay problemas en la asignación de salidas.

En la revisión de cada columna, si la suma es igual o menor a 1, este resultado se escribe en un arreglo de una dimensión llamado `mul[]` en la posición "i". De esta manera se obtiene un arreglo con el cual se peina el arreglo que guarda el patrón digital a enviar, así solo se envían los datos que tienen habilitada su salida. Esto se logra utilizando la función AND entre el resultado de las salidas asignadas y cada renglón del patrón digital asignado. Véase figura 4.7.

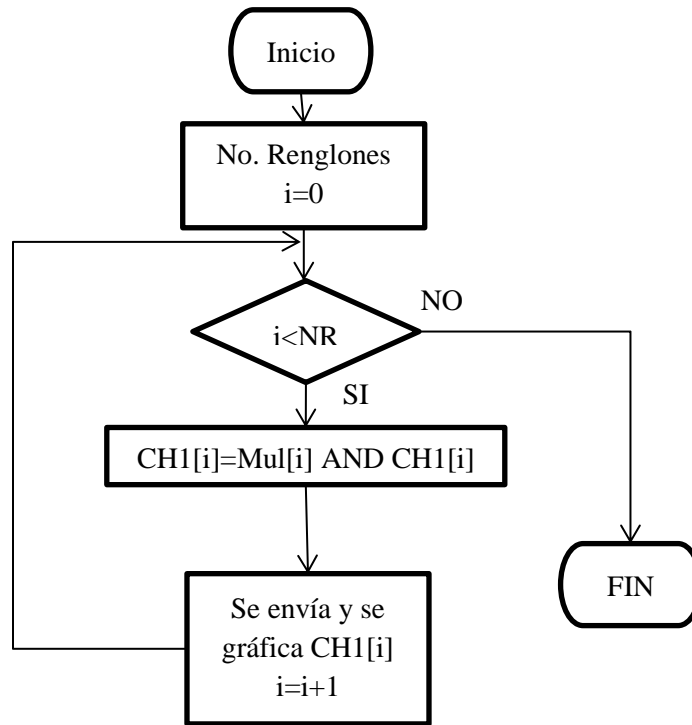


Figura 4.7: Diagrama de flujo de los datos a enviar y graficar según la habilitación de las salidas.



Figura 4.8: Ejemplo de error en la asignación de salidas.

La figura 4.8 muestra un ejemplo de una doble asignación de una salida a dos buses diferentes, esto crea un error y la barra indicadora cambia a color rojo.

## 4.2 Voltaje de los niveles lógicos de las salidas.

Por niveles lógicos se conocen los intervalos de tenciones que interpretan como un cero o un uno lógicos a la entrada y a la salida. Estos valores de tensión se especifican utilizando unos valores límites; la figura 75 muestra los rangos de los niveles lógicos para una señal TTL [27]:

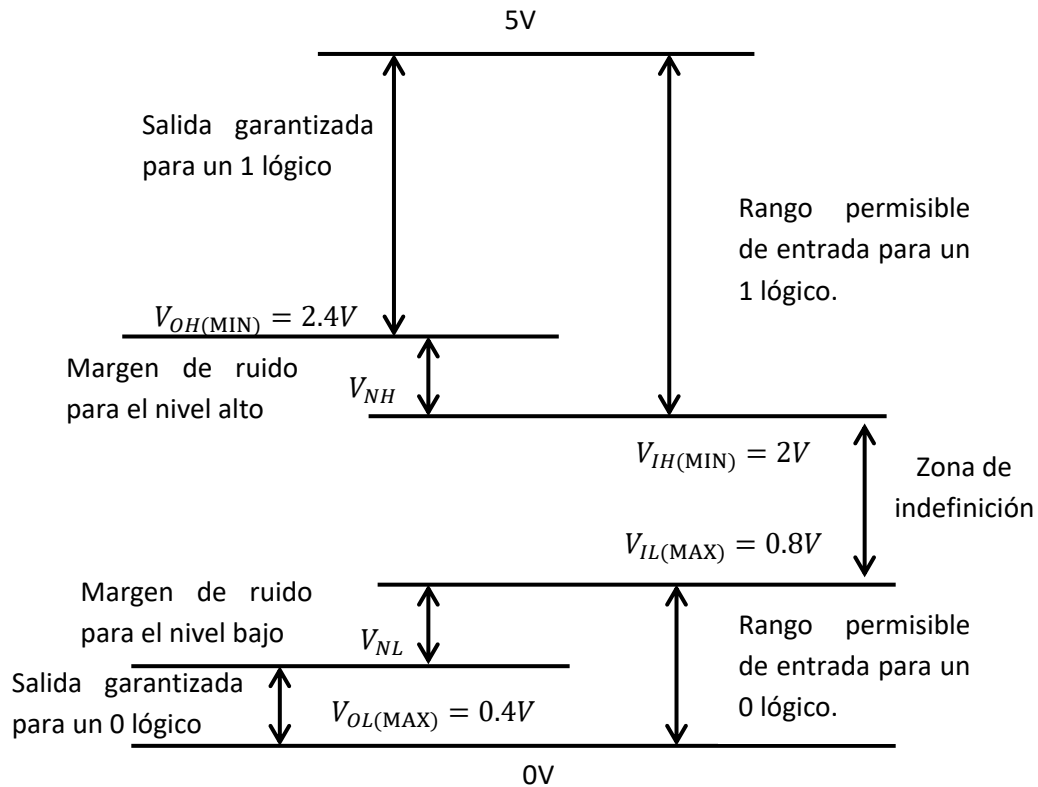


Figura 4.9: Niveles lógicos de entrada-salida y márgenes de ruido para puertas TTL estándar.

$V_{OH(MIN)}$ : Tensión mínima que produce la salida de un dispositivo digital cuando genera un 1 lógico.

$V_{OL(MAX)}$ : Tensión máxima que produce la salida de un dispositivo digital cuando genera un 0 lógico.

$V_{IH(MIN)}$ : Tensión mínima que puede aplicarse a la entrada de un dispositivo digital para que interprete la señal como un 1 lógico.

$V_{IL(MAX)}$ : Tensión máxima que puede aplicarse a la entrada de un dispositivo digital para que interprete la señal como un 0 lógico.

En la figura 4.9 se muestran estos valores para la familia TTL que son:

$$V_{OH(MIN)} = 2.4V$$

$$V_{OL(MAX)} = 0.4V$$

$$V_{IH(MIN)} = 2V$$

$$V_{IL(MAX)} = 0.8V$$

Para la familia CMOS los valores de salida coinciden con los de alimentación y para los de entrada hay un 30% de tolerancia. Así para un dispositivo CMOS alimentado a 5V se tiene que:

$$V_{OH(MIN)} = 5V$$

$$V_{OL(MAX)} = 0V$$

$$V_{IH(MIN)} = 3.5V (70\% \text{ de } V_{DD})$$

$$V_{IL(MAX)} = 1.5V (30\% \text{ de } V_{DD})$$

#### 4.2.1 Margen de ruido.

El ruido se define como toda perturbación que afecta al correcto funcionamiento del sistema digital. Estas perturbaciones son tensiones inducidas, que pueden presentarse bien como impulsos de tensión alterna o como una componente de continua, que se superpone y modifica la amplitud de las señales digitales con las que trabaja. Las fuentes de ruido son diversas, radiaciones electromagnéticas de alta frecuencia, variaciones de la tensión de alimentación, etcétera. Para no verse afectados por el ruido, los circuitos digitales deben tener una cierta tolerancia a las fluctuaciones de tensión en su entrada, sin que cambie la tensión de salida. [27]

Se conoce como margen de ruido, de una familia de circuitos integrados digitales, la amplitud máxima de la perturbación que puede producirse a la entrada de una puerta sin que repercuta en su salida, por lo tanto el margen de ruido cuantifica o mide la inmunidad al ruido de un circuito y se expresa en voltios. [27]

Para un circuito integrado se establecen dos valores de margen de ruido, que no tienen por qué ser iguales: margen de ruido para nivel alto  $V_{NH}$  y margen de ruido para nivel bajo  $V_{NL}$  que se definen como [27]:

$$V_{NH} = V_{OH(MIN)} - V_{IH(MIN)}$$

$$V_{NL} = V_{IL(MAX)} - V_{OL(MAX)}$$

Para puertas TTL el margen de ruido es de 0.4V mientras que para las puertas CMOS es un 30% de la tensión de alimentación.

## 4.2.2 Compatibilidad con las tecnologías digitales

La tecnología de un circuito digital está relacionada con los niveles de voltaje utilizados para representar un 1 o un 0 lógico, lo cual representa la tecnología utilizada en ese circuito digital. Los diferentes niveles de voltaje, alcanzados por el software y hardware del generador de patrones digitales son los siguientes:

Tabla 4.1: Niveles de voltaje de las salidas del generador de patrones digitales.

TECNOLOGIA	VOLTAJE
<b>a. TTL</b> Lógica de Transistor a Transistor	1:5V 0:0V
<b>b. LVCMOS</b> Low Voltaje Complementary Metal Oxide Semiconductor	1:3.3 0:0-0.8V
<b>c. PECL</b> Positive Emitter Coupled Logic	1: $V_{CC}-0.8$ 0: $V_{CC}-1.8$
<b>d. ECL</b> Emitter Coupled Logic	1:-0.8 0:-1.8

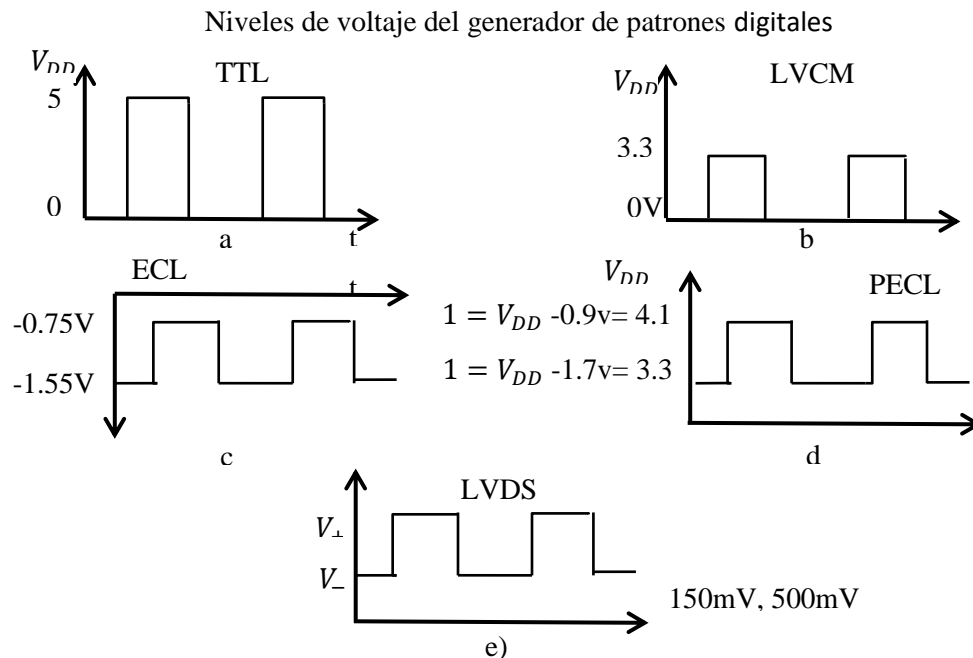


Figura 4.10: Niveles de voltaje de las salidas del generador de patrones digitales.

El generador de patrones digitales reproduce el patrón digital a diferentes niveles lógicos, pues las tecnologías digitales varían en cuanto al nivel lógico de voltaje asignado a un “1” o un “0”. Por ello es de suma importancia poder reproducir el patrón digital a los niveles lógicos más comerciales como son CMOS, TTL, ECL, PECL; véase figura 4.10.

Desde software se envía el comando 230, junto con las palabras para el voltaje de nivel alto y bajo del patrón digital en los 10 bits menos significativos. La dirección de la salida asignada se escribe en los siguientes cinco bits. La figura 4.11 muestra como está organizado el bus de 40 bits con el comando 230, “E6”.

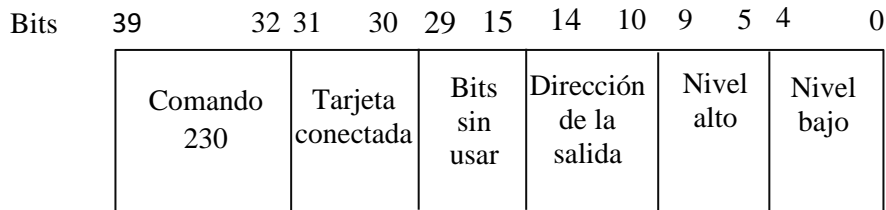


Figura 4.11: Estructura del comando 230.

Para habilitar y deshabilitar la asignación de los diferentes niveles de voltaje, se coló un botón con el que se indica que la tarjeta para los diferentes niveles de voltaje digitales, está conectada al FPGA, en consecuencia se habilita la configuración del nivel de voltaje para cada salida del patrón digital. Si el botón está en la posición de “ON” en los bits 31 y 30 se envía el número dos y si está en la posición de “OFF” se envía el número uno, de esta manera se le indica al firmware si los datos salen directamente del FPGA o tienen que pasar por la tarjeta de acoplamiento de voltaje.

Para llevar a cabo la configuración del nivel de voltaje de cada salida, se propusieron tres posibilidades, “POR CANAL”, “INDIVIDUAL”, “EDITAR NIVEL”.

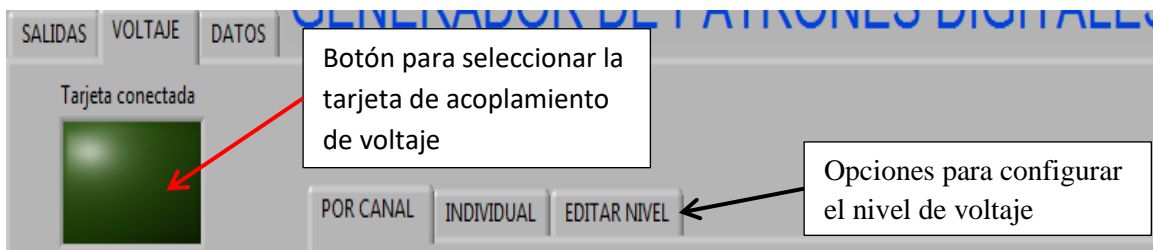


Figura 4.12: Opciones de configuración de los niveles de voltaje de las salidas.

La figura 4.12 muestra la pestaña “Voltaje” y describe las opciones de configuración de los posibles niveles de voltaje. La figura 4.13 es el diagrama de flujo de la pestaña “Voltaje”, en el cual se explican las opciones de asignación de voltaje.

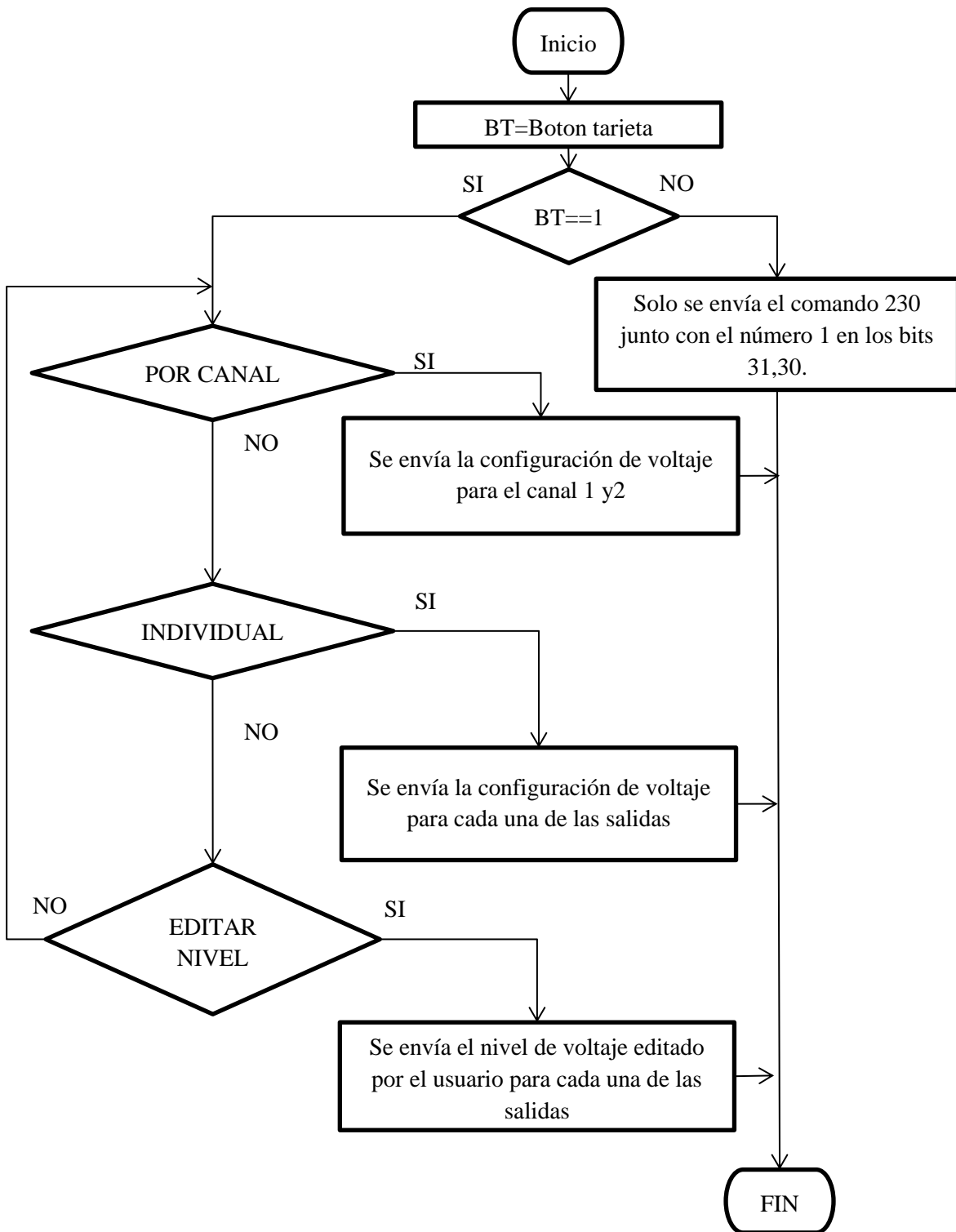


Figura 4.13: Diagrama de flujo de la habilitación de la tarjeta de acoplamiento de voltaje y configuración.

### 4.2.1 Por canal

En esta opción los bits habilitados; en la pestaña salidas; de cada canal se configuran al mismo nivel de voltaje, tanto en alto como en bajo, esto no implica que los bits habilitados del otro canal tengan la misma configuración de voltaje. Para ello en esta pestaña se presentan dos cajas, cada una contiene las 5 opciones posibles para cada canal, las cuales son TTL, CMOS, ECL, PCEL y EDITAR. Véase figura 4.14.

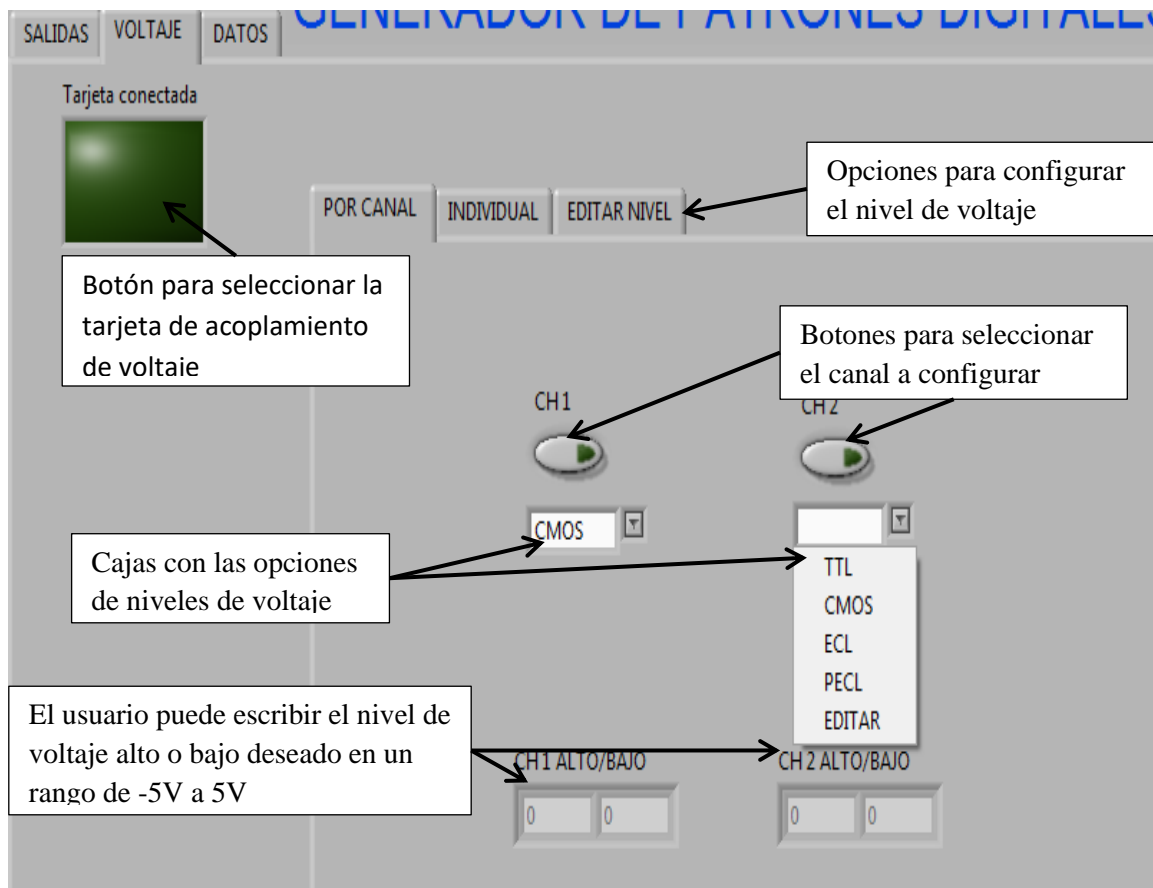


Figura 4.14: Descripción de la pestaña por canal de la configuración de voltaje.

En la figura 4.14 se describen los botones de configuración de la pestaña "por canal". La figura 4.15 describe el funcionamiento de la pestaña "por canal" y explica la asignación del nivel de voltaje, para el canal 1, siendo el mismo procedimiento para el canal 2.

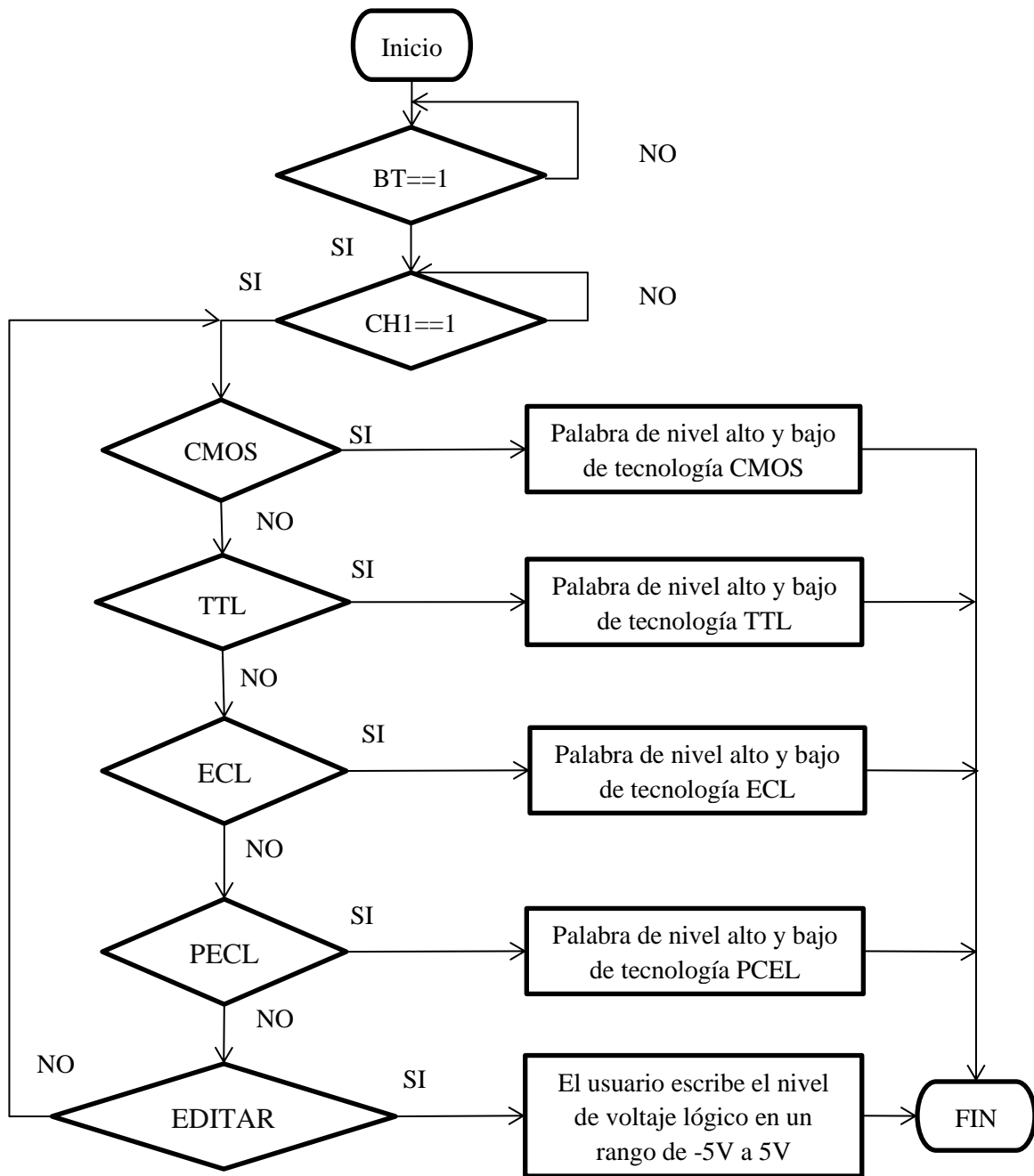


Figura 4.15: Diagrama de flujo de la selección de niveles de voltaje por canal 1 para el canal 2 es el mismo procedimiento.

### 4.2.1 Individual

En esta opción el usuario puede seleccionar el tipo de tecnología para cada salida, el sistema es más flexible, debido a que cada salida de cada canal puede tener un nivel de voltaje diferente, haciendo posible que se puedan hacer pruebas con diferentes tecnologías al mismo tiempo, hacer comparaciones entre ellas, así como experimentar con la compatibilidad entre ellas.

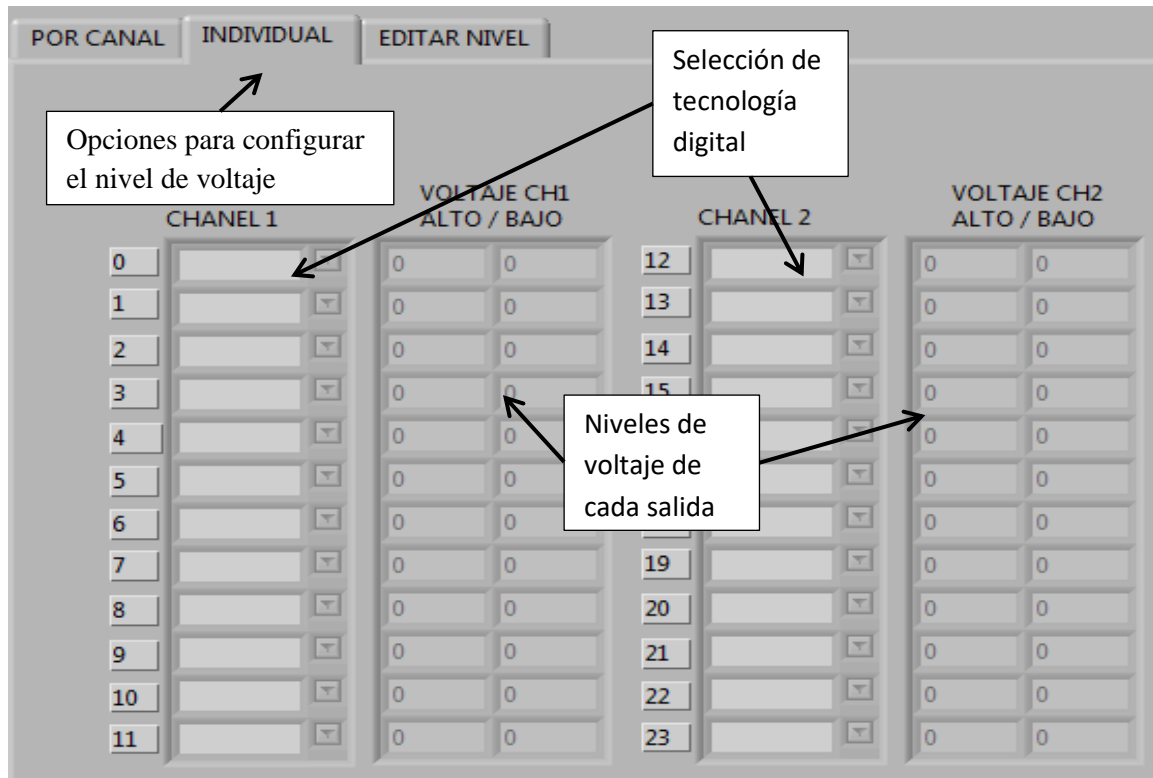


Figura 4.16: Descripción de la pestaña "individual" de la configuración de voltaje.

La figura 4.16 muestra las opciones de configuración en la pestaña individual. Cada salida puede tener una tecnología digital diferente. El diagrama de flujo de la figura 4.17 explica el funcionamiento de la pestaña "individual", en donde para llevar a cabo la lectura de cada una de las salidas de cada canal, se realiza un ciclo FOR. En cada interacción del ciclo se realiza una comprobación para conocer si la salida configurada a cierta tecnología fue habilitada en la pestaña "SALIDAS", si no lo fue entonces los niveles de voltaje para esa tecnología no son escritos en el arreglo de "NIVELES DE VOLTAJE CH1, CH2". Asegurando que esa salida mantenga un voltaje a su salida de cero volts, pues no fue habilitada en la pestaña salidas.

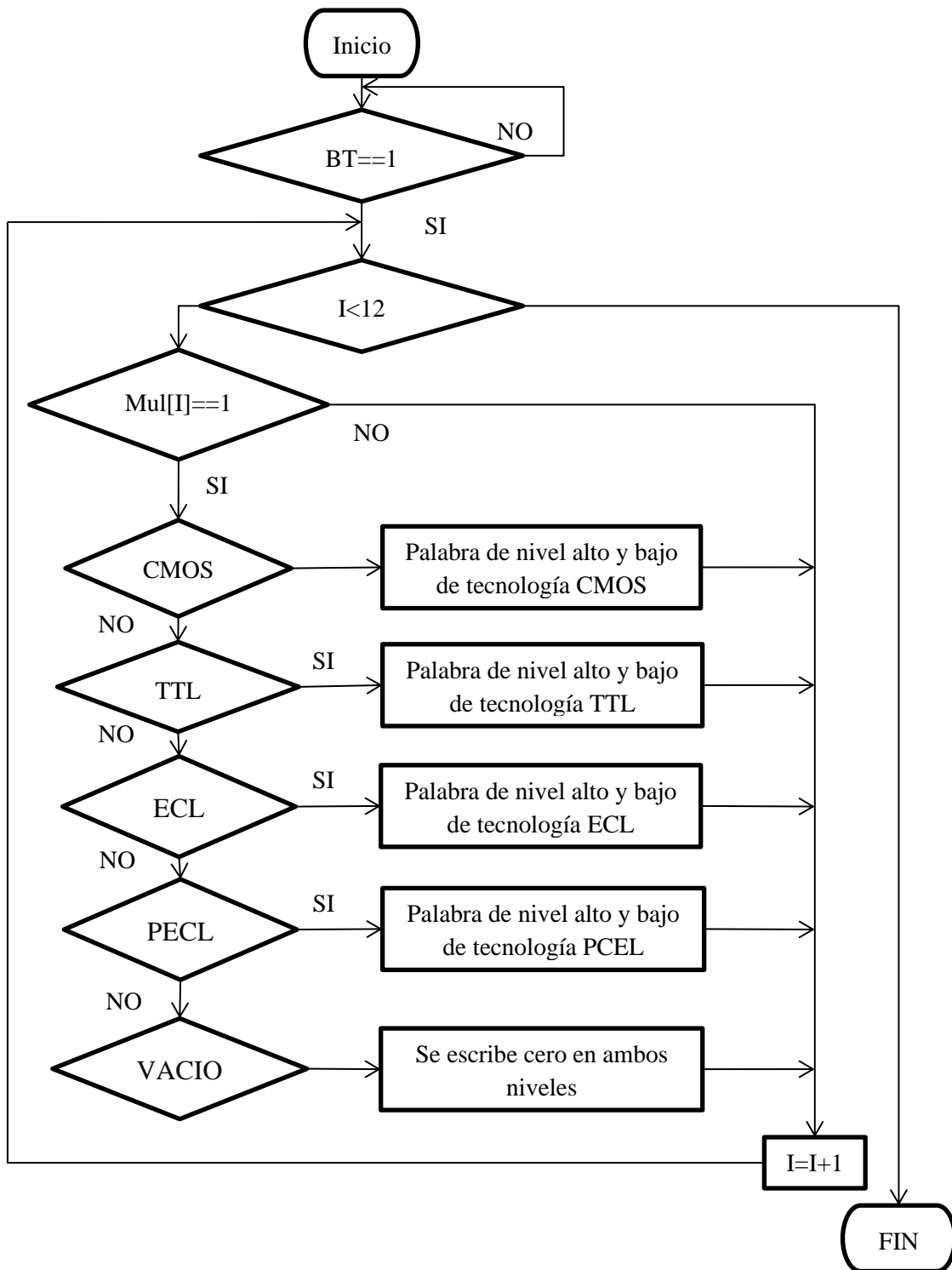


Figura 4.17: Diagrama de flujo de la configuración de salidas de manera individual.

### 4.2.3 Editar nivel

El usuario puede configurar el nivel lógico de cada salida, de manera independiente. Permiéndole caracterizar los componentes digitales, respecto al rango de nivel de voltaje en el que el dispositivo puede trabajar. El usuario podrá implementar otros niveles de voltaje que requiera en la lógica a desarrollar. En la figura 4.18 se describe la pestaña configurar "Editar nivel".

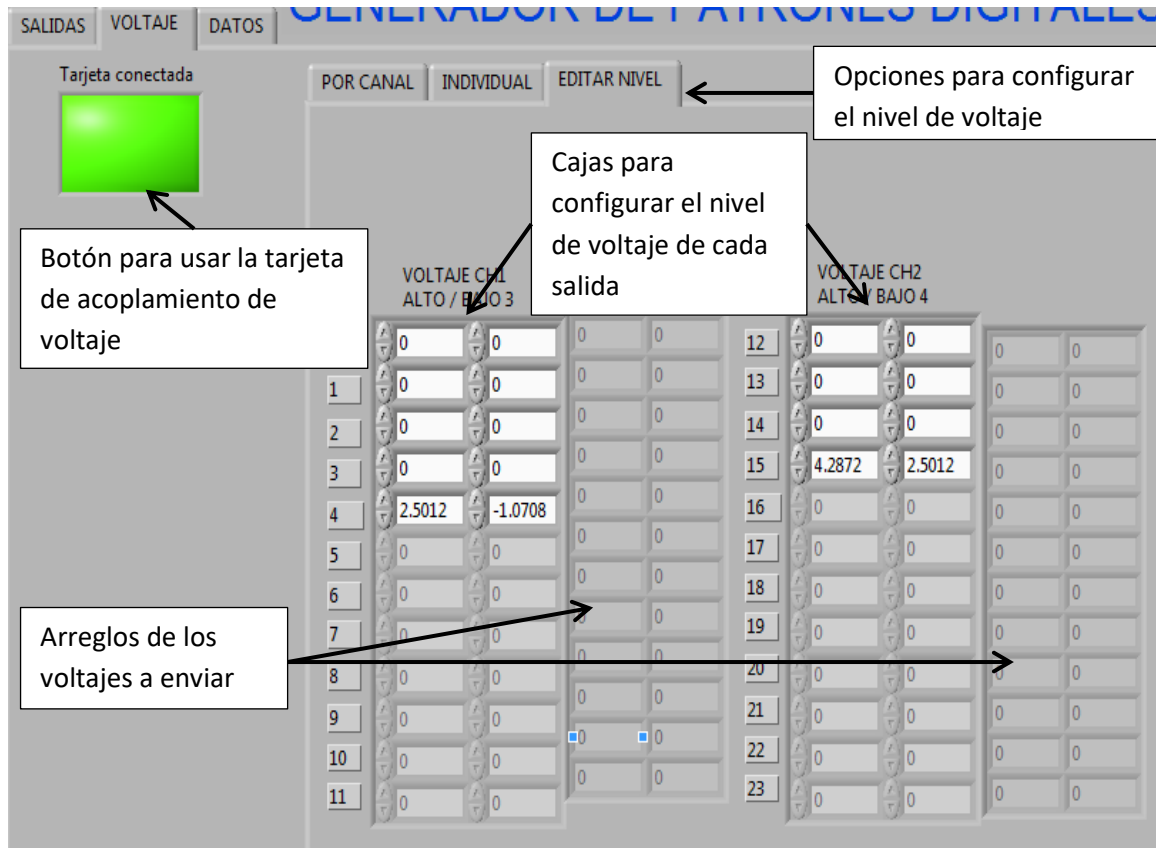


Figura 4.18: Pestaña de la opción "Editar Nivel".

Antes de ingresar el valor editado por usuario, al arreglo de los voltajes a enviar, se comprueba que la salida con el voltaje asignado este habilitada, en caso contrario, a ese espacio se le asigna un cero pues la salida no esta habilitada. Cada caja de asignacion de voltaje tiene un rango el cual limita el voltaje minimo (-5V) ó maximo (5V) que se pueda escribir en cada una de ellas.

Debido a la resolucion de 32 bits y la ganancia del circuito de acoplamiento, el usuario tiene un incremento ó decremento de 0.33384V cada vez que precione el botón de aumentar o disminuir, el numero ira aumentando o disminuyendo cada 0.33384V.

En la figura 4.19 se explica el funcionamiento de la pestaña "Editar nivel". En donde, al igual que en las opciones anteriores cada salida es verificada con el vector de salidas habilitadas MUL[].

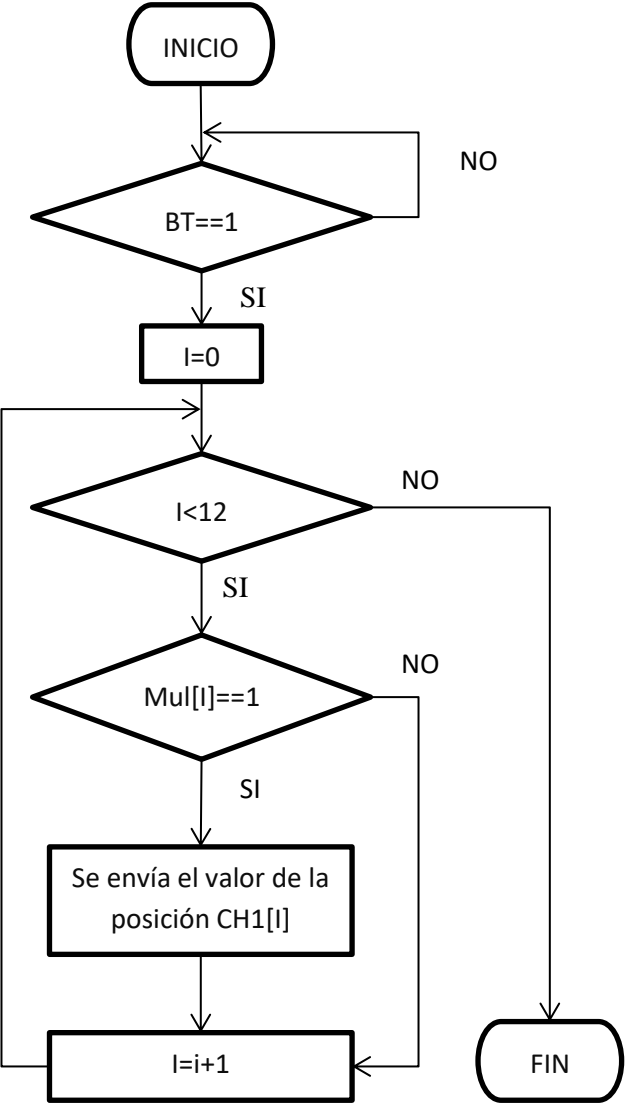


Figura 4.19: Diagrama de flujo que describe el funcionamiento de la pestaña "Editar nivel".

Para CH2 el proceso es el mismo y la asignación de niveles de voltaje está condicionada con las salidas habilitadas de CH2.



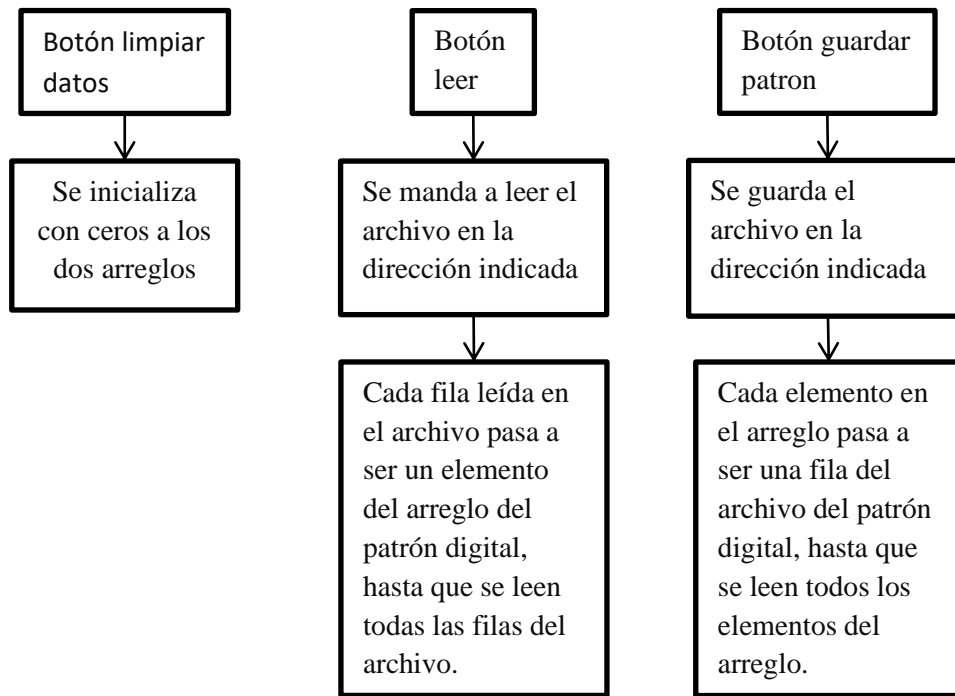


Figura 4.21: Acciones a realizar por los botones: limpiar datos, leer, guardar patrón.

En la figura 4.21 se describen los botones a utilizar con el patrón digital. Para poder guardar un patrón digital capturado, primero se debe escribir la dirección donde se va a guardar el archivo generado; esta se escribe en la caja “Dirección”, después se nombra al archivo en la caja “File name” y por ultimo presionamos el botón “guardar patrón”. Esto creara un archivo con el nombre asignado en la dirección asignada.

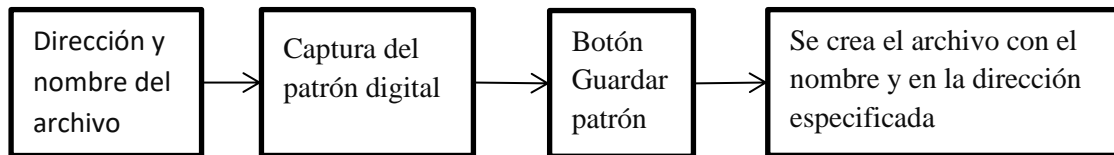


Figura 4.22: Diagrama a bloques para guardar un nuevo patrón digital.

El proceso para guardar un patrón digital recién capturado se explica en la figura 4.22. Para poder leer un patrón digital previamente capturado, primero se debe escribir la dirección donde se encuentra el archivo, en la caja “Dirección”, después se escribe el nombre del archivo en la caja “File name” y por ultimo presionamos el botón “leer”. La información leída se escribe en los arreglos del patrón digital y después se manda a graficar para visualizar el patrón previamente guardado. La figura 4.23 describe el proceso para leer un patrón digital previamente guardado.

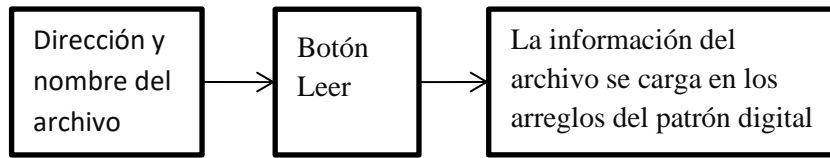


Figura 4.23: Diagrama a bloques para leer un patrón digital previamente guardado.

La descripción de la pestaña “datos” se realiza en la figura 4.24.

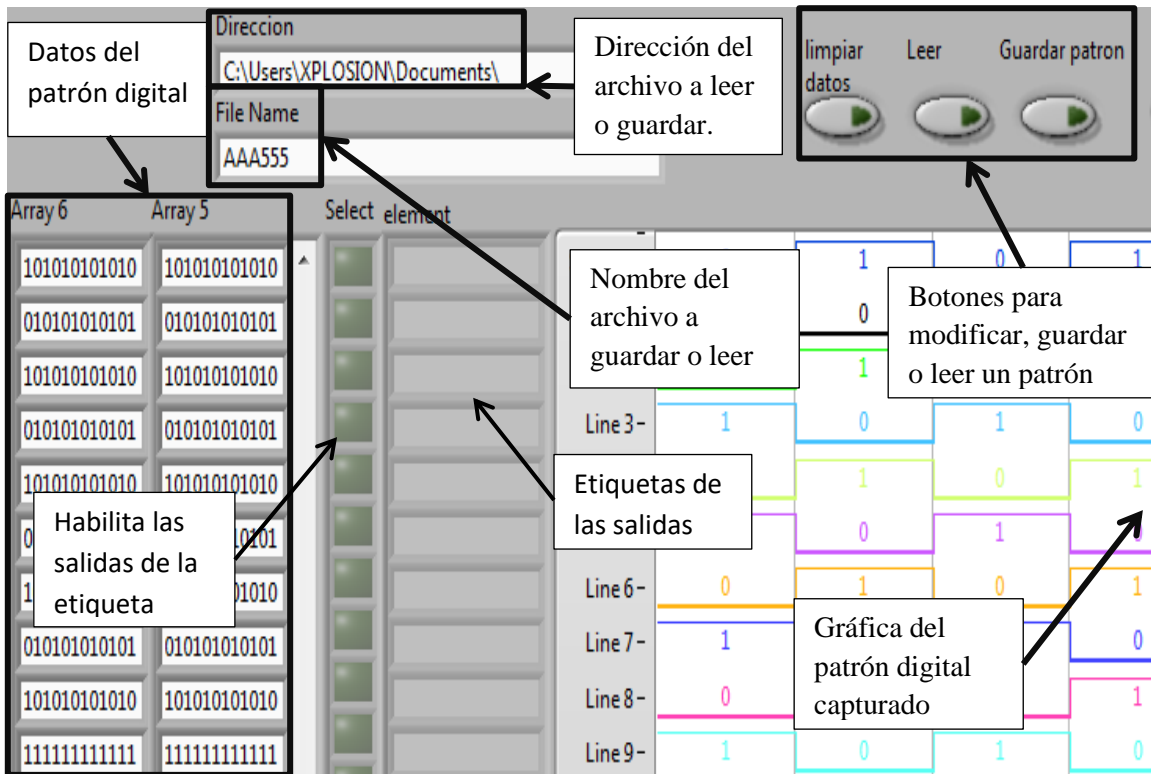


Figura 4.24: Descripción de la pestaña datos.

Cada vez que el usuario ingresa un valor al patrón digital, la gráfica se actualiza mostrando el patrón digital capturado por el usuario. De esta manera el usuario puede observar la gráfica del patrón digital antes de que sea enviado al FPGA y hacer las correcciones necesarias en caso de que fuera necesario. Los botones del arreglo “SELECT” habilitan las salidas de las etiquetas editadas en la pestaña “SALIDAS”; vease figura 4.24; si el botón no está pulsado, las salidas de la etiqueta que se muestre adelante del botón, estarán deshabilitadas y por lo tanto no se mostrarán en la gráfica del patrón digital y tampoco serán enviadas al firmware para su reproducción.

La posibilidad de habilitar y deshabilitar las salidas asignas a las etiquetas, permite al usuario seleccionar las salidas deseadas sin tener que modificar o editar el patrón digital.

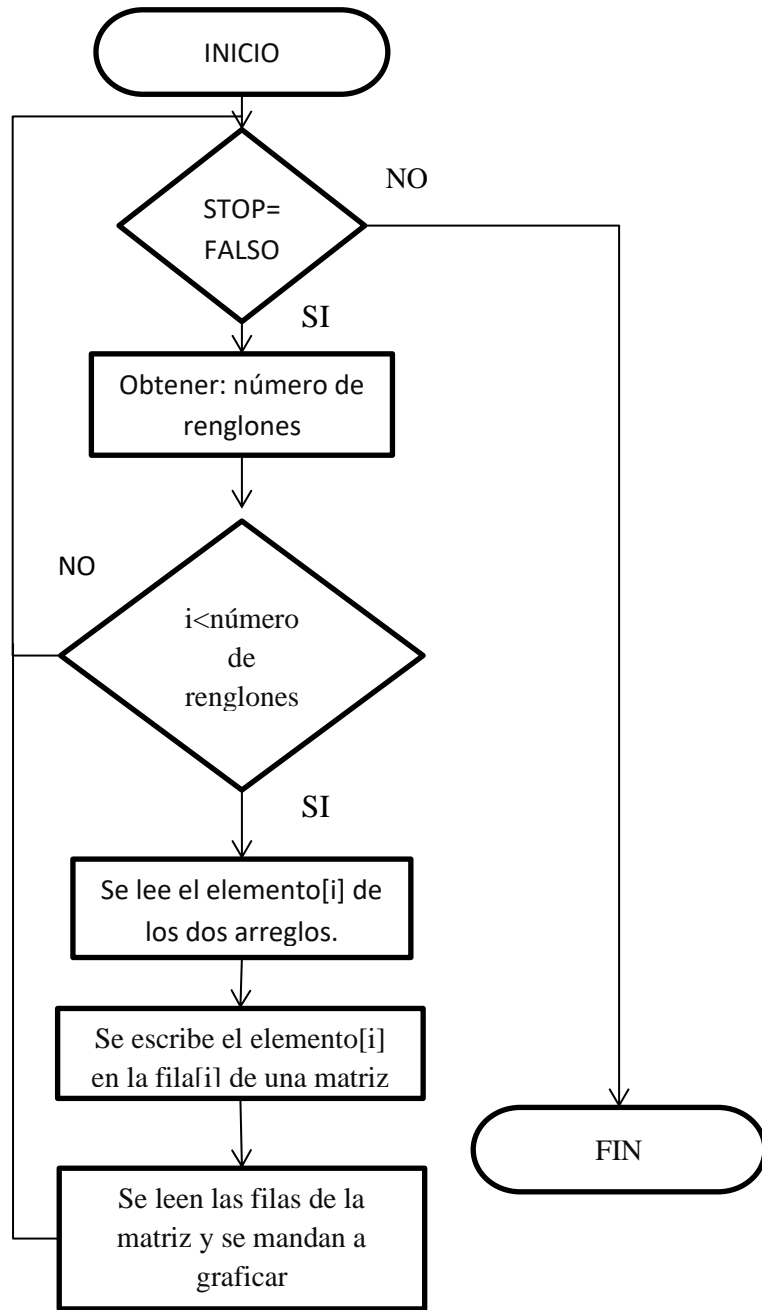


Figura 4.25: Diagrama de flujo de la gráfica que representa el patrón digital capturado.

El diagrama de flujo de la figura 4.25 describe el funcionamiento de la gráfica del patrón digital mientras el usuario edita el patrón digital y así puede observar la gráfica de cada salida y verificar si el patrón capturado es correcto.

## 4.4 Patrones digitales preestablecidos.

Dentro del software se han agregado 4 patrones digitales previamente escritos, como son alternancia de unos (AAA55), recorriendo uno de derecha a izquierda, todos unos todos cero, recorriendo el uno de izquierda a derecha.

### 4.4.1 Alternancia de unos

En este patrón digital se escribe 101010101010 (AAA) y al siguiente renglón se escribe el complemento 010101010101 (555).El patrón digital está formado por nueve renglones de alternancia y uno de todos unos, el cual se usa para indicar el termino del patrón digital. Véase figura 4.26 y 4.27.



Figura 4.26: Patrón digital capturado en los dos arreglos.

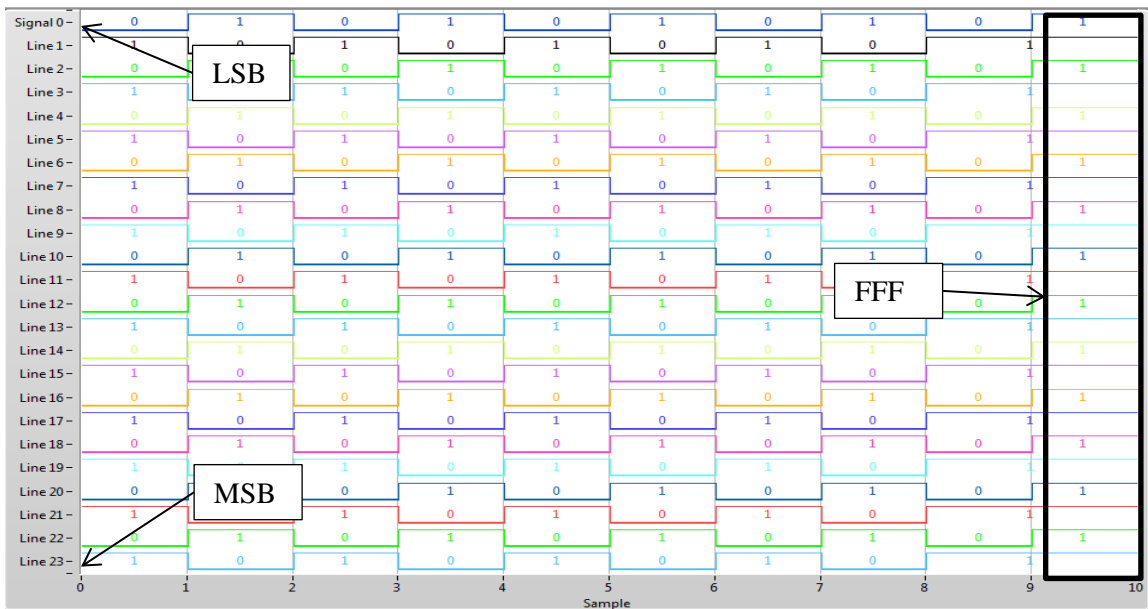


Figura 4.27: Gráfica del patrón digital AAA555.

#### 4.4.2 Recorriendo el uno de derecha a izquierda.

En este patrón el uno se escribe en el bit menos significativo y en cada renglón se va recorriendo una posición a la izquierda, en este patrón solo se ocupan doce bits, en cada interacción el uno se va recorriendo. En la figura 4.28 se muestra el patrón digital y su gráfica.

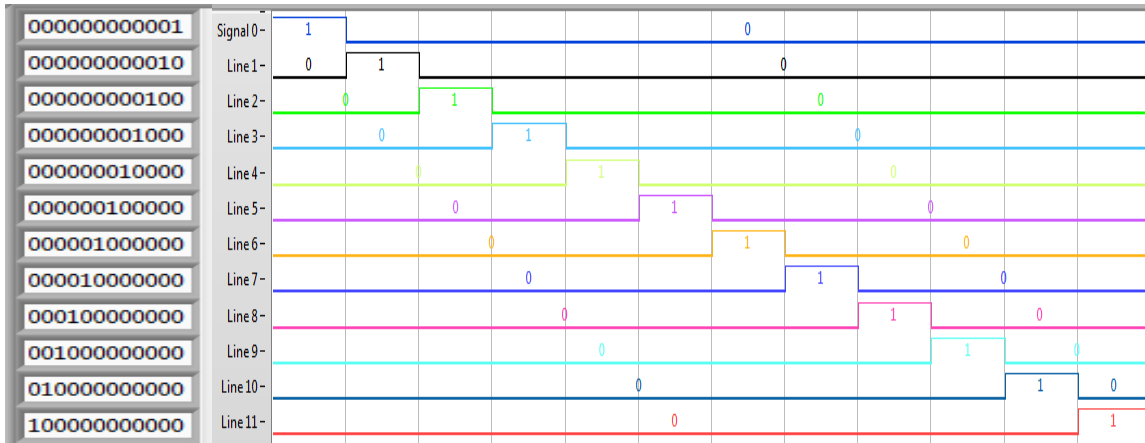


Figura 4.28: Patrón digital y gráfica para recorrer el uno de derecha a izquierda.

#### 4.4.3 Todos uno, todos cero.

En este patrón todas las salidas son cero y en el siguiente renglón todas las salidas son uno

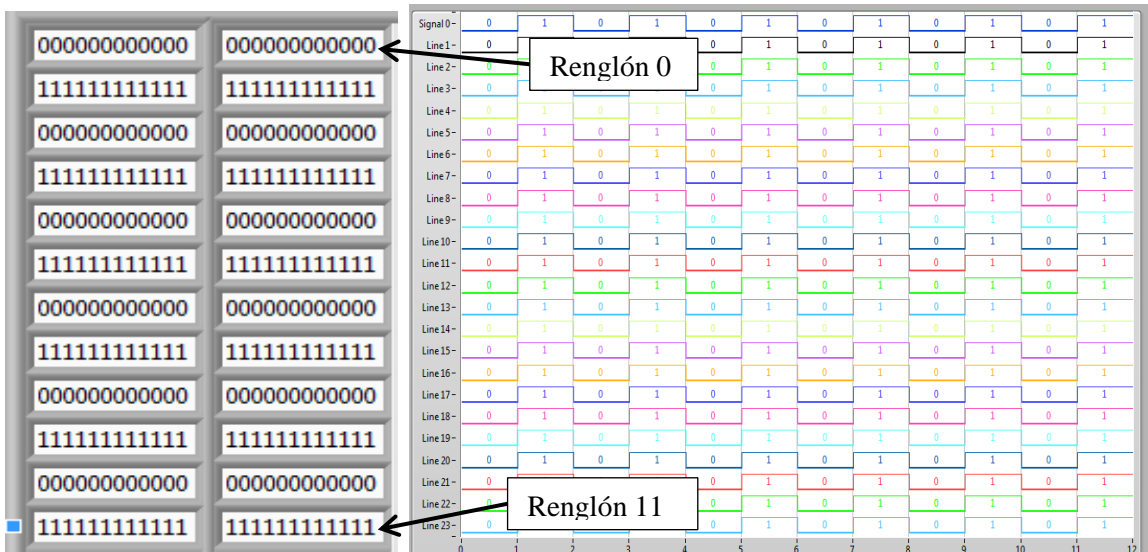


Figura 4.29: Patrón digital y gráfica donde todas las salidas alternan en uno y cero.

En la figura 4.29 se muestra el patrón digital formado por once renglones los cuales son graficados para poder visualizar el patrón digital de todos uno, todos cero.

#### 4.4.4 Recorriendo el uno de izquierda a derecha

En este patrón el uno se escribe en el bit más significativo; en este caso es el bit 11; en cada renglón se va recorriendo una posición a la derecha, hasta llegar al bit 0; en este patrón solo se ocupan doce bits. Véase figura 4.30, se muestra el patrón digital con su gráfica.

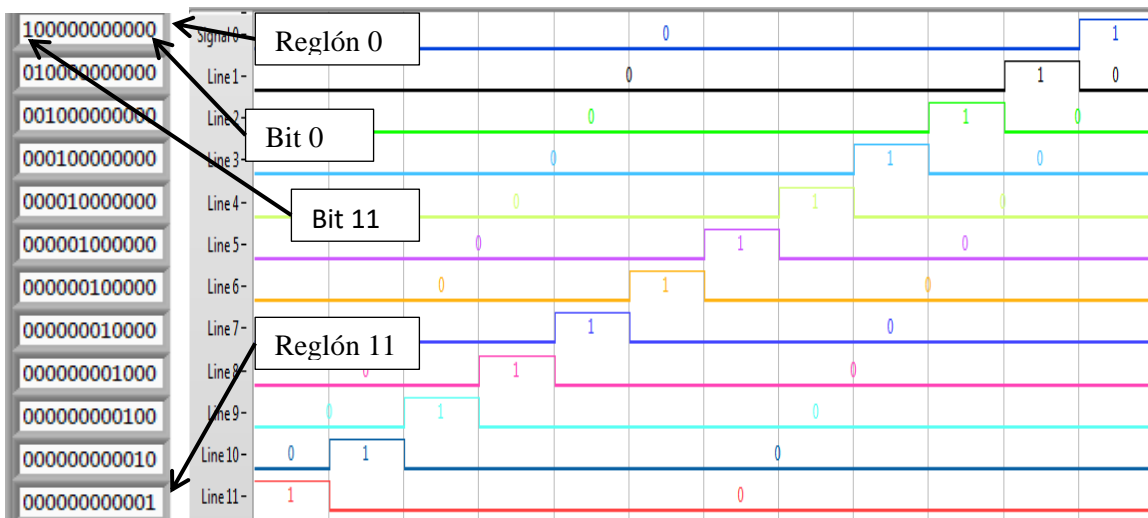


Figura 4.30: Patrón digital y gráfica donde el uno se va recorriendo desde el bit 11 hacia el bit 0.

#### 4.5 Single y cíclico, pasó de tiempo,

Los botones “single (una vez), cíclico”, de la figura 4.31, son los encargados de que la reproducción del patrón digital se realice indefinidamente o solo una vez; en la caja “paso de tiempo” se asigna el tiempo en el que debe mostrarse cada renglón del patrón digital.

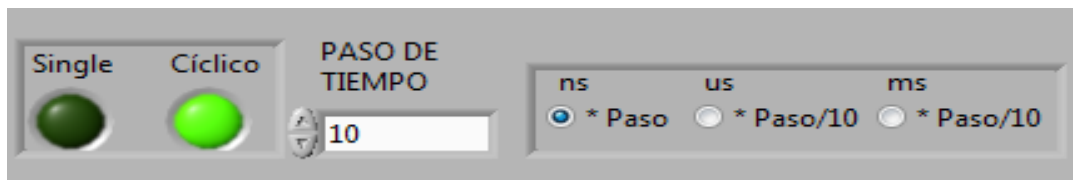


Figura 4.31: Botones single, cíclico y caja de paso de tiempo.

La figura 4.32 describe la función de los botones single y cíclico en el firmware.

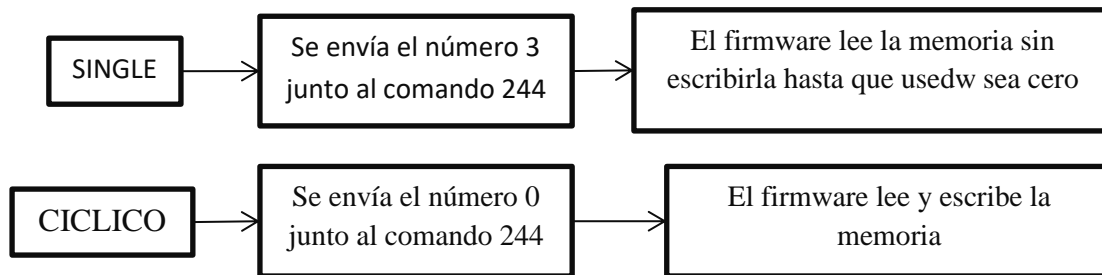


Figura 4.32: Función de los botones cíclico y single.

Para detener la reproducción o iniciarla de nuevo es necesario que el usuario cambie de la pestaña “Graficas” a la pestaña de “Configuración”, de esta manera el software envía el comando 242 con el cual se borra la memoria y se detiene la reproducción. El usuario puede elegir un paso de tiempo a su conveniencia en base al paso mínimo de 10 ns, así como también usar los radio button para seleccionar la unidad de tiempo: ns,us,ms. Con un contador de 28 bits y una unidad de tiempo de 10 ns el paso máximo que el firmware puede realizar es de:

$$t = (2^{28} - 1)10^{-9}s = 2.684s \quad (4.1)$$

La frecuencia mínima que se puede obtener es de 0.5Hz.En la gráfica de la figura 4.33 podemos observar que cada salida dura un lapso de 4 tiempos antes de cambiar de nivel.

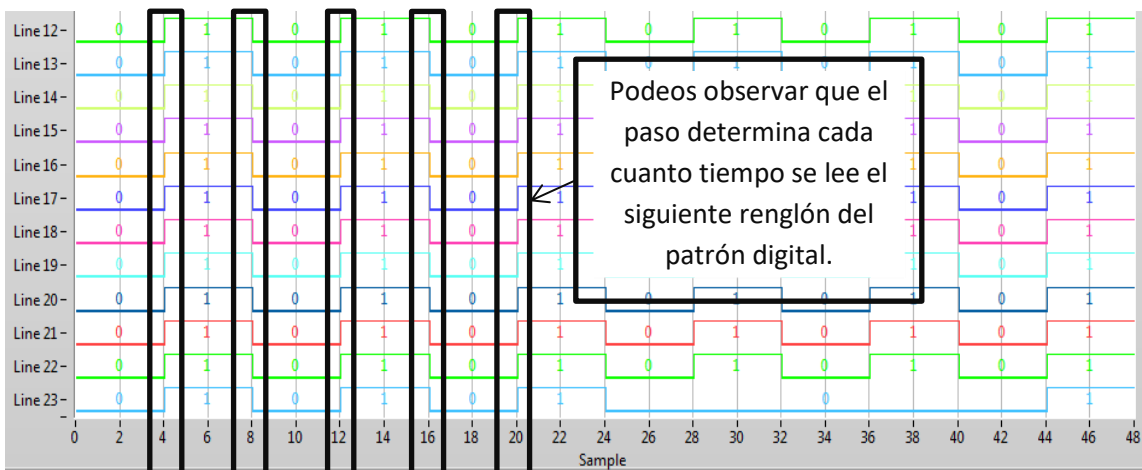


Figura 4.33: Gráfica del patrón digital todos uno, todos cero con un paso de tiempo de 4.

El valor de la caja de “paso de tiempo” describe el tiempo que se quiere mantener cada renglón del patrón digital, aumentando el tiempo de reproducción del patrón digital.

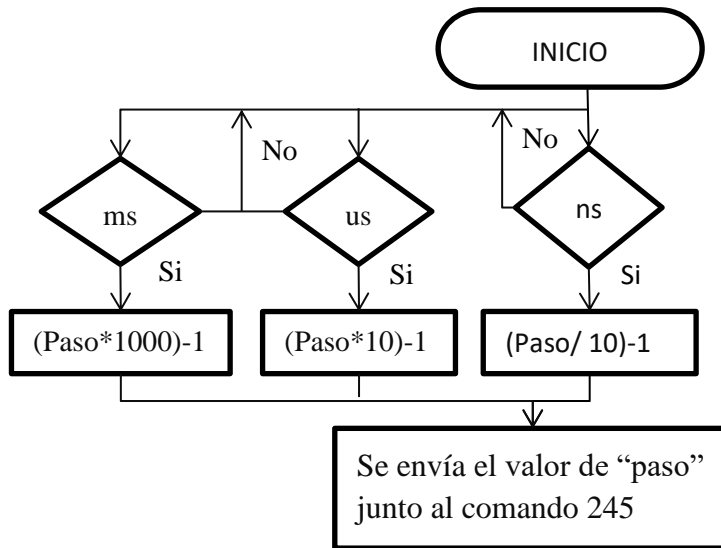


Figura 4.34: Diagrama de flujo del envío del paso de cada renglón del patrón digital.

El diagrama de flujo de la figura 4.34 explica la asignación del paso de reproducción del patrón digital

#### 4.6 Gráfica y envío de datos

Cada vez que se pasa a la pestaña GRAFICAS, el software envía el comando 242 (F2) junto con DATO=0 para detener la reproducción de la memoria y borrar la memoria, así la memoria FIFO estará lista para recibir el nuevo patrón digital, después de borrar la memoria se envía el comando 241 (F1) junto con el dato del patrón digital, al terminar de enviar todos los datos del patrón digital envía el paso asignado por el usuario con el comando 245(F5) antes de cerrar el puerto se envía el comando 244 (F4) junto con DATO= número de renglones del arreglo del patrón digital. Esta secuencia se explica en la figura 4.35.

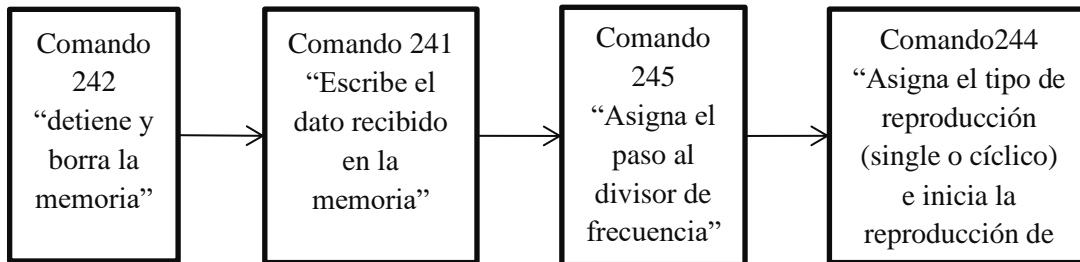


Figura 4.35: Secuencia de comandos enviados al FPGA desde el software del generador de patrones digitales.

La secuencia del envío de datos y configuración de la reproducción del generador de patrones digitales se muestra en la figura 4.36.

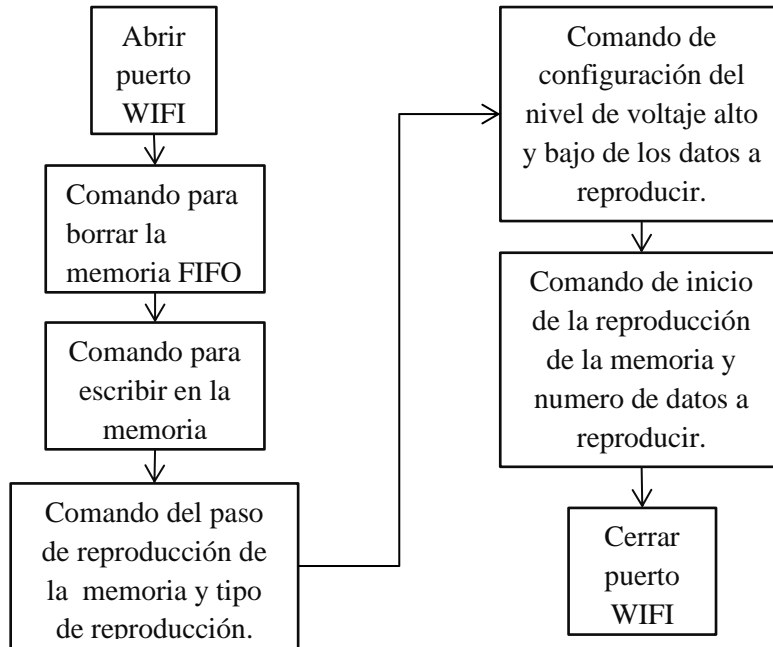


Figura 4.36: Secuencia del envío de datos al firmware del generador de patrones digitales.



Figura 4.37: Tercera Pantalla de la interfaz gráfica sin patrón digital.

En la figura 4.37, se muestra el área de gráfica del generador de patrones digitales en el cual el usuario será capaz de visualizar los datos que se enviaron al FPGA, en formato hexadecimal o en binario; en este último es posible observar los cambios de nivel de cada salida; observando los niveles lógicos del patrón digital. Esto facilita que el usuario pueda encontrar posibles errores en el patrón digital previamente capturado y realizar su posterior corrección.

Cada vez que se envía la orden de reproducir, junto con esta se envía el tipo de reproducción, la cual puede ser “cíclica o única”. Cuando la reproducción es cíclica, el patrón digital se reproduce indefinidamente una y otra vez hasta que el usuario presione el botón stop o regrese a la pestaña de configuración. Si la opción de reproducción es única, entonces el patrón digital es reproducido una sola vez y aunque el usuario siga en la pestaña de gráficas, el patrón digital no se vuelve a reproducir hasta que el usuario regrese a la pestaña de configuración y de ahí pase de nuevo a la de gráficas.

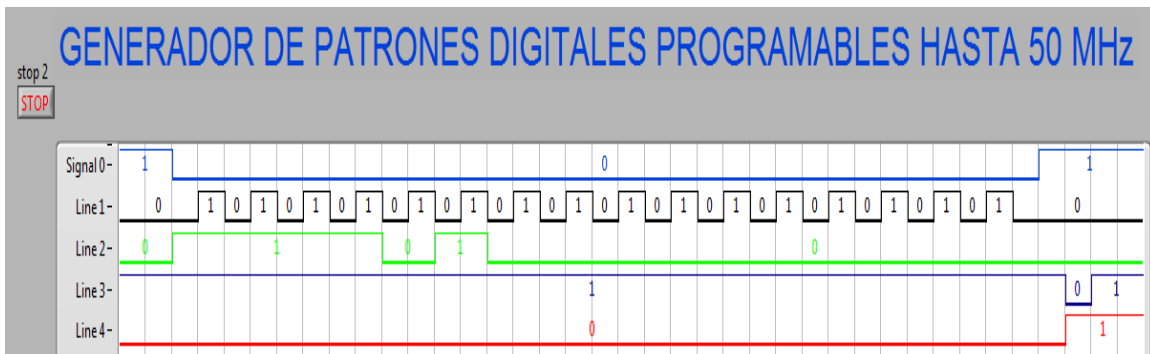


Figura 4.38: Gráfica de un patrón digital de 5 líneas de salida.

La figura 4.38 muestra la gráfica de los datos enviados de un patrón digital de un protocolo de comunicación SPI. Para la transmisión de datos por WiFi y la creación de la gráfica del patrón digital el software sigue la siguiente lógica:

1. Se lee el elemento [i] del arreglo del patrón digital.
2. Los 12 bits del elemento[i] son colocados en el renglón [i] de una matriz de 16 columnas por el número de renglones del arreglo.
3. Para extraer solo los 12 bits del arreglo se reduce la matriz a través del módulo obtener submatriz.
4. La matriz obtenida de 12 columnas por el número de renglones del arreglo del patrón digital.
5. Cada renglón de la segunda matriz es enviado a graficar.
6. Posterior mente el dato graficado se envía al FPGA por el puerto WIFI.
7. Se reinicia el proceso hasta que se lean todos los renglones de la matriz los cuales coinciden con el número de datos del arreglo del patrón digital capturado por el usuario.

En la figura 4.39 se muestra como se crea la gráfica del patrón digital y el envío de los datos al FPGA.

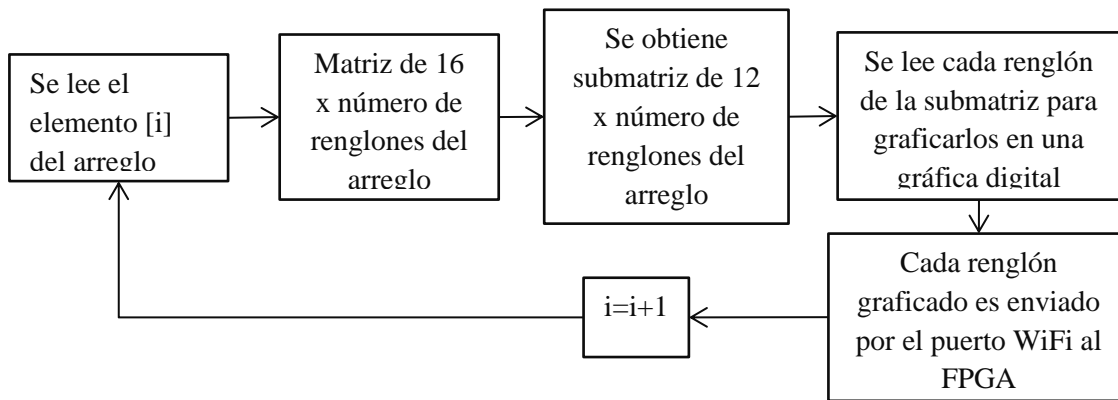


Figura 4.39: Secuencia de la creación de la gráfica y envío de los datos por WiFi.

Una vez terminado de enviar todos los datos del patrón digital, el software queda a la espera de que el usuario cambie de pestaña o presione el botón “parar” lo cual mandaría el comando de parar y borrar la memoria. De esta manera el hardware queda preparado para recibir otro patrón digital y no contiene datos no deseados del patrón digital anterior. Es importante hacer notar que cada renglón de la gráfica del patrón digital representa una columna del arreglo del patrón digital; ver figura 4.40. Es por ello que en cada arreglo se tienen 12 bits los cuales representan en la gráfica 12 renglones de cada canal, haciendo un total de 24 renglones (bits).

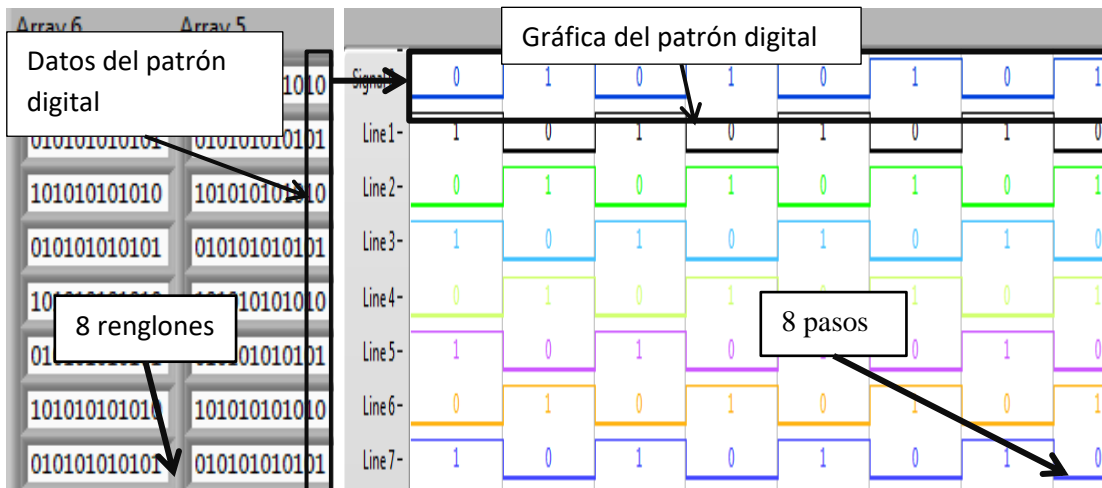


Figura 4.40: Explicación de cómo se grafican los datos del patrón digital.

Podemos observar que la primera columna de derecha a izquierda representa el patrón digital del bit menos significativo, es decir el estado de cada salida del patrón digital depende de la secuencia de cada columna del arreglo de datos. Otra manera de explicarlo, es que cada paso de la gráfica es un renglón del arreglo de datos, si se tienen 8 renglones entonces se grafican 8 pasos en la gráfica.

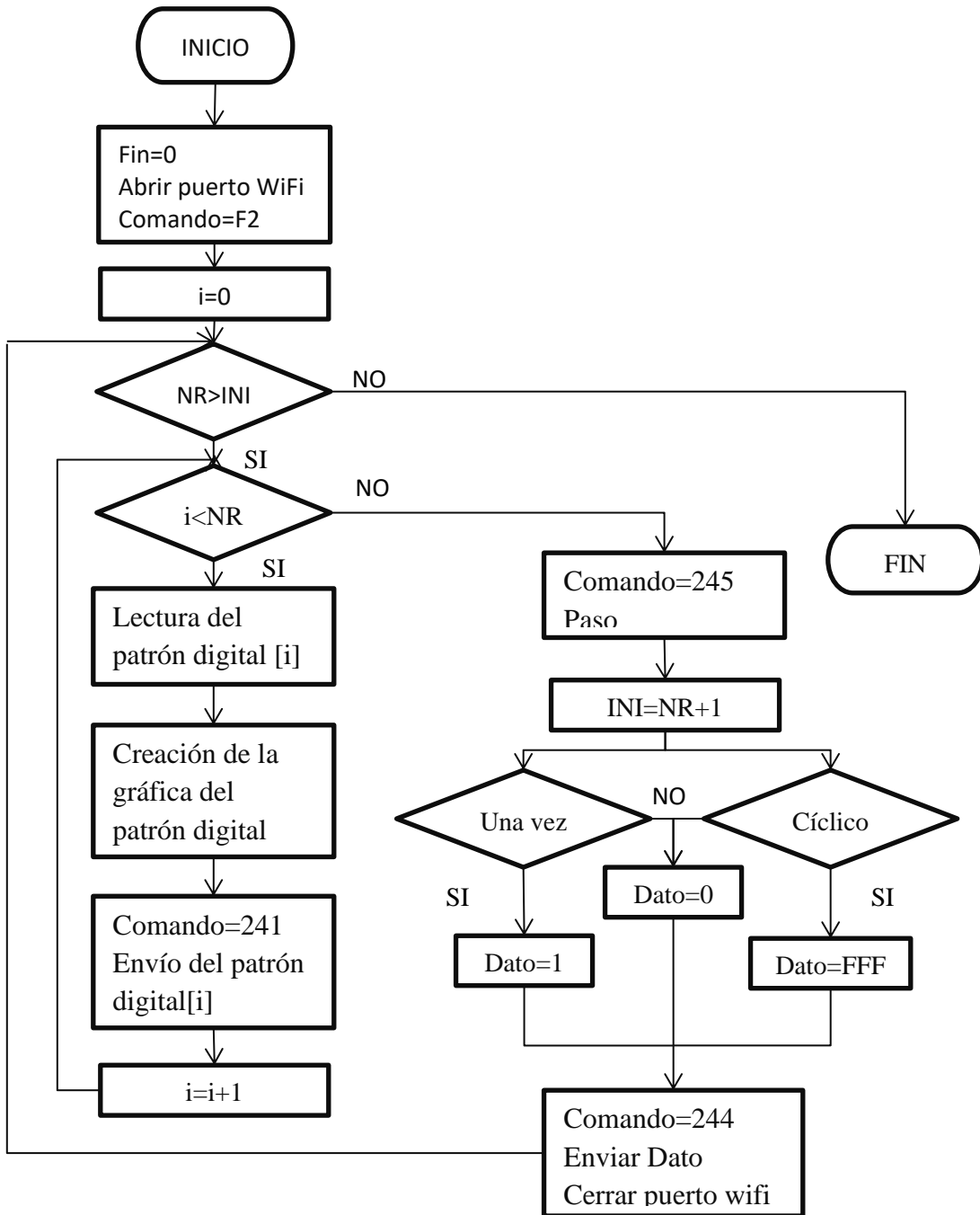


Figura 4.41: Diagrama de flujo para la lectura del arreglo del patrón digital y gráfica del patrón digital en Labview.

El diagrama de flujo de la figura 4.41 explica la secuencia el procedimiento para leer el patrón capturado, graficarlo y enviarlo por el puerto WiFi al FPGA.

## 4.7 Conclusiones

La tendencia actual de la tecnología, es la de desarrollar sistemas embebidos minimalistas, que sean capaces de enviar la información a sistemas más grandes, donde esta sea procesada e enviada de regreso a los pequeños sistemas embebidos. Lo cual disminuye el costo del producto, el tiempo de diseño y fabricación. Podríamos decir que casi todo el mundo tiene acceso a una computadora, por lo tanto un sistema basado en una PC es muy versátil, pues solo se preocupa de capturar la información, enviarla a la PC para su procesamiento y mostrar el resultado obtenido por la PC.

El uso de herramientas virtuales disminuye el tiempo de desarrollo del dispositivo, en consecuencia el tiempo y costo del desarrollo del mismo, haciendo más accesible la tecnología desarrollada con este sistema; garantizando que investigadores y alumnos tengan un fácil acceso a herramienta de laboratorio que les ayude a desarrollar sus investigaciones, proyectos o prácticas de sistemas digitales.

## CAPITULO 5 Resultados experimentales

En este capítulo se presentan los resultados experimentales para 12 de los 24 bits disponibles sin la opción de nivel de voltaje. Debido al número de pines necesarios para cambiar el nivel de voltaje de cada salida, solo se realiza este acondicionamiento de señal para una sola salida de las 24 disponibles. Se realiza la reproducción de diferentes patrones digitales, previamente editados y guardados con el software del generador de patrones digitales.

La intención es demostrar que hardware, firmware y software; trabajan juntos para crear el patrón digital que el usuario capturo en el software, ya sea para caracterizar dispositivos o implementar protocolos de comunicación digital.

### 5.1 Resultados visualizados en el analizador lógico Agilent

Para esta prueba se conectaron 12 bits de los 24 bits disponibles en el puerto asignado en el FPGA. Para realizar la prueba primero se capturo el patrón digital, posteriormente se envió al FPGA para su reproducción y muestreo en el analizador lógico de la empresa Agilent.

#### 5.1.1 Patrón digital AAA555

Para este patrón digital se capturan los datos que se muestran en el arreglo de la figura 5.1 a):

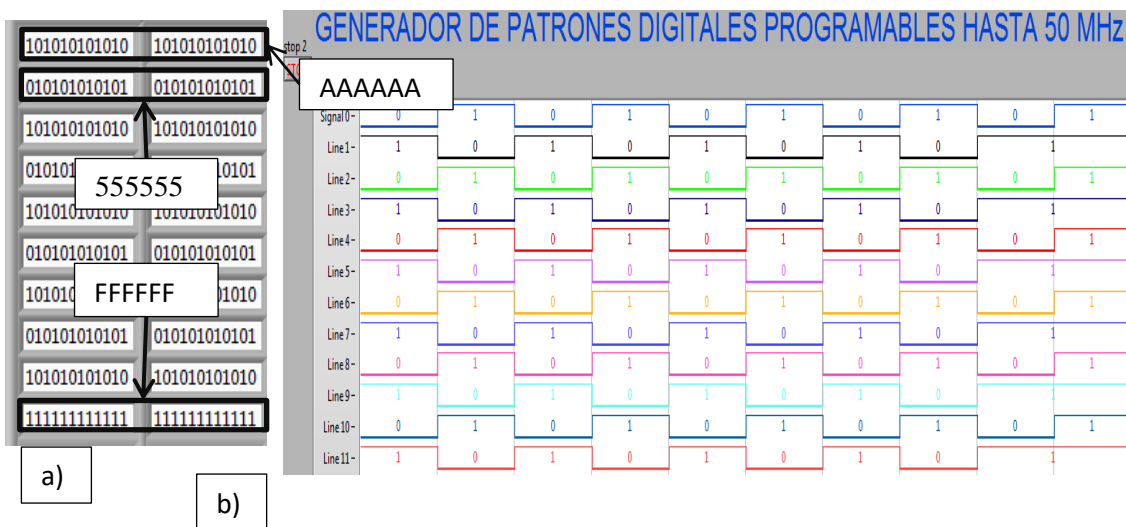


Figura 5.1: a) Patrón digital AAA555, b) Gráfica del patrón en el software del generador de patrones digitales.

En este patrón digital, figura 5.1 a), se observa la alternancia de unos en cada salida del generador, solo hasta el final todas las salidas son unos para indicar el fin del patrón digital. La configuración del paso para este patrón digital es de 10ns, el cual representa el tiempo mínimo o la velocidad máxima de lectura de reproducción para cada renglón del patrón digital.

La gráfica de la figura 5.2 se obtuvo del analizador lógico de la marca Agilent, en donde se visualiza el cambio de salidas hasta que todas son uno.

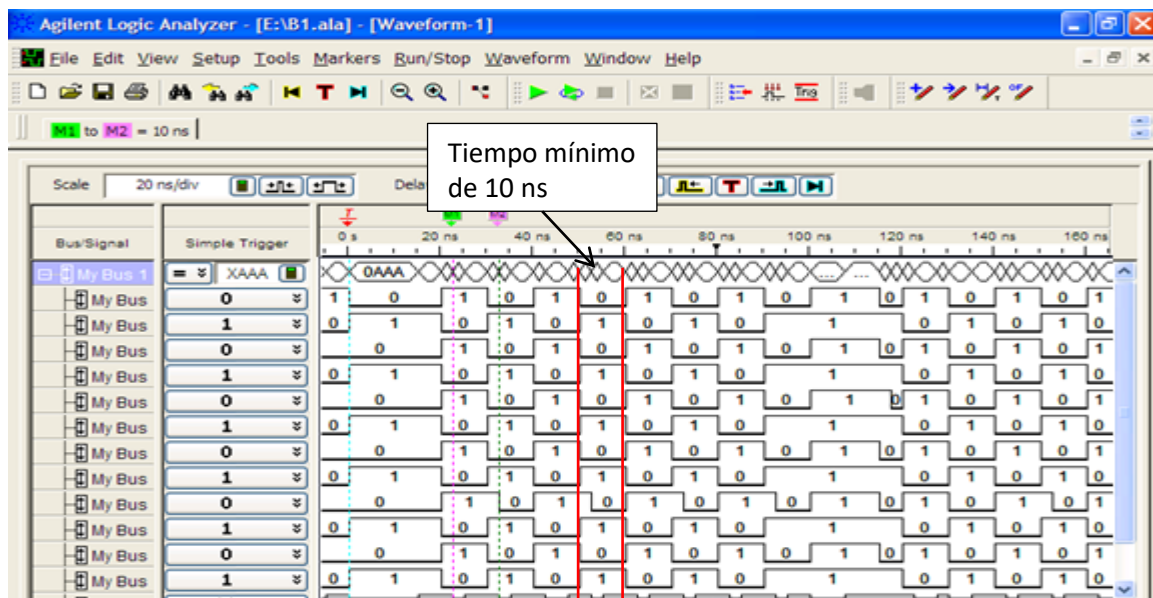


Figura 5.2: Gráfica del patrón digital AAA55 en analizador lógico Agilent.

### 5.1.1 Patrón digital recorriendo 1 Derecha a Izquierda

Su nombre deriva del patrón que escribe el "1" para caminar de derecha a izquierda, implica escribir un valor binario "000000000001" en la primera ubicación del arreglo. En la siguiente ubicación el valor se desplaza una posición hacia la izquierda para que el valor binario sea "000000000010". Cuando el "1" se ha desplazado los 12 bits, el valor "000000000001" se reutiliza y el desplazamiento a la izquierda comienza de nuevo desde allí. Este patrón de desplazamiento hacia la izquierda en un bit luego de escribir el nuevo valor en la siguiente ubicación de memoria se repite o no dependiendo del tipo de reproducción cíclico o single; en el caso de cíclico la reproducción se detiene hasta que el usuario cambie de la pestaña de "Gráfica" a la pestaña de "Configuración" o presione el botón de stop.

Como aplicaciones de este patrón digital tenemos: identificación de cortos en las líneas de transmisión y crosstalk (interferencia entre canales)

Para este patrón digital se capturan los siguientes datos que se muestran en el arreglo de la figura 5.3 a):

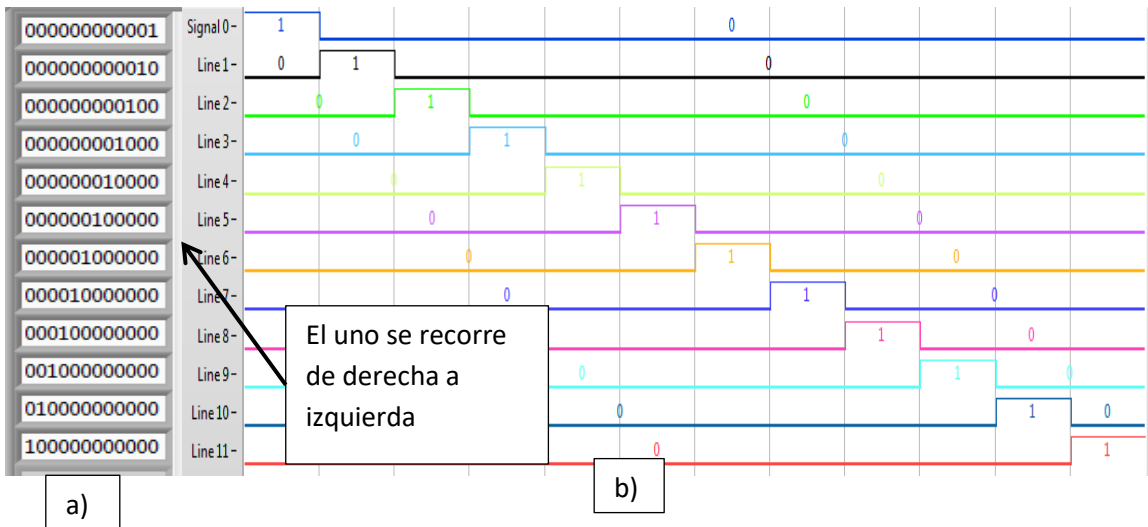


Figura 5.3: a) Datos, b) gráfica del patrón digital recorriendo uno D-I.

En la gráfica de la figura 5.4 podemos ver como el uno se va recorriendo de derecha a izquierda, es decir del bit menos significativo al más significativo, con un paso de 50ns.

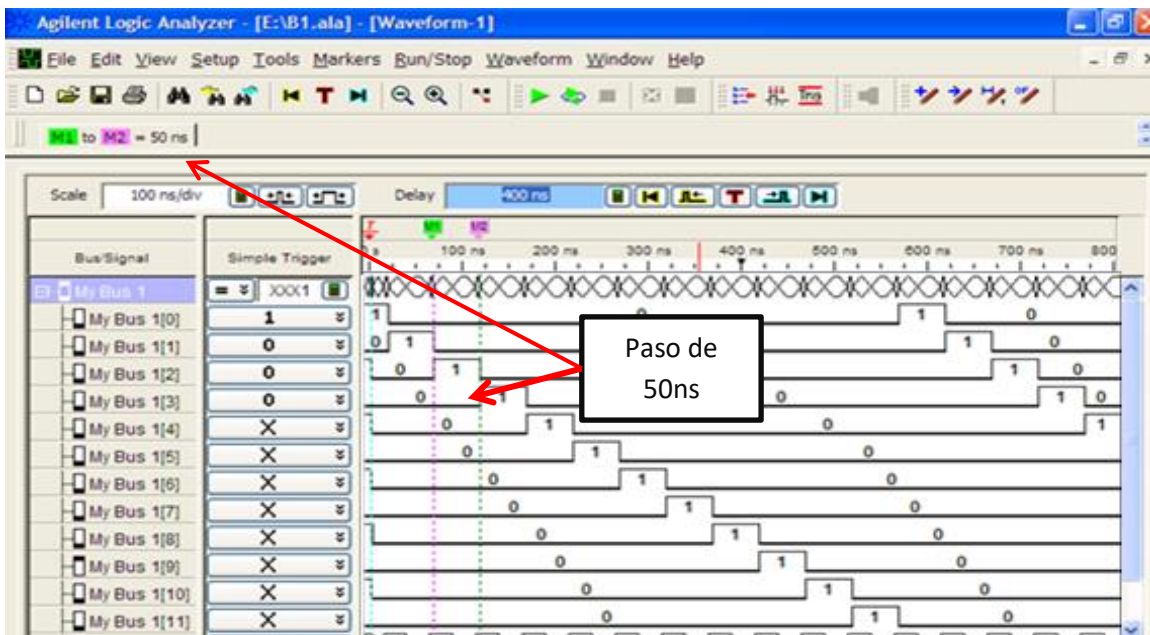


Figura 5.4: Gráfica del patrón digital recorrer uno D-I en analizador lógico Agilent.

### 5.1.2 Patrón digital recorriendo 1 Izquierda a Derecha

Semejante al patrón anterior se escribe el "1" para caminar de izquierda a derecha, implica escribir un valor binario "100000000000" en la primera ubicación del arreglo. En la siguiente ubicación el valor se desplaza una posición hacia la derecha para que el valor binario sea "010000000000". Cuando el "1" se ha desplazado los 12 bits, el valor "100000000000" se reutiliza y el desplazamiento a la derecha comienza de nuevo desde allí. Este patrón de desplazamiento hacia la derecha en un bit luego de escribir el nuevo valor en la siguiente ubicación de memoria se repite o no dependiendo del tipo de reproducción cíclico o single; en el caso de cíclico la reproducción se detiene hasta que el usuario cambie de la pestaña de "Grafica" a la pestaña de "Configuración" o presione el botón de stop.

Como aplicaciones de este patrón digital tenemos: identificación de cortos en las líneas de transmisión y crosstalk (interferencia entre canales). Para este patrón digital se capturan los siguientes datos de la figura 5.5 a):

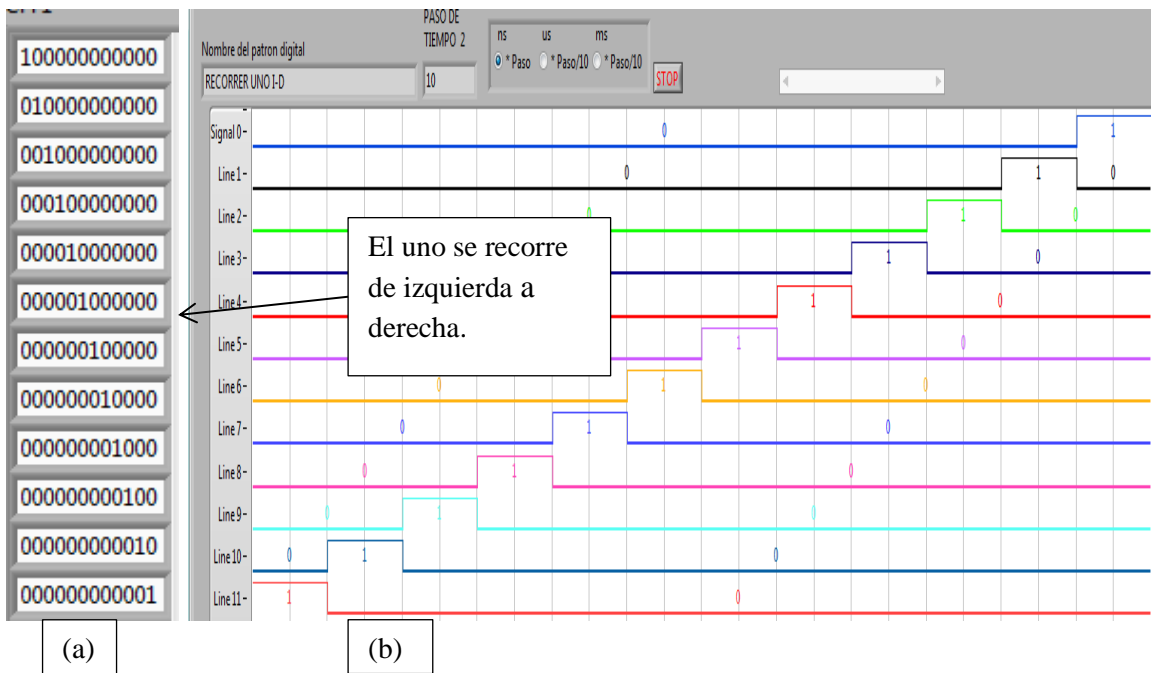


Figura 5.5: (a) Datos, (b) gráfica del patrón digital recorriendo 1 I-D.

La gráfica de la figura 5.6 muestra el patrón digital recorriendo el 1 de I-D, visualizado en el analizador lógico de Agilent con un paso de reproducción de 50 ns.

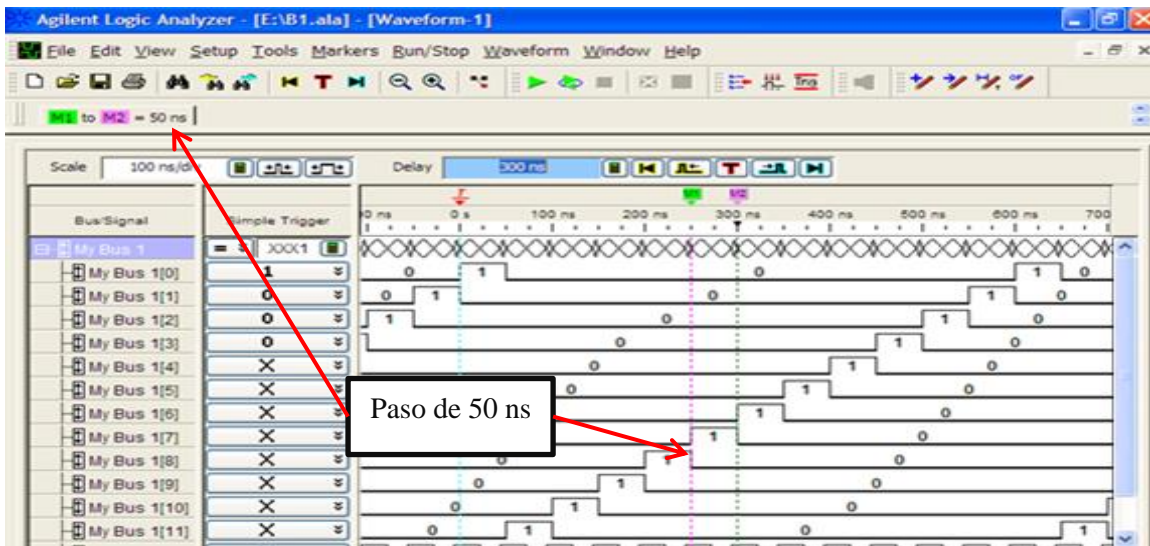


Figura 5.6: Gráfica del patrón digital recorrer uno D-I visualizado en el analizador lógico de Agilent.

### 5.1.2 Patrón digital todo uno, todo cero.

El patrón escribe “1” en todas sus salidas y en el siguiente paso, todas son “0”, este tipo de patrón digital ayuda a verificar el envío de datos en un bus digital. Los datos capturados don los de la figura 5.7 (a).

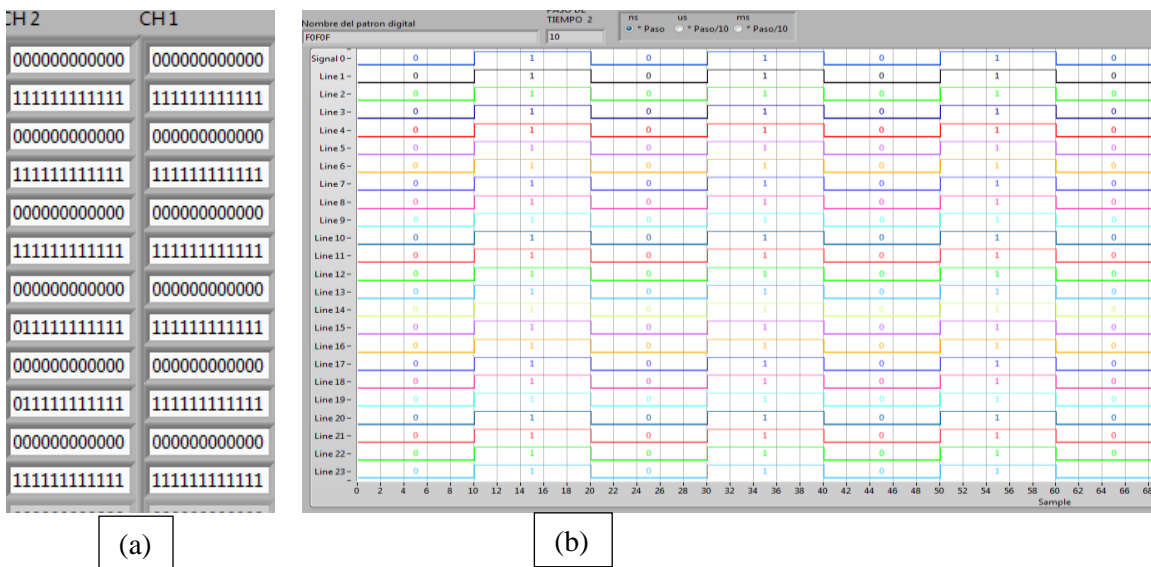


Figura 5.7: (a) Datos, (b) gráfica del patrón digital todos uno, todos cero.

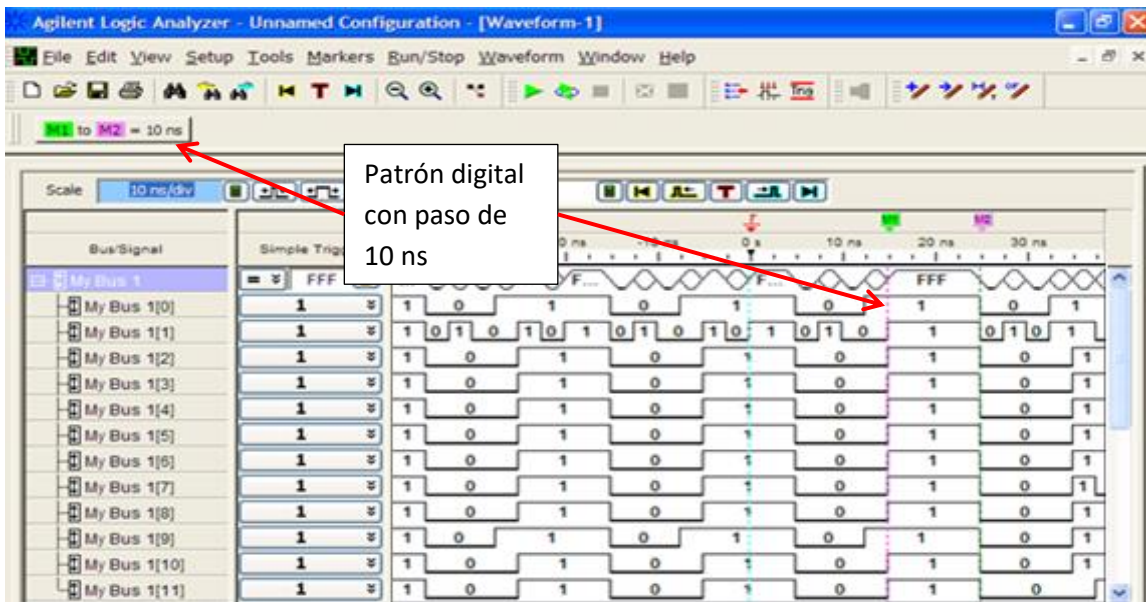


Figura 5.8: Gráfica del patrón digital todos uno, todos cero, capturado en el analizador lógico de la marca Agilent.

La gráfica de la figura 5.8 muestra el patrón digital de todos uno, todos cero con un paso de reproducción de 10 ns.

#### 5.1.4 Gráficas a diferentes pasos de tiempo.

En las siguientes gráficas se muestra la variación del paso de tiempo, con el patrón digital todo uno, todo cero a diferentes pasos de tiempo: 100 ns, 500 ns, 1  $\mu$ s, 1 ms y 2 ms. Con lo cual podemos variar el tiempo de reproducción. El usuario puede elegir un paso de tiempo a su conveniencia en base al paso mínimo de 10 ns, así como también usar los radio button para seleccionar la unidad de tiempo: ns,  $\mu$ s, ms. Ver figura 116. Con un contador de 28 bits y una unidad de tiempo de 10 ns el paso máximo, es decir la frecuencia mínima que el firmware puede realizar es de:

$$t = (2^{28} - 1)10^{-9}s = 2.684s \quad (5.1)$$

Para la primera gráfica de la figura 5.9, se colocó un valor de 100 ns al paso de tiempo, podemos visualizar que el analizador lógico muestra un cambio de nivel lógico cada 100 ns y debido a que la opción de cíclico estaba seleccionada el patrón digital se repite indefinidamente hasta que el usuario cambie a la pestaña de configuración.

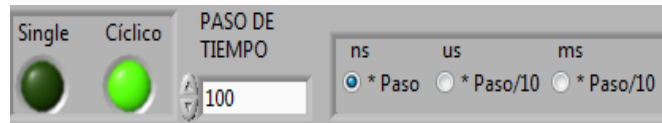


Figura 5.9: Configuración del tipo de reproducción y paso del patrón digital.

El software del analizador lógico cuenta con dos cursores con los cuales se puede medir la diferencia de tiempo según la posición de estos. El cursor uno es M1 y el cursor dos es M2 y el tiempo medido que existe entre ellos es la diferencia de M1 menos M2.

$$\Delta t = M1 - M2 \quad (5.2)$$

Con lo cual es posible mostrar la duración de cada renglón del patrón digital.

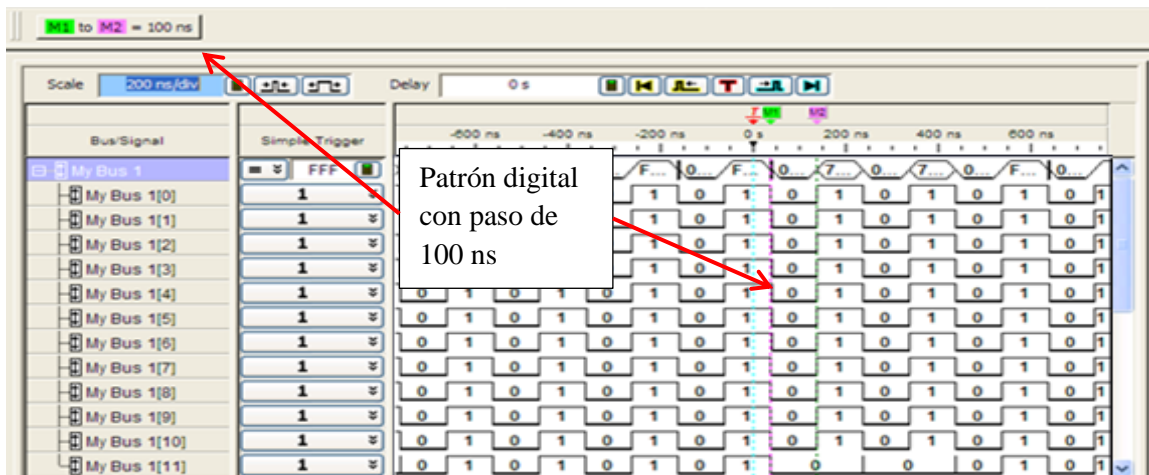


Figura 5.10: Gráfica del patrón todo uno, todo cero con un paso de 100ns.

En la figura 5.10 se observa que la diferencia de tiempo entre M1 y M2 es de 100 ns, demostrando que cumple con la configuración de paso de tiempo de 100 ns, asignada en el software del generador de patrones digitales, figura 5.10.

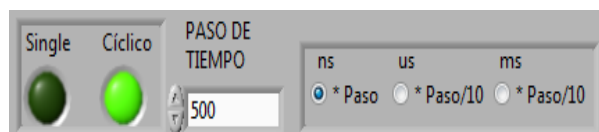


Figura 5.11: Configuración del paso a 500 ns.

Para la siguiente gráfica del analizador lógico, el paso de tiempo del generador de patrones digitales se configura a 500 ns, ver figura 5.11; en la figura 5.12 podemos observar que la duración del nivel alto del pulso es de 500 ns, cumpliendo con la configuración del paso de tiempo editada en la interfaz de usuario.

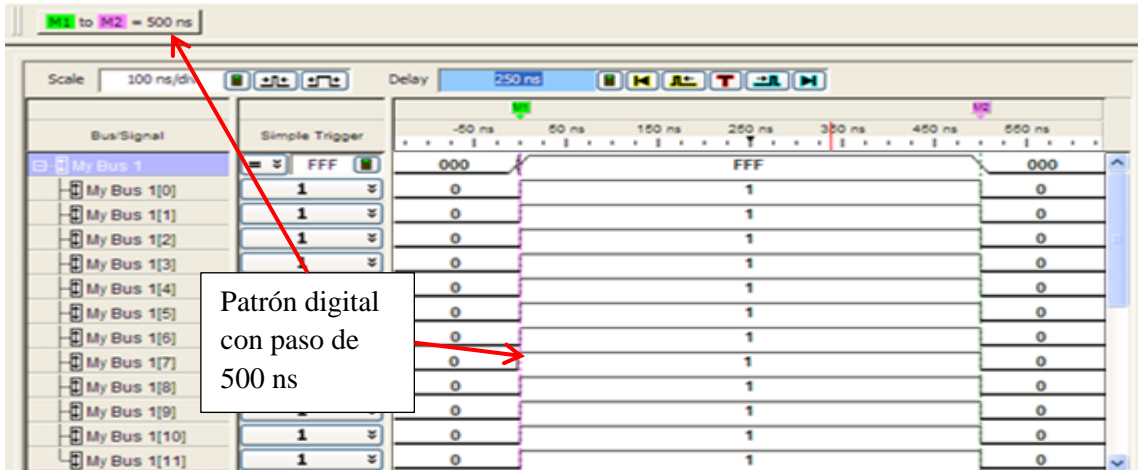


Figura 5.12: Patrón digital todo uno, todo cero con un paso de 500 ns

La duración del nivel alto del pulso de la figura 5.12 parece no concordar con la escala de tiempo de la figura 5.10 y es debido a que la escala de tiempo de la figura 5.10 es de 200 ns/div y en la figura 5.12 la escala es de 100 ns/div; en la figura 5.12 se muestra que el curso M1 está colocado en el cero de la barra de tiempo del analizador lógico y M2 está colocado en 500 ns.

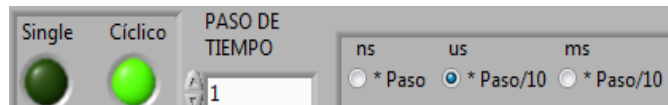


Figura 5.13: Paso de tiempo configurado a 1  $\mu$ s.

En la figura 5.13 se muestra la configuración en la interfaz de usuario del paso de reproducción a 1  $\mu$ s. El analizador lógico muestra que el tiempo del nivel alto y bajo del patrón digital todos uno, todos cero dura 1  $\mu$ s tal como se configura en el software del generador de patrones digitales; en este caso los 12 renglones del patrón digital hacen que cada 12  $\mu$ s se repita el patrón digital.

La gráfica de la figura 5.14 muestra el patrón digital con un paso de reproducción de 1  $\mu$ s.

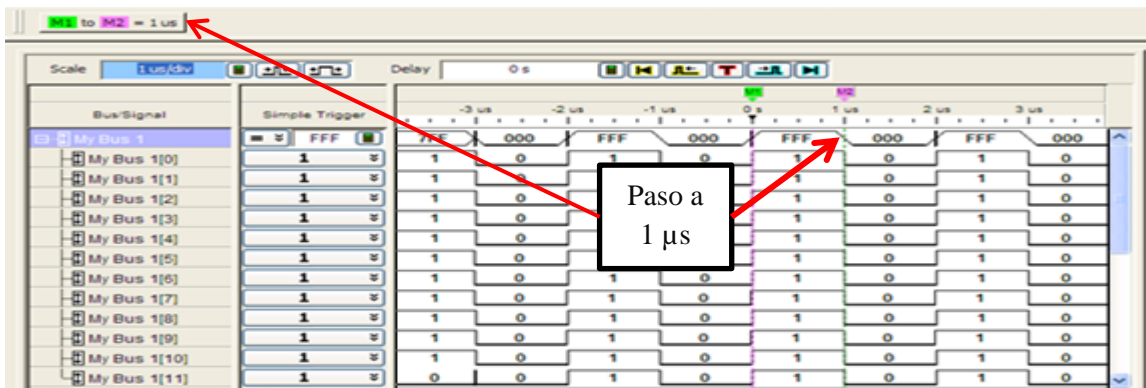


Figura 5.14: Patrón digital con un paso de 1  $\mu$ s.

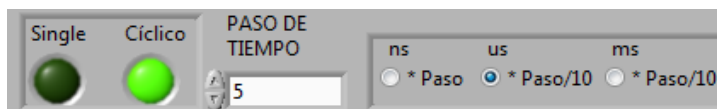


Figura 5.15: Paso de tiempo configurado a 5  $\mu$ s.

Para la siguiente gráfica de la figura 5.16 el paso de tiempo se configuró a 5  $\mu$ s, como se muestra en la figura 5.15; por lo tanto el nivel alto y bajo tienen una duración de 10  $\mu$ s

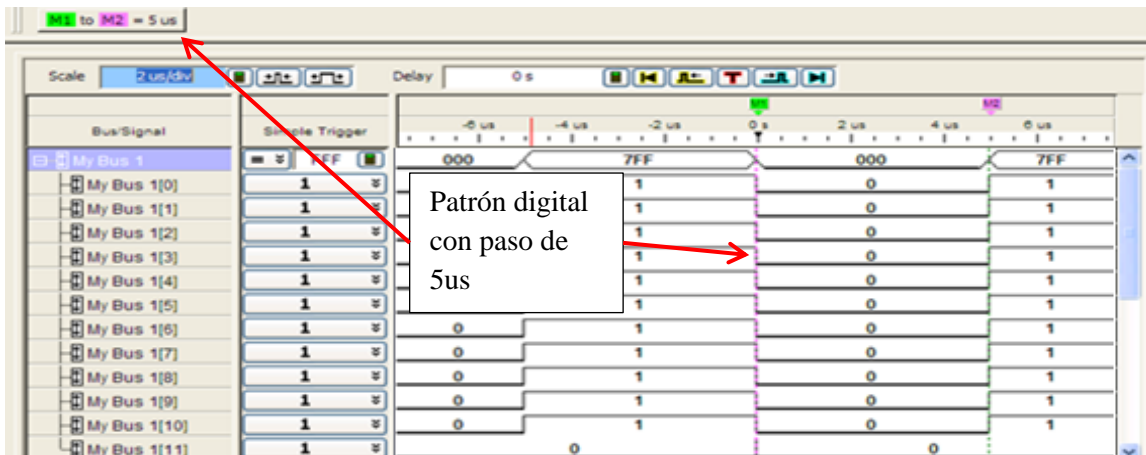


Figura 5.16: Patrón digital con un paso de 5  $\mu$ s.

En la figura 5.17 se visualiza la configuración del paso de reproducción a 1 ms, el patrón digital es visualizado en el analizador lógico, gráfica de la figura 5.18 y se demuestra que el firmware cumple con la configuración del paso de reproducción a 1 ms.

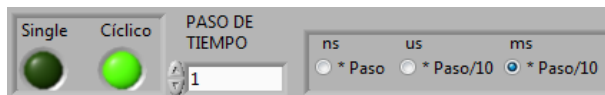


Figura 5.17: Configuración del paso de tiempo a 1 ms.

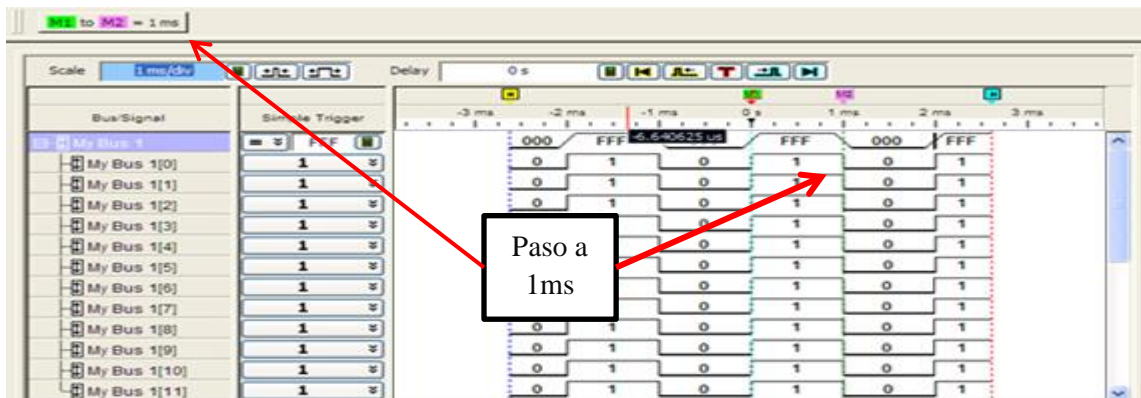


Figura 5.18: Patrón digital con un paso de 1 ms.

Las figuras 5.19 y 5.20 muestran la configuración del paso de reproducción a 2 ms y la visualización del patrón digital en el analizador lógico.

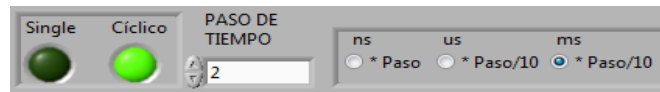


Figura 5.19: Configuración del paso de tiempo a 2 ms.

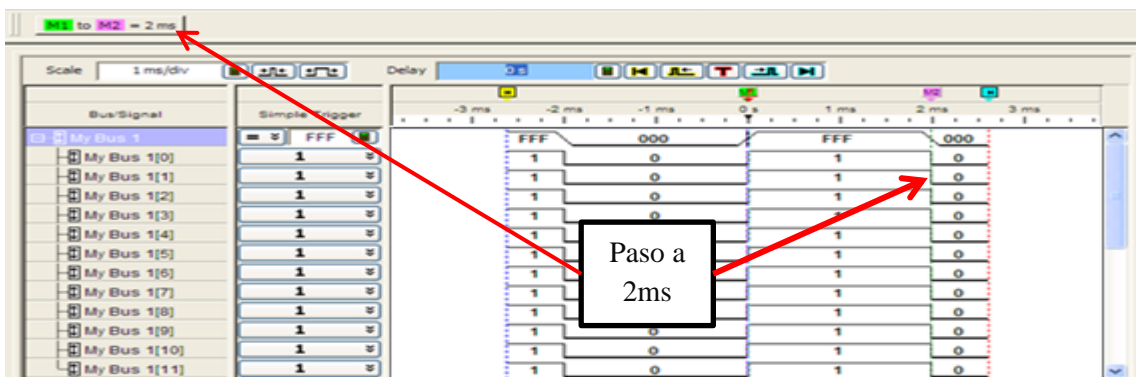


Figura 5.20: Patrón digital con un paso de 2ms.

## 5.2 Resultados de la tarjeta acopladora de voltaje, visualizados en el osciloscopio

Con la integración de software, firmware y hardware del generador de patrones digitales, se realizaron las siguientes pruebas. La intención es la de verificar el paso de reproducción, el funcionamiento de la tarjeta acopladora de voltaje, observar si se alcanzan los voltajes para las tecnologías digitales mencionadas y los niveles de voltaje editados por el usuario.

Esta tarjeta cuenta con un convertidor digital analógico THS5651A de 125 MPS, dos amplificadores OPA 659; estos amplificadores tienen la característica de tener un ancho de banda de 650 MHz con una ganancia unitaria, una fuente simétrica de  $\pm 5 V$ ; para la atenuación del ruido eléctrico presente en el ambiente, se le colocó un plano de tierra, al diseño del circuito impreso y se agregó un filtro pasa bajas.

### 5.2.1 Gráficas de la salida para acoplamiento con la tecnología ECL.

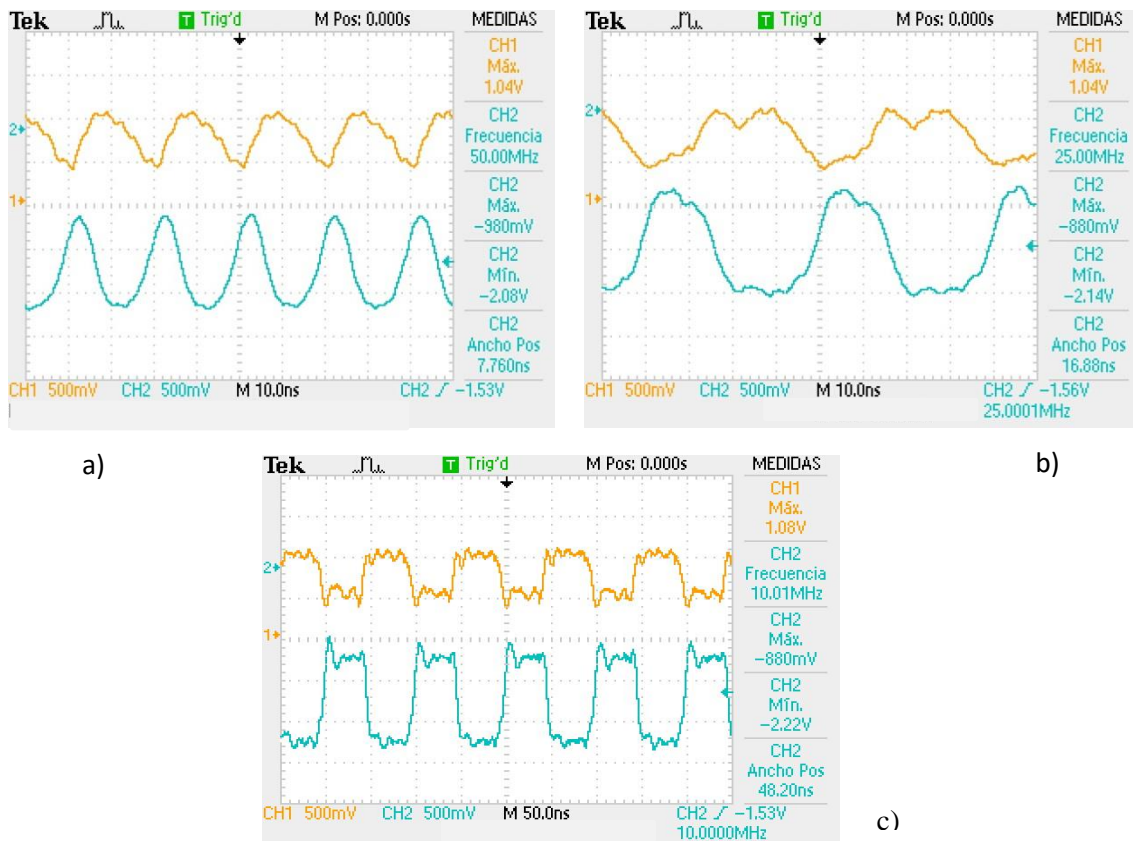


Figura 5.21: Gráfica de la tecnología ECL. -0.8: 1 Lógico, -1.8: 0 Lógico. a) 50 MHz, b) 25 MHz, c) 10 MHz.

En la figura 5.21 se muestran los niveles de voltaje para la tecnología ECL, aunque la señal tiene unos picos en los niveles alto y bajo, la amplitud de la señal se mantiene dentro de las tolerancias de la señal ECL.

La frecuencia de la señal "a)" es de 50 MHz y debido a que el ancho de banda del osciloscopio es de 200 MHz, no pasan todos los armónicos de la señal y la señal se ve senoidal.

En la figura 5.21 b) se muestran los mismos niveles de voltaje a una frecuencia de 25 MHz, es por ello que se muestran más componentes de la señal pero aun no son los suficientes para que la señal se muestre cuadrada.

En la figura 5.21 c) se muestran los mismos niveles de voltaje a una frecuencia de 10 MHz; se observa que la señal ya casi es cuadrada pues ya pasan más armónicos para formar la señal cuadrada.

A esta familia lógica se le conoce como lógica de acoplamiento por emisor (ECL), la cual opera en base al principio de conmutación de corriente; ver figura 5.22; en el cual una corriente de polarización fija menor que  $I_c$  (sat) se cambia del colector de un transistor a otro. Debido a esta operación en modo de corriente, a esta forma lógica también se le conoce como lógica en modo de corriente (CML) [8] [9] [28].

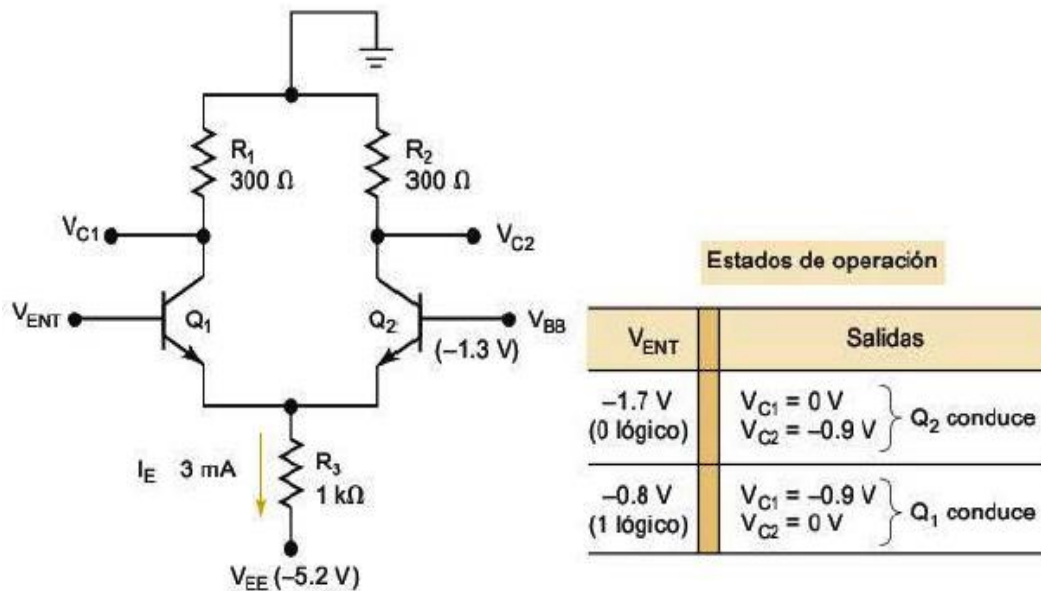


Figura 5.22: Circuito ECL básico.

En otras palabras, esta corriente cambia entre el colector de Q1 y el colector de Q2 a medida que VENT cambia entre sus dos niveles lógicos de -1.7 V (0 lógico para ECL) y -0.8 V (1 lógico para ECL).

## 5.2.2 Gráficas de la salida para acoplamiento con la tecnología PECL

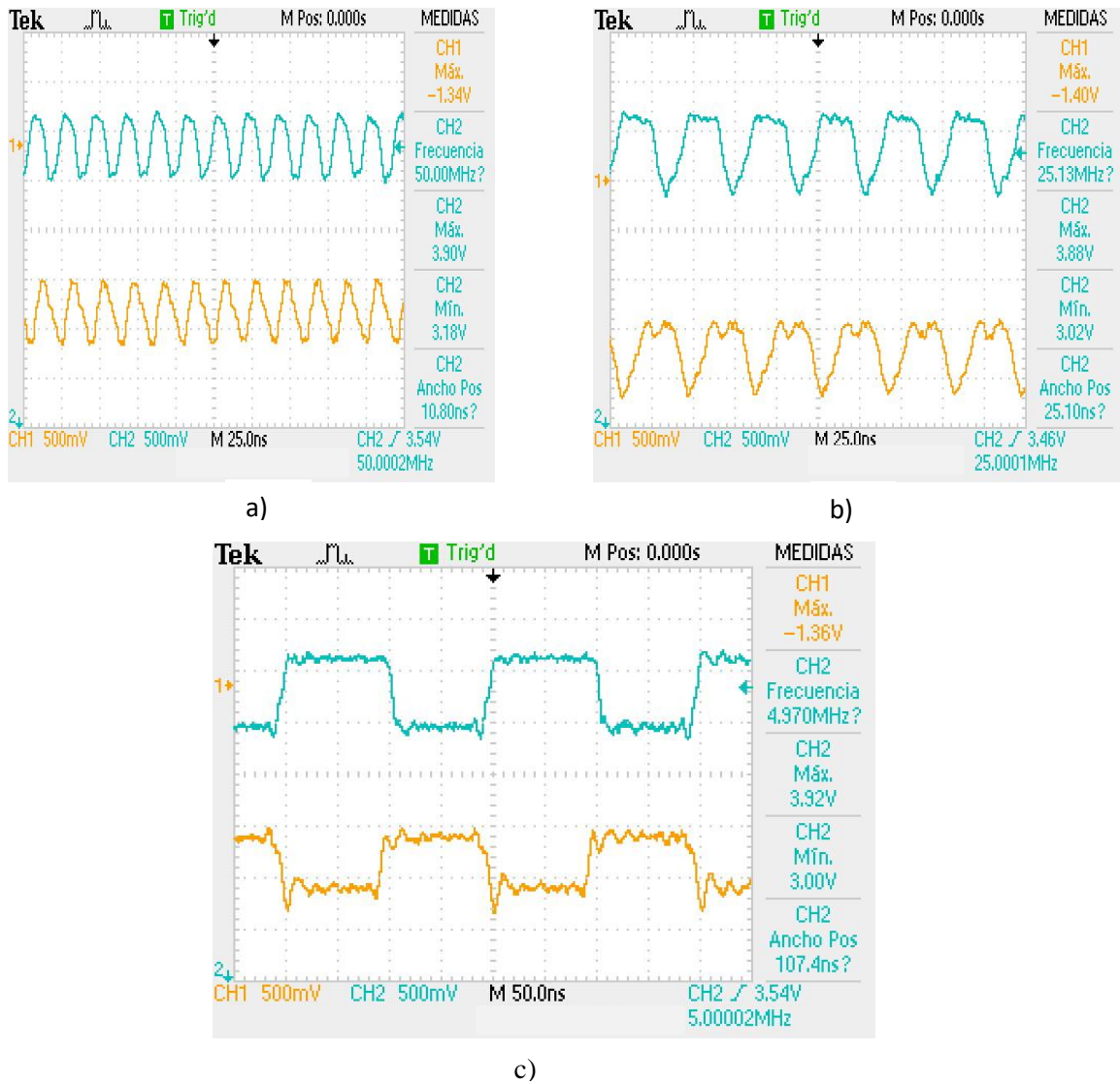
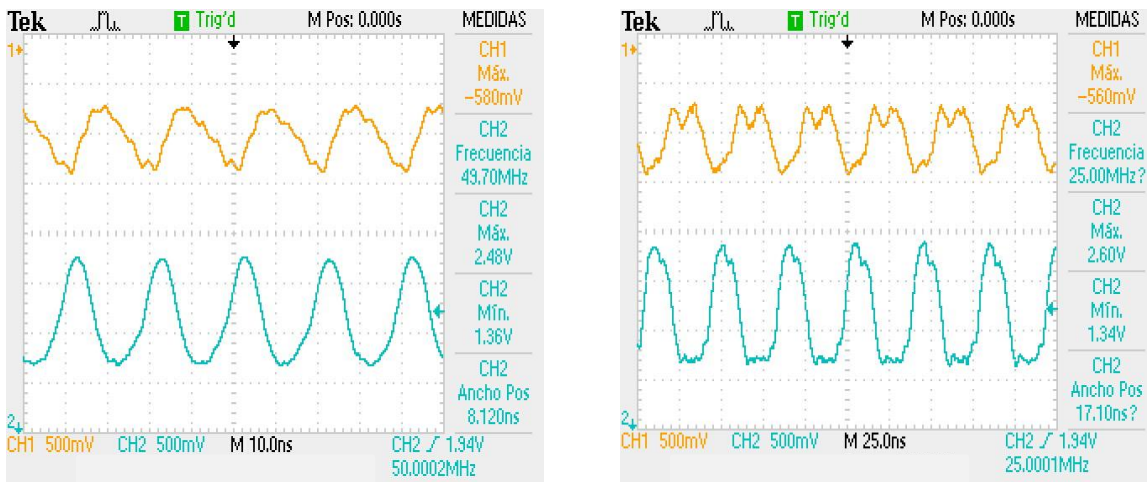


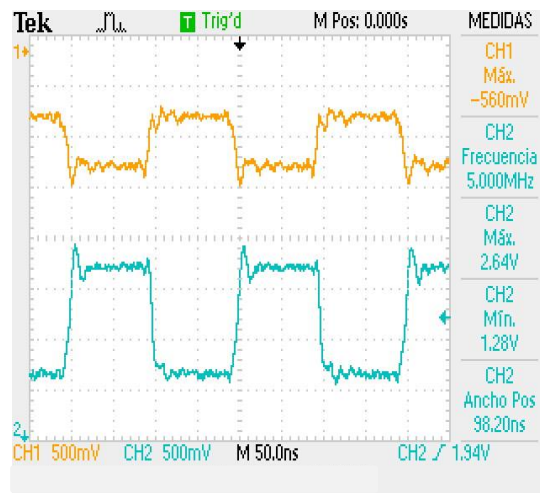
Figura 5.23: Gráfica de la señal PECL a) 50 MHz, b) 25 MHz, c) 5 MHz.

Para la tecnología PECL se muestra la Figura 5.23 en estas gráficas se visualiza un nivel bajo de 3.2 V para el 0 y de 4 V para el 1, estos niveles están dentro de los parámetros de interpretación del 0 y 1 lógicos de la tecnología PECL. Cabe destacar que la señal tiene una frecuencia de 50 MHz y debido al ancho de banda del osciloscopio la señal no se aprecia como una señal cuadrada; mas sin embargo al disminuir la frecuencia la señal empieza a verse cuadrada.



a)

b)



c)

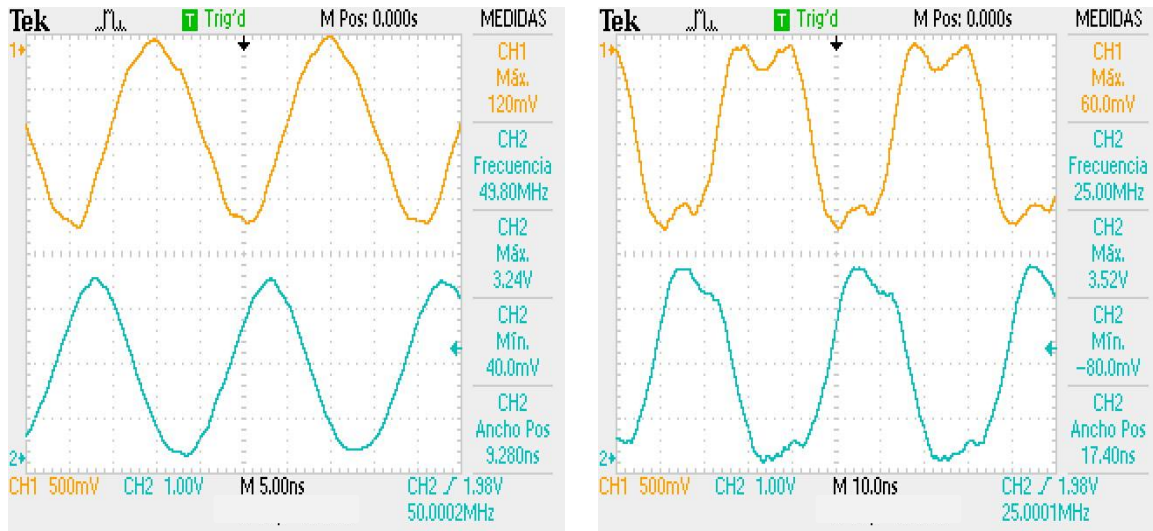
Figura 5.24: Gráfica de la señal LVPECL a) 50 MHz, b) 25 MHz, c) 5 MHz.

En la figura 5.24 la señal PECL es  $3.3\text{ V}-0.8\text{ V}=2.5\text{ V}$  para el 1 y  $3.3\text{ V}-1.8\text{ V}=1.5\text{ V}$  para el 0 lógico. Esta tecnología Digital es conocida como LVPECL [26]. A continuación se muestran las gráficas de la tecnología LVPECL a la frecuencia de 25 MHz y 1 MHz.

### 5.2.3 Gráficas de la salida para acoplamiento con la tecnología CMOS

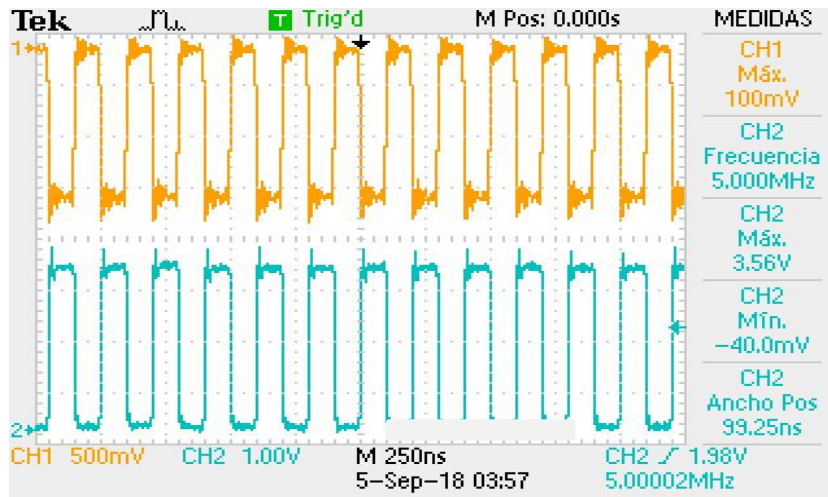
A continuación se muestran las gráficas de la señal de salida de la tarjeta acopladora de voltaje, para niveles de voltaje de la tecnología CMOS. En la figura 5.25 a) se aprecia una señal con niveles de voltaje de 0 V a 3.3 V a una frecuencia de 50 MHz, la señal naranja es la señal de la salida del primer amplificador en configuración de inversor, la señal azul es la salida del siguiente operacional

configurado como inversor en donde se obtiene la salida de la señal con los niveles de la tecnología CMOS.



a)

b)



c)

Figura 5.25: Gráfica de una señal CMOS a) 50 MHz, b) 25MHz, c) 5MHz.

En la figura 5.25 b) se muestra la misma señal pero con una frecuencia e 25 MHz y en la figura 132 c) se muestra a una frecuencia de 5 MHz; en esta última gráfica de la señal CMOS se puede observar que la señal es cuadrada, debido a que las componentes de la señal son las suficientes para que esta se vea cuadrada.

## 5.2.4 Gráficas de la salida para acoplamiento con la tecnología TTL

A continuación se muestran las gráficas de una señal TTL a diferentes frecuencias.

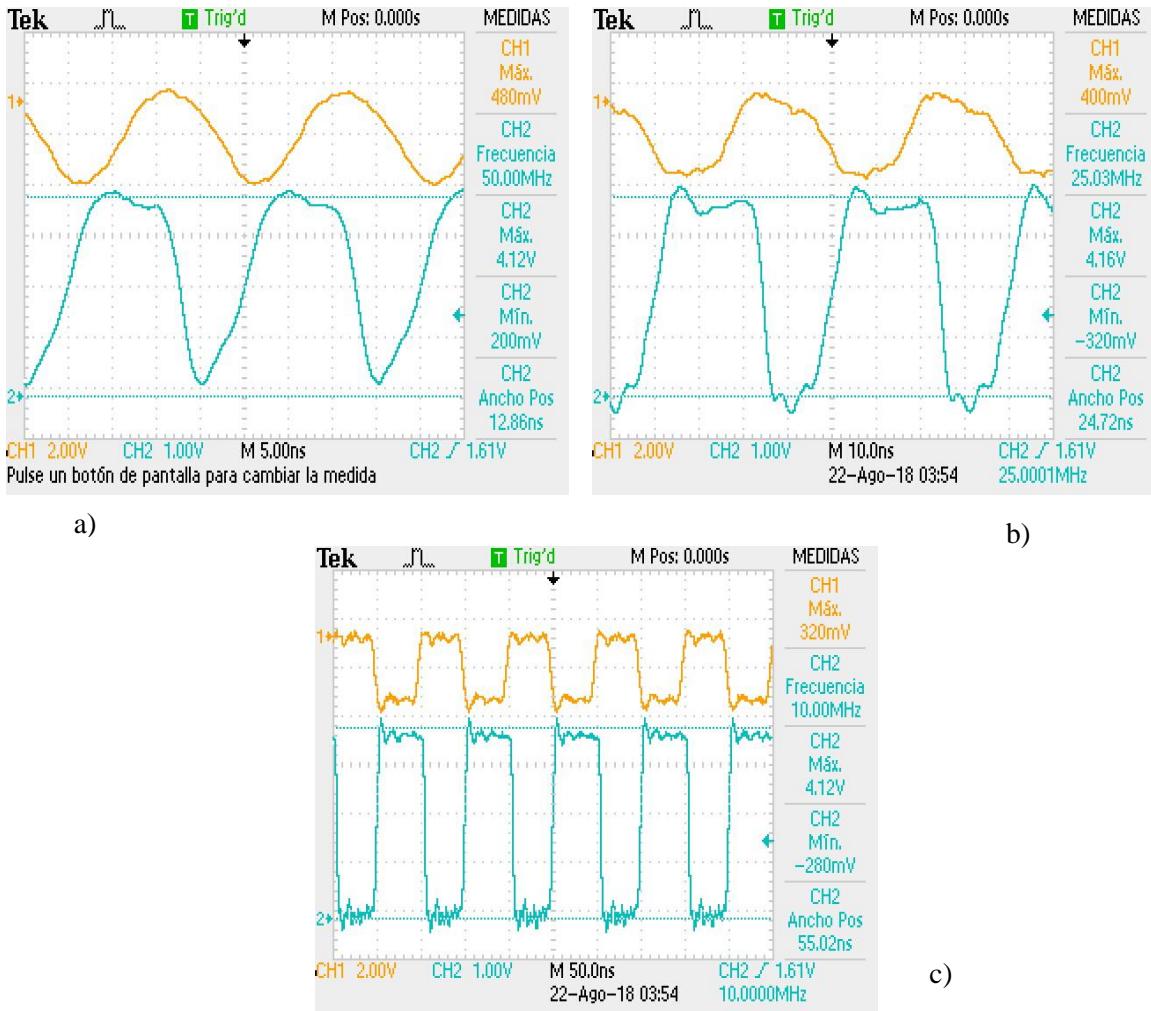


Figura 5.26: Gráfica de una señal TTL a) 50 MHz, b) 25 MHz, c) 10 MHz.

En la gráfica de la figura 5.26 a) se observa una señal TTL a una frecuencia de 50 MHz, el valor de voltaje de los niveles lógicos de esta señal están dentro de las tolerancias de la señal TTL. Los niveles de voltaje lógicos de esta señal son 0 lógico=0.2 V y el 1 lógico= 4.1 V.

Esto se debe a que el voltaje de alimentación de los operaciones OPA659 es de  $\pm 5 V$  y el voltaje máximo de salida de los operacionales es de 4.2 V y debido a que los voltajes de entrada para los niveles lógicos de la tecnología TTL son los siguientes:

$$V_{IH(MIN)} = 2 V$$

$$V_{IL(MAX)} = 0.8 V$$

La señal de salida de la tarjeta acopladora de voltaje configurada para los niveles TTL cumple con los niveles de voltaje para ser interpretada como una señal TTL y pueda ser aplicada a dispositivos que trabajen con esta tecnología digital. Señal en color anaranjado a la salida del primer operacional. Señal en color Azul a la salida del segundo operacional.

### 5.3 Aplicaciones del generador de patrones digitales.

A continuación se mencionan algunas de las aplicaciones del generador de patrones digitales, para ello se realiza la configuración por protocolo I2C de un potenciómetro digital y un DAC por protocolo SPI.

#### 5.3.1 Protocolo SPI

Se muestra el protocolo de comunicación del DAC MCP4921; figura 5.27; el cual trabaja por protocolo SPI, figura 5.28; la frecuencia máxima para la señal de reloj es de 20 MHz. Cuenta con una resolución de 12 bits, y una entrada para la habilitación de los datos a convertir. En este DAC la señal de reloj es usada para sincronizar cargar los cuatro bits de configuración y los doce bits de datos.

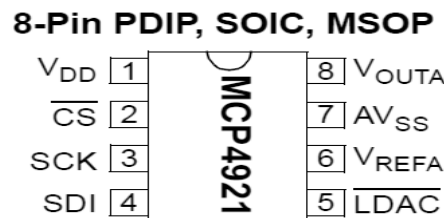


Figura 5.27: Descripción de los pines del DAC MCP4921.

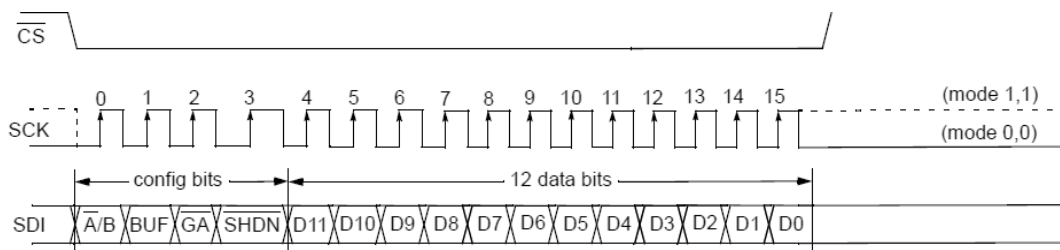


Figura 5.28: Diagrama de tiempos del DAC MCP4921.

Se capturo el protocolo de comunicación con el valor a la salida del DAC en la interfaz de usuario del generador de patrones digitales el cual se ingresan los valores del diagrama de tiempos para configurar el DAC, esto se muestra en la siguiente figura:

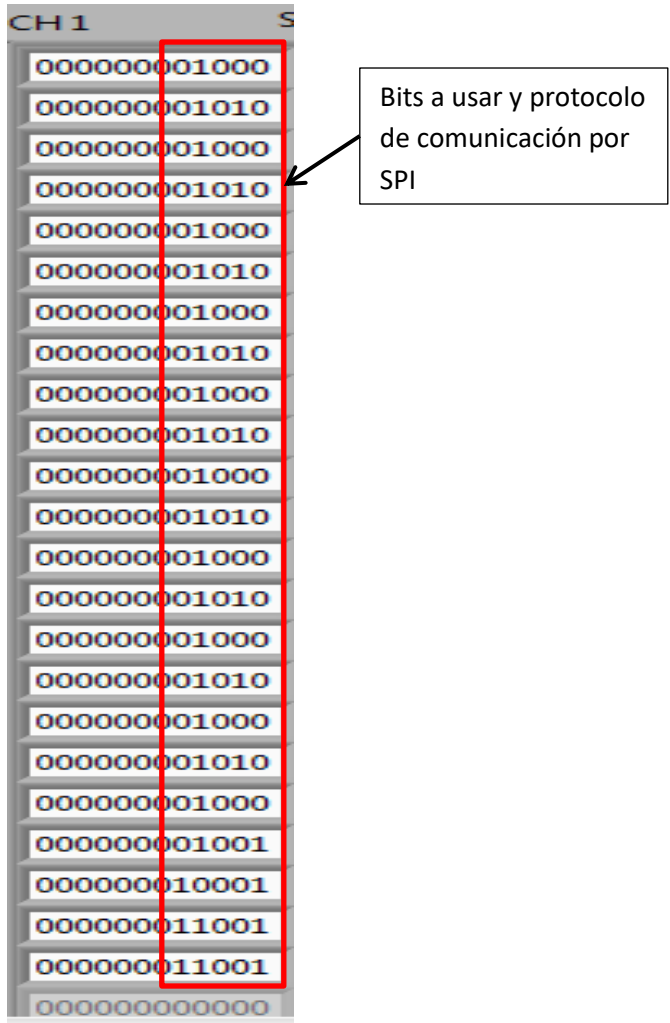


Figura 5.29: patrón digital para configurar el DAC MCP4921.

El patrón capturado se muestra en la Figura 5.29, donde el usuario se puede dar cuenta de los datos enviados y de la configuración asignada al DAC. En la figura 5.30 se visualiza el mismo patrón digital pero ahora en el analizador lógico de Altera. En la figura 5.31 el protocolo SPI es visualizado en el analizador lógico de la empresa Aligent; esto es a la salida del FPGA. Se observa que lo que se capturo como patrón digital para emular el protocolo de comunicación SPI, realmente se obtiene en los pines de salida asignados en el FPGA, estos datos son mostrados en la gráfica de la figura 5.32.



En la figura 5.33 tenemos la gráfica del generador de patrones digitales en la cual muestra el bit de datos del protocolo i2c.

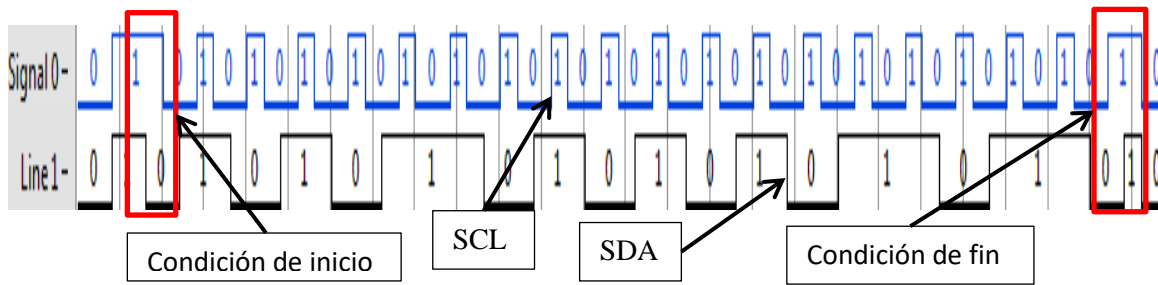


Figura 5.33: Protocolo de comunicación I2C.

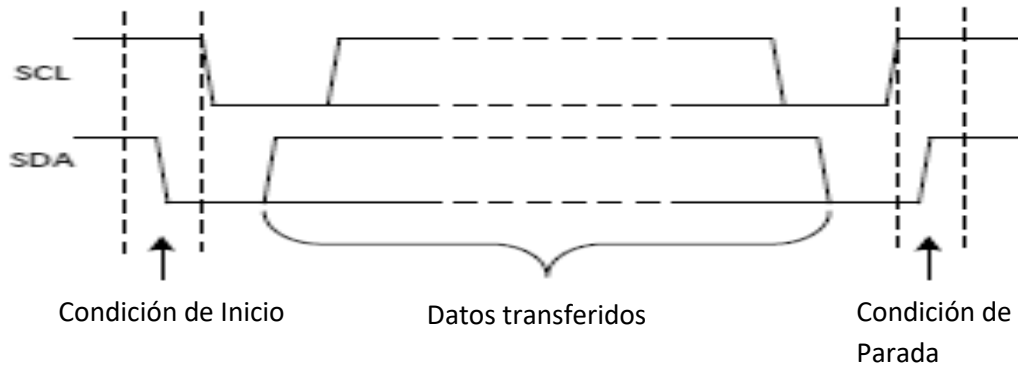


Figura 5.34: Definición de las condiciones de inicio y final del protocolo i2c.

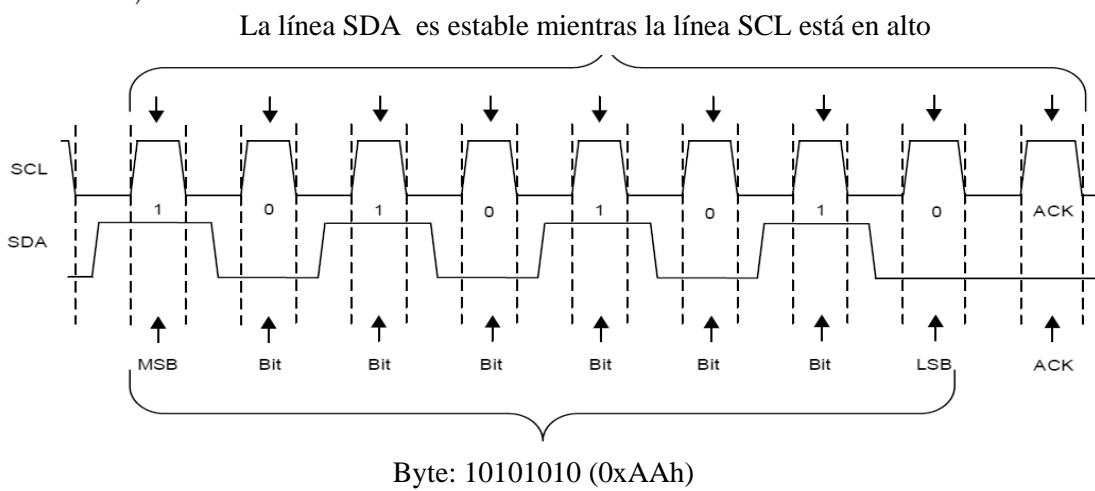


Figura 5.35: Diagrama de tiempos del protocolo I2C.



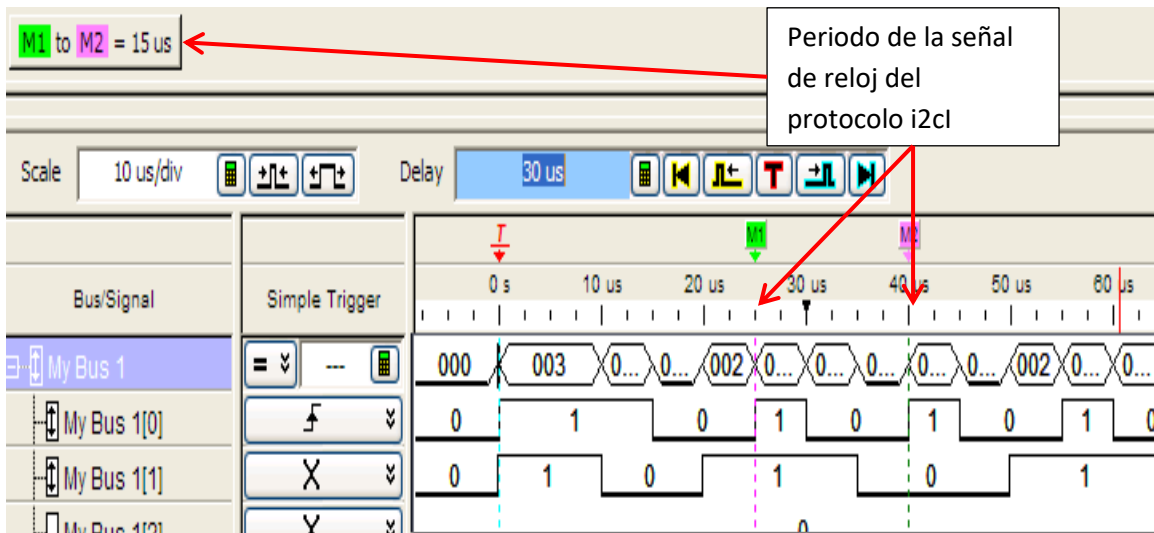


Figura 5.38: Periodo de la señal de reloj del protocolo i2c, frecuencia de 66.6666MHz.

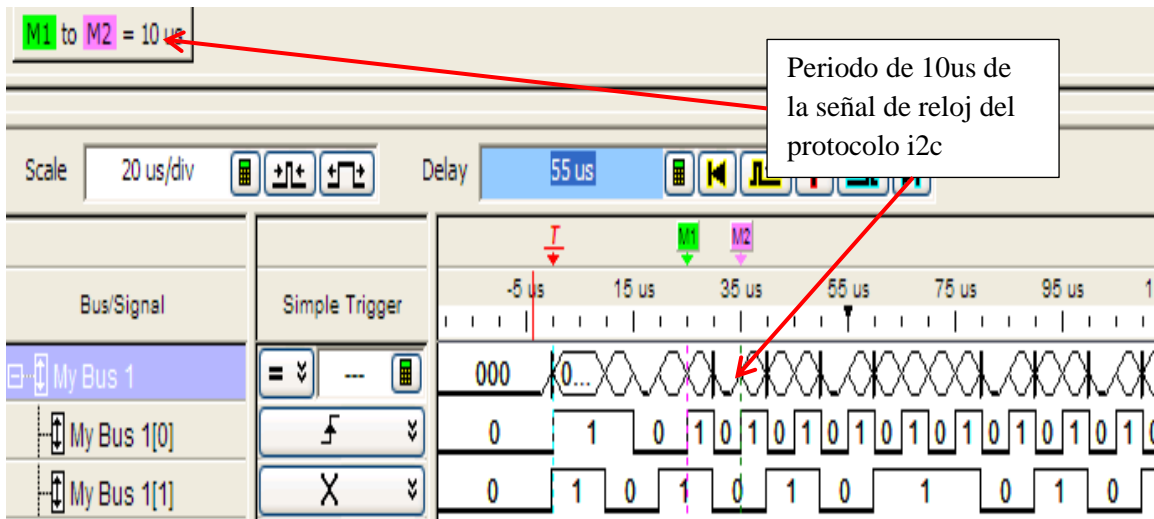


Figura 5.39: Periodo de la señal de reloj del protocolo i2c, frecuencia de 100kHz.

## 5.4 Conclusiones

La integración de hardware, firmware y software permitieron desarrollar e implementar el generador de patrones digitales. El uso del FPGA ayudo a disminuir la cantidad de componentes electrónicos necesarios, así como facilitar el diseño y desarrollo del dispositivo. La creación de la interfaz de usuario mediante la herramienta de LabView facilito la programación de la misma y ayudo a crear un ambiente intuitivo para el usuario.

## Capítulo 6 Conclusiones Generales

En este trabajo se realizó el diseño e implementación de un generador de patrones digitales programable a 50 MHz, con las características: paso máximo de reproducción 20ns, 24 bits de salida divididos en 2 canales de 12 bits cada uno, profundidad de memoria de 4 KBytes y un rango de amplitud por salida de  $\pm 5V$ . Adicionalmente, este cuenta con patrones digitales preestablecidos, con la capacidad de capturar y guardar nuevos patrones digitales editados por el usuario, la comunicación entre la PC y el dispositivo es inalámbrica, gracias al uso del módulo WiFi.

Como fue propuesto, el diseño del firmware del generador de patrones digitales se realizó en AHDL, con una interfaz de usuario intuitiva en LabView. Esta permite la asignación de salidas, niveles de voltaje digital, según la tecnología a usar, el paso de reproducción y la captura del patrón digital a reproducir, así como la previsualización del patrón digital capturado.

El generador de patrones desarrollado en este trabajo de tesis presenta las siguientes ventajas respecto a los dispositivos comerciales del mismo género: 1) una sería que el dispositivo es de arquitectura abierta, por lo cual puede ir mejorándose por generaciones de estudiantes futuras; 2) fácil de configurar el instrumento a través del software interactivo desarrollado, ya que la forma en que se capturan las tramas digitales es muy simple de hacer y reduce substancialmente el tiempo de captura, esto respecto a las interfaces usadas en los instrumentos comerciales más sofisticados.

El instrumento desarrollado presenta una reducción de tamaño y peso considerable respecto a un instrumento comercial de características semejantes, por lo que también se tiene una mejoría en la portabilidad, ya que el instrumento puede ser transportado junto una laptop en una mochila.

La comunicación por WIFI del generador de patrones facilita el uso y transportación del instrumento, ya que reduce significativamente el uso de cables planos que por general se dañan fácilmente, con lo cual el dispositivo reduce la posibilidad de fallas.

El generador de patrones puede producir señales de hasta 50 MHz en un intervalo de voltaje seleccionado por el usuario entre  $\pm 4.2V$  en forma individual de cada uno de los 24 canales del dispositivo. Aunque existen generadores de patrones digitales que pueden reproducir tres diferentes tipos de tecnologías digitales (CMOS, ECL y PCEL), en nuestro caso podemos reproducir esas tres, más la tecnología TTL; además se tiene la opción de editar personalmente los niveles lógicos deseados dentro del intervalo  $\pm 5V$ .

Así, de acuerdo a los resultados experimentales obtenidos en este trabajo de tesis podemos comentar que los objetivos planteados fueron cumplidos satisfactoriamente.

Un trabajo futuro sería mejorar la comunicación entre el instrumento y los circuitos que modifican los voltajes de las señales, con el fin de reducir el número de terminales conectadas entre el FPGA y dichos circuitos, ya que ahora se necesitan 6 por canal, lo que produce un total de 144 terminales del FPGA usadas para este fin. Si se logrará reducir este número de líneas, se tendría una mayor resolución en voltaje en las salidas.

También queda como trabajo futuro la implementación de protocolos de comunicación  $I^2C$  y SPI, pre-configurados por el sistema para que el usuario sólo introduzca la dirección del dispositivo, el o los datos a enviar, además que indique la frecuencia a utilizar como señal de reloj. Todo esto con la intención de facilitar el entendimiento de estos protocolos de comunicación y optimizar la caracterización o emulación de dispositivos que utilicen este tipo de protocolos.

# Apéndice A Participación en congreso

Constancia de Ponencia



## Diseño e Implementación de un Generador de Patrones Digitales Programable a 50 MHz dentro de un FPGA

T. M. Vargas<sup>1, a</sup>, S. Vergara<sup>1, b</sup>, M. A. Vargas<sup>1, c</sup>, F. Reyes<sup>1, d</sup>, J. Cid<sup>1, e</sup>

<sup>1</sup>Maestría en Ciencias de la Electrónica opción Automatización, Facultad de Ciencias de la Electrónica, Benemérita Universidad Autónoma de Puebla

<sup>a</sup>miguelxplo88@gmail.com, <sup>b</sup>svergara2@hotmail.com, <sup>c</sup>auroravargas@hotmail.com, <sup>d</sup>fernando.reyes@correo.buap.mx  
<sup>e</sup>jaime.cid@correo.buap.mx

**Resumen.** El avance de la tecnología ha permitido que sistemas, máquinas, procesos; involucrados en diferentes campos de la ciencia pueden ser automatizados reduciendo tiempo, riesgos y mejorando su productividad. El objetivo es el diseño e implementación de un generador de patrones digitales; siendo esencial en la investigación o desarrollo de circuitos integrados, automatización de pruebas de tarjetas electrónicas, sistemas de comunicación, verificación e investigación de fallas inducidas al sistema a examinar. Para el hardware se usa la tarjeta WI-FI RN-171-XV y el FPGA Cyclone IV; reciben los datos de la interfaz de usuario; donde el patrón digital es capturado; se indica la frecuencia de reproducción, tipo de reproducción. El protocolo de comunicación ofrece una ventaja simplificando la adquisición y reproducción de datos, los datos se reproducen correctamente obteniendo frecuencias desde 0.1Hz hasta 50MHz.

**Palabras clave** FPGA, Patrón Digital, Firmware, Interfaz de Usuario.

### 1. Introducción

En últimos años ha sido de gran interés científico el desarrollo de instrumentos de medición con el fin de aprovechar las ventajas de la instrumentación virtual, y fabricar generadores para laboratorio a un menor costo, aprovechando las ventajas que ofrecen los sistemas embebidos.

En el diseño de circuitos y sistemas digitales se requieren realizar pruebas con diferentes datos a diferentes frecuencias, con el propósito de verificar que el circuito integrado o sistema digital cumpla con las especificaciones del diseño. Así como también, se requieren realizar las mismas pruebas con el objetivo de conocer el funcionamiento de nuevos sistemas que son introducidos al mercado. Estas pruebas se logran hacer con un generador de datos digital.

Estos aparatos generan una serie de datos digitales (patrones) los cuales son aplicados al sistema que se requiere probar. En cada caso los patrones generados deben satisfacer las condiciones de la prueba a efectuar.

J.W. Mauchly (1956) diseño el dispositivo de chequeo de errores, el cual es capaz de comprobar el funcionamiento defectuoso de equipos electrónicos que utilizan patrones de impulsos eléctricos de naturaleza cíclica o repetitiva. Su

propósito es examinar errores transitorios o de estado estable en el patrón de impulsos y dar una indicación de tal situación [1].

El HP8175A (1987) es un generador de señales capaz de proveer hasta tres tipos de estímulos diferentes dependiendo de la configuración seleccionada. Los patrones de datos se pueden generar en 24 canales, cada uno de 1Kbit de profundidad [2].

El DG2020A (1995) por parte de la empresa Tektronix provee a los diseñadores con herramientas de alto performance para evaluar semiconductores digitales y circuitos lógicos. Cualquiera que fuera el proceso de diseño, caracterización, debug, validación o verificación [3].

Debido a que en muchas aplicaciones se requiere velocidad de procesamiento en tiempo real, utilizar una PC no resulta ser la mejor opción. Por ello la tecnología FPGA provee diferentes recursos los cuales facilitan la implementación de diferentes aplicaciones, además de que permite cambiar su configuración cuando se desee a medida del diseño [4].

En este trabajo se lleva a cabo el diseño e implementación de un generador de patrones digitales usando un FPGA como sistema embebido el cual se encarga de almacenar el patrón digital y reproducirlo en tiempo real. Para la captura del patrón digital se hace uso de la interfaz de usuario en LabVIEW una vez capturado el patrón digital es enviado de forma inalámbrica mediante el protocolo de comunicación WI-FI, recibiéndolos en el FPGA para su almacenamiento y reproducción. La interfaz de usuario está desarrollada para que su uso sea intuitivo, en ella se muestran dos Arreglos de 12 bits cada uno, donde se ingresan los datos del patrón digital, se ingresa la frecuencia de reproducción deseada y el tipo de reproducción repetitiva o solo una vez. Utilizar un FPGA como tarjeta embebida permite la adquisición y almacenamiento de los datos mediante la implementación de una memoria FIFO. Debido a las características que posee el FPGA permite el diseño de un generador de patrones digitales con un rango de frecuencias desde 0.1Hz hasta 5MHz, profundidad de memoria de 8K palabras de 24 bits, lo que permite crear patrones digitales con una gran gama de posibles aplicaciones.

## 2. Diagrama general del generador de patrones digitales

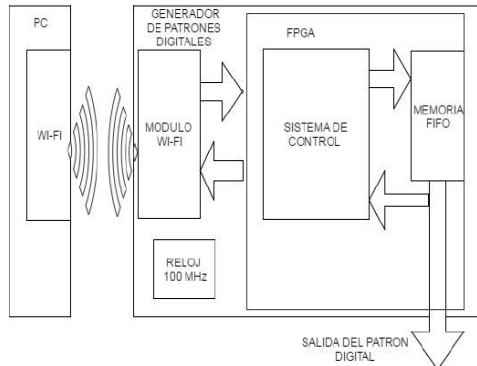


Fig. 1 Diagrama general del generador de patrones digitales

El generador de patrones digitales está formado por tres partes fundamentales el hardware, el firmware y el software.

El hardware del generador de patrones digitales se compone del módulo WI-FI, el FPGA. La tarjeta WI-FI se encarga de recibir la información que es enviada por el puerto WI-FI de la PC, estos datos ahora son enviados en formato serie al FPGA.

Por sí solo el FPGA es un integrado con un gran número de compuertas lógicas sin conectar. El firmware es quien configura las compuertas lógicas del FPGA para crear el "sistema de control" y la "memoria FIFO". El "sistema de control" se encarga de determinar si la información recibida por el módulo WI-FI es alguna instrucción del firmware. Si el dato recibido es una instrucción válida entonces procede a: escribir el dato en la memoria FIFO, leer la memoria FIFO, configurar la frecuencia de lectura, detener la lectura de la memoria FIFO; según la instrucción recibida. La "memoria FIFO" almacena la información del patrón digital, para su posterior lectura y así crear el patrón digital que se recibió por el módulo WIFI. Para el correcto funcionamiento del FPGA se necesita una señal síncrona y debido a que el objetivo es reproducir el patrón digital a 50MHz entonces se utiliza una señal de reloj de 100MHz.

En el bloque de PC, se muestra el software de LabVIEW, en el cual se lleva a cabo el desarrollo de la interfaz de usuario del generador de patrones digitales; en la primera ventana de esta interfaz, se captura el patrón digital que el usuario desee o se lee uno previamente guardado, también se configura la frecuencia de reproducción del patrón digital, así como el tipo de reproducción; ya sea repetitiva o solo una vez.

Los datos previamente capturados o leídos, son enviados por el puerto WI-FI a la tarjeta del generador de patrones digitales en donde se determina si el dato recibido es un comando o forma parte del patrón digital, el cual se almacena en una memoria FIFO, para su posterior lectura y escritura. Cada dato enviado está formado por una palabra de 5 bytes

(40 bits), en donde el byte más significativo es el comando a ejecutar en el hardware del generador de patrones digitales.

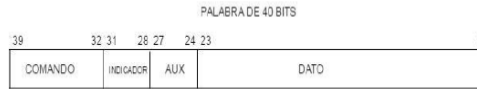


Fig. 2 Organización de la palabra de 40 bits

## 3. Hardware del instrumento

Los FPGA's son dispositivos que están compuestos por una matriz elementos lógicos programables. La programación de dichos dispositivos define la manera en que funciona cada bloque lógico. Los FPGA's a diferencia de los microcontroladores y microprocesadores no son secuenciales, sino que son circuitos combinacionales, sus salidas dependen de entradas que provienen de otros elementos lógicos, son de propósito específico.



Fig. 3 Tarjeta DE0-Nano de la marca Terasic

Para el desarrollo del generador de patrones digitales se eligió como tarjeta de pruebas la DE0-NANO de Terasic la cual cuenta con un FPGA Cyclone IV EP4CE22F17CN6 de la marca altera de 22,320 elementos lógicos con un reloj de 50 MHz, 32 MB de SDRAM, 2K EEPROM y 64 Mb de memoria de configuración, además de un convertidor analógico a digital de 8 canales de 12 bits, un acelerómetro de 3 ejes. La tarjeta DE0-Nano incluye USB Blaster para programar el FPGA.[4]



Fig. 4 Tarjeta RN-171\_XV

El módulo RN-171-XV provee conectividad WI-FI usando los estándares legales del protocolo 802.11 b/g, el cual incorpora un radio 802.11 b/g, un procesador SPARC de 32 bits, secuencia TCP/IP, reloj de tiempo real. Este módulo esta precargado con firmware para simplificar la integración y minimizar el tiempo en el desarrollo de las aplicaciones. En la configuración sencilla para crear una conexión inalámbrica, el módulo solo requiere de cuatro conexiones: Alimentación, TX, RX, GND.

UART--El Universal Asynchronous Receiver Transmitter, permite establecer una comunicación serial entre dos dispositivos. Ambos dispositivos deben ser configurados para

que funcionen a un flujo de datos predeterminedado, a esta tasa de transferencia se le conoce como baudrate o tasa de bits por segundo [5]. La comunicación básica se muestra en la siguiente figura:

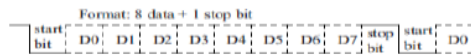


Fig. 5 Protocolo de comunicación UART-232

Cuando la transmisión cambia de "1" a "0" y a este bit se le conoce como bit de inicio. Los bits de información son transferidos del menos significativo al más significativo usualmente son 8 bits y por último el bit de finalizado.

La velocidad de transferencia está dada por:

$$\text{bits por segundo (bps)} = \frac{1}{\text{tiempo por bit}} \quad (1)$$

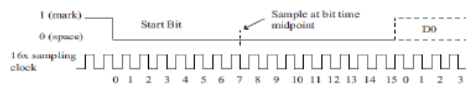


Fig. 6 Tiempo de muestreo del dato recibido por WI-FI

El total de bits por segundo es el inverso del tiempo que se toma en transferir un bit. Para garantizar una transferencia adecuada, el tiempo de muestreo debe ser 16 o 64 veces mayor que el de transferencia; por si existiera algún desfase.

#### 4. Firmware del instrumento

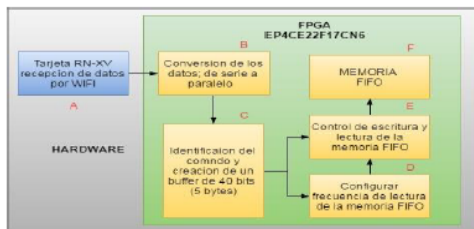


Fig. 7 Diagrama de bloques del firmware

Para la instrumentación del firmware del generador de patrones digitales, implementando en un FPGA CYCLONE IV, se utiliza el lenguaje AHDL en el software QUARTUS II. A continuación se muestra el firmware del generador de patrones digitales en diagrama de bloques.

El bloque "A" representa la tarjeta RN-171-XV; recibe los datos enviados por la computadora y los muestra en su terminal Rx; son recibidos en el bloque "B" llamado UART\_RX2. Este bloque se encarga de convertir los datos seriales de la tarjeta WI-FI a paralelos. Para identificar en que

momento debe a empezar a armar la palabra recibida, el bloque hace uso del protocolo UART232

El bloque "C" WOR32\_28\_1 se encarga de discriminar la información; cuando recibe un comando valido; arma el bus de 40 bits y lo distribuye sus salidas.

El bloque "D" CLKFIFO; funciona como divisor de una frecuencia principal (200 MHz) la cual es dividida por un contador interno el cual es configurado desde la interfaz del usuario.

El bloque "E" clkfifowifi2 se encarga de controlar la memoria FIFO; si el comando recibido indica escribir en la memoria, genera las condiciones necesarias para que el dato recibido sea escrito en la memoria FIFO; si el comando recibido indica lectura de la memoria, se generan las condiciones necesarias para leer y escribir en la memoria el dato leído; así mantiene la información en la memoria y puede ser leída nuevamente sin ningún problema.

El bloque "F" FIFOWifi es la memoria FIFO de 8K de 24 bits de palabra donde se guardan los datos del patrón digital, a esta memoria se le habilita la entrada CLR asincrónica.

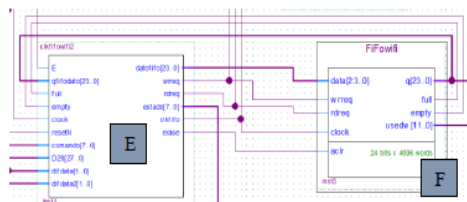


Fig. 8 Bloque "clkfifowifi2" controla la memoria y el bloque "FiFowifi" memoria FIFO

#### 5. Software del instrumento

Para el diseño de la interfaz de usuario se desarrolló en LabVIEW. El software se divide en secciones que permiten ejecutar las instrucciones en serie como se muestra en el diagrama

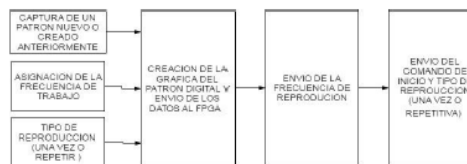


Fig. 9 Diagrama de bloques del software

En la primera ventana "DATOS" del software se muestran: los dos arreglos donde se ingresan los datos del patrón digital, en la caja de texto se ingresa el nombre del archivo que se a crear o leer, los botones para leer o guardar el patrón digital, los botones para elegir el tipo de reproducción (una vez o repetitiva), el botón limpiar datos borra el patrón capturado o leído.

Al pasar a la segunda ventana "GRAFICAS", el software procede a identificar el número de renglones que forman el patrón digital capturado por el usuario; para determinar el número de renglones que se tienen que graficar y enviar al FPGA.

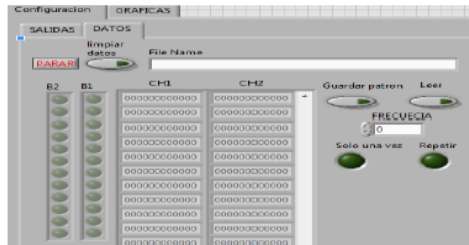


Fig. 10 Interfaz de usuario programada en LabVIEW

Se lee cada renglón de los dos arreglos y se va construyendo la gráfica del patrón digital, para después enviar el dato leído por WI-FI junto con el comando "escribir"; debido a que el sistema donde se está ejecutando nuestro software es multitarea la velocidad de transmisión de datos es muy lenta compara con la velocidad de escritura del FPGA (100MHz); surge la necesidad de poder guardar el dato recibido una sola vez; excepto que el patrón digital indique lo contrario; se utilizan dos bits de la palabra de 5 bytes para poder indicar que se trata de un dato valido a guardar en la memoria FIFO. Una vez que termino de graficar y enviar el patrón digital; procede a enviar el comando de la frecuencia de reproducción de la memoria FIFO junto al valor asignado por el usuario, para finalmente enviar el comando de inicio de reproducción del patrón digital.

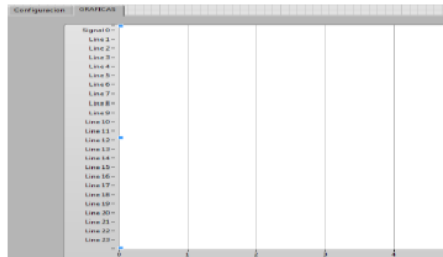


Fig. 11 Pestaña de visualización del patron digital en la interfaz de usuario

## 6. Resultados experimentales

Buscando aprovechar las capacidades que ofrece la PC y el FPGA para el procesamiento de los datos, por lo que cada sistema cumple una función diferente. Por parte de la PC se programó la interfaz la cual grafica los datos de patrón digital, de esta manera el FPGA cumple la función de la adquisición de los datos, además de que reproduce el patrón digital a la frecuencia a la cual se desea trabajar.

Para mostrar el funcionamiento del generador de patrones digitales; se muestran diferentes patrones digitales, los cuales

son enviados al FPGA para mostrarlos en el analizador lógico del software Quartus II.

### A. Todos uno, todos cero.

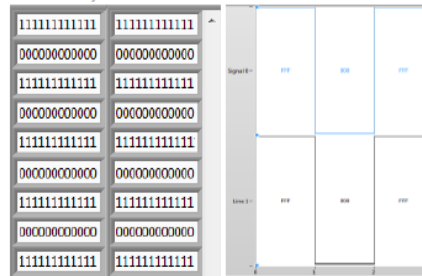


Fig. 12 Patrón digital: todos uno, todos cero y su grafica en bus de datos

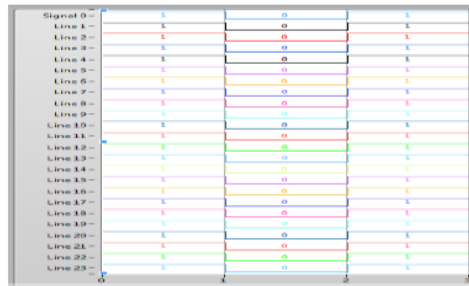


Fig. 13 Grafica de todos uno todos cero con los bus expandidos

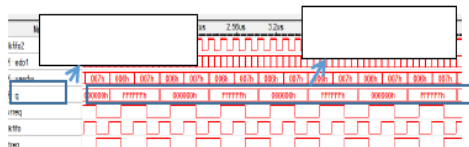


Fig. 14 Grafica en tiempo real de todos uno, todos cero en el analizador lógico en modo bus

### B. Recorriendo uno, de derecha a izquierda.

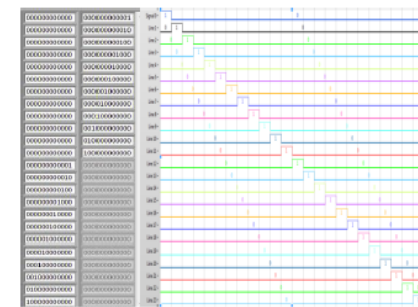


Fig. 15 Patrón digital de recorrer el uno y grafica del patrón digital

Es importante hacer notar que la gráfica digital en LabVIEW empieza de arriba hacia abajo es decir el bit menos significativo es el que está arriba y el más significativo esta abajo; en el analizador lógico de Quartus II el bit menos significativo es el que esta abajo y el más significativo hasta arriba.

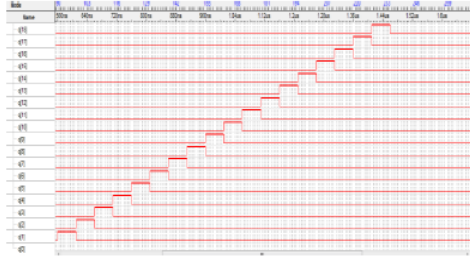


Fig. 16 Gráfica en tiempo real del analizador lógico del patrón digital recorrer el uno.

### C. Protocolo de comunicación SPI del DAC MCP4921

El DAC 4921 es un convertidor digital-analógico de 12 bits, el cual recibe sus datos en formato serie, en la entrada SDI.

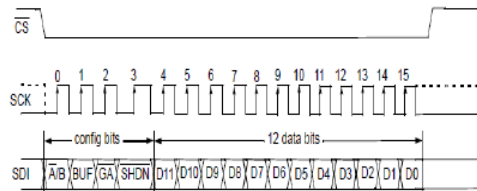


Fig. 17 Diagrama de tiempos del protocolo SPI del DAC MCP4921

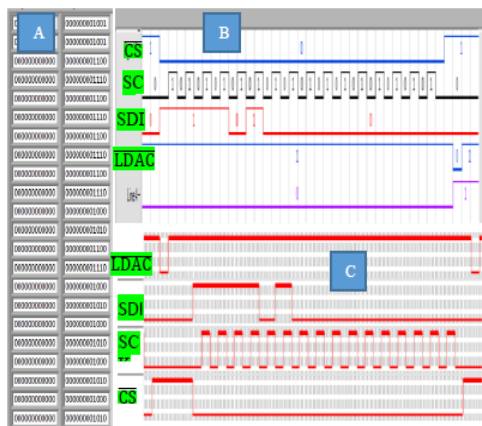


Fig. 18 "A" Datos del patrón digital del protocolo SPI. "B" gráfica del patrón digital; en la interfaz de usuario del protocolo SPI del DAC MSP4921; "C" Gráfica en tiempo real del protocolo en el analizador lógico de Quartus II

## 7. Conclusiones

Gran parte del generador de patrones digitales, se ha desarrollado e implementado está controlado desde una PC con una interfaz de usuario intuitiva. Gracias al uso de la instrumentación virtual es posible realizar futuras mejoras al software. El emplear un FPGA como sistema embebido nos da la capacidad de poder implementar la mayoría de los circuitos involucrados en un solo integrado; aumentando el nivel de integración en la tarjeta, disminuyendo los componentes externos, del generador de patrones digitales; así como mejorar el firmware; como aumentar el número de salidas, controlar diferentes niveles de voltaje para las salidas del sistema.

El generador de patrones digitales es una herramienta importante en el desarrollo o investigación de nuevos dispositivos.

Conocer el comportamiento del control de un sistema con las señales emuladas es de suma importancia.

A partir del análisis de resultados, las conclusiones deberán exponer claramente las contribuciones del trabajo realizado.

De acuerdo a las pruebas realizadas la memoria FIFO no debe trabajar a más de 100 MHz; se recomienda no trabajar a más de esa frecuencia, aun cuando las hojas de datos indiquen que puede trabajar a más.

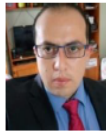
## 8. Referencias

- [1] Patente US3105955A Error Checking Device. Disponible en: <https://www.google.ch/patents/US3105955>  
J. W. MAUCHLY ERROR CHECKING DEVICE, Archivado 28 de Marzo de 1956 oci. 1, 1963 s vit@ Patented Oct. 1, 1963 3,105,955 ERROR CHECKING DEVICE John W. Mauchly, Ambler, Pa., assigner to Sperry Rand Corporation, New York, NN., a corporation of Dela- Ware Filed Mar. 2s, 195s, sa. Ne. 574,546 7 claims. (ci. 34e-146.1)
- [2] HP-8175A. Hewlett-Packard Journal, Abril 1987, volumen 38, numero 4.
- [3] DG2020A datasheet Data pattern Generator
- [4] G.R. Tsai, M.C. Lin, G.S. Sun, Y.S. Lin. "Single chip FPGA-based reconfigurable instruments". The International Conference on Reconfigurable Computing and FPGAs, sep. 2004.
- [5] DE0-NANO user manual.2003-2013 Terasic technologies
- [6] Javier Ruiseco López, "Control de un robot articular de 1 grado de libertad utilizando un sistema embebido via WiFi"; Tesis de Maestría en Ciencias de la Electrónica opción Automatización, FCE-BUAP, 2012.
- [7] Ronal J. Tocci, Neal S. Widmer, Gregory L. Moss, Decima edicion (2007) "Sistemas digitales. Principios y aplicaciones".
- [8] Microchip, "WiFly Command Reference, Advanced Features and Applications User's Guide."
- [9] H. Goldberg. "What is Virtual Instrumentation" , IEEE Instrumentation & Measurement Magazine, dic. 2000, pp. 10-13.
- [10] FPGA CPLD and ASIC from Altera, Disponible en: <http://www.altera.com/>

[11] XYZs of Signal Generators, Tektronix, 17 de octubre 2007

[12] Charles K. Alexander, Matthew N.O. Sadiku, "Fundamentos de circuitos Electricos", tercera edición 2006

## 9. Biografías



**Tomás Miguel Vargas García** Nació el 28 de enero de 1988 en Puebla, Puebla, México. Realizo su licenciatura en Ingeniería en Mecatrónica en la Facultad de Ciencias de la Electrónica en la Benemérita Universidad Autónoma de Puebla en 2014, estudiante de la maestría de la Facultad de Ciencias de la Electrónica de la Universidad Autónoma de Puebla, México.



**Sergio Vergara Limón**, nació en la ciudad de Puebla, Puebla el 7 de octubre de 1970. Licenciatura en Electrónica de 1970. Licenciatura en Electrónica FCE-BUAP Puebla, Puebla, México, 1994. Maestría en Ciencias especialidad en Optoelectrónica FCFM-BUAP, Puebla, Puebla, México, 1997, Doctorado en Ciencias especialidad en Optoelectrónica, FCFM-BUAP, Puebla, Puebla, México, 2000. Él es desde el 2005 Profesor Investigador en la Facultad de Ciencias de la Electrónica de la BUAP en la ciudad de Puebla, Puebla, México, sus líneas de investigación son la instrumentación electrónica, la robótica, el control y la automatización.

El Dr. Vergara pertenece al Sistema Nacional de Investigadores nivel I, posee perfil PROMEP y es profesora de la Maestría en Ciencias de la Electrónica opción Automatización, es autor de más de 100 artículos en revistas indexadas, asesor de 24 tesis de maestría terminadas, tiene colaboración internacional con el centro europeo de Investigaciones Nucleares CERN en el experimento ALICE.



**Maria Aurora Diozcora Vargas Treviño**, nació en la ciudad de Puebla, Puebla el 14 de diciembre de 1972. Licenciatura en Electrónica FCE-BUAP Puebla, Puebla, México, 1994. Maestría en Ciencias especialidad en Optoelectrónica FCFM-BUAP, Puebla, Puebla, México 1997, Doctorado en Ciencias especialidad en Optoelectrónica, FCFM-BUAP, Puebla, México, 2000. Ella es desde el 2005 Profesor Investigador en la Facultad de Ciencias de la Electrónica de la BUAP en la ciudad de Puebla, Puebla, México, sus líneas de investigación son la instrumentación electrónica, la robótica, el control y la automatización.



**Fernando Reyes**, Es profesor investigador titular C de la FCE-BUAP. En 1984 obtuvo la Licenciatura en Ciencias de la Electrónica en la FCM-BUAP. En 1990 obtuvo la Maestría en Ciencias con Especialidad en Electrónica en el INAOE. Bajo la asesoría del Dr. Rafael Kelly, en 1997 culminó el Doctorado en Ciencias con Especialidad en Electrónica y Telecomunicaciones en el CISESE. Es miembro del Sistema Nacional de Investigadores desde 1993 a la fecha. Actualmente es nivel II. Es autor de más de 150 artículos nacionales e internacionales.



**Jaime Cid Monjaraz**. Nació en Puebla, Puebla, México el 28 de enero de 1959. Realizó su licenciatura en Electrónica en la Facultad de Ciencias Físico Matemáticas de la BUAP en 1985, Maestría en Control Automático en el Instituto Tecnológico de Puebla en 1995 y el Doctorado en Ingeniería Mecatrónica en la Universidad Popular Autónoma del Estado de Puebla en el año de 2007 en Puebla, México. Él actualmente labora en la Benemérita Universidad Autónoma de Puebla, es Investigador en la Facultad de Ciencias

de la Electrónica. Profesor en la BUAP desde 1981 a la fecha impartiendo los cursos en las áreas de Matemáticas, Ciencias Computacionales, Control Moderno. Es autor de 3 libros, con más de 100 publicaciones científicas, en congresos nacionales e internacionales y revistas, asesor en más de 50 trabajos de tesis profesionales de licenciatura y maestría, tiene 9 solicitudes de patente, su línea de investigación es en Visión y Robótica.

El Dr. Cid es senior member del IEEE y miembro de distintas asociaciones académicas y científicas. Ha sido director de la Facultad de Ciencias de la Electrónica de la BUAP, Director de Divulgación Científica de la Vicerrectoría de Investigación y Estudios de Posgrado de la BUAP, miembro del Padrón de Investigadores de la VIEP y del Padrón de Consultores de la BUAP.

## Apéndice C Manual de usuario

El propósito de este manual es el de facilitar al usuario, la operación de diferentes funcionalidades que se muestran en la interfaz de usuario del generador de patrones digitales.

- a) Requerimientos del hardware
  - 1GB de memoria RAM
  - Tarjeta WiFi
- b) Requerimientos del software
  - Windows 7 o superior.
  - LabVIEW 2015
- c) Requerimientos del sistema
  - Fuente de alimentación simétrica de  $\pm 12V$
  - Fuente de alimentación a 5V

### 1.- Suministro de alimentación

Para encender la tarjeta del generador de patrones digitales primero conectamos la fuente de 5V a la tarjeta DE0-NANO ya sea con un eliminador de voltaje o con el puerto USB de una computadora. Posteriormente se conecta la fuente simétrica de  $\pm 12V$  a la tarjeta acopladora de voltaje.

### 2.- Conectividad

Una vez conectada la tarjeta DE0-NANO, la tarjeta de red WiFi creara una red dedicada al generador de patrones digitales. Con la computadora debe buscar la red Wifly-xxx y da click para conectar (mientras la computadora se encuentre conectada a este punto de acceso, no tendrá acceso a internet). La computadora se conectara al generador de patrones digitales.

### 3.-Acceso a la interfaz de usuario.

Una vez establecida la conexión WiFi, podrá ingresar a la interfaz de usuario en el software de LabVIEW. La interfaz de usuario se divide en las siguientes secciones: configuración y gráficas.

En la pestaña configuración tenemos otras tres pestañas:

- Salidas
- Voltaje
- Datos



Manual figura 1.1 Pestañas del software del generador de patrones digitales.

## DESCRIPCIÓN DEL SOFTWARE

### SALIDAS

En la pestaña Salidas tenemos los 24 bits de salidas disponibles, el arreglo de Nombre del Bus y los bits asignados a cada Bus.



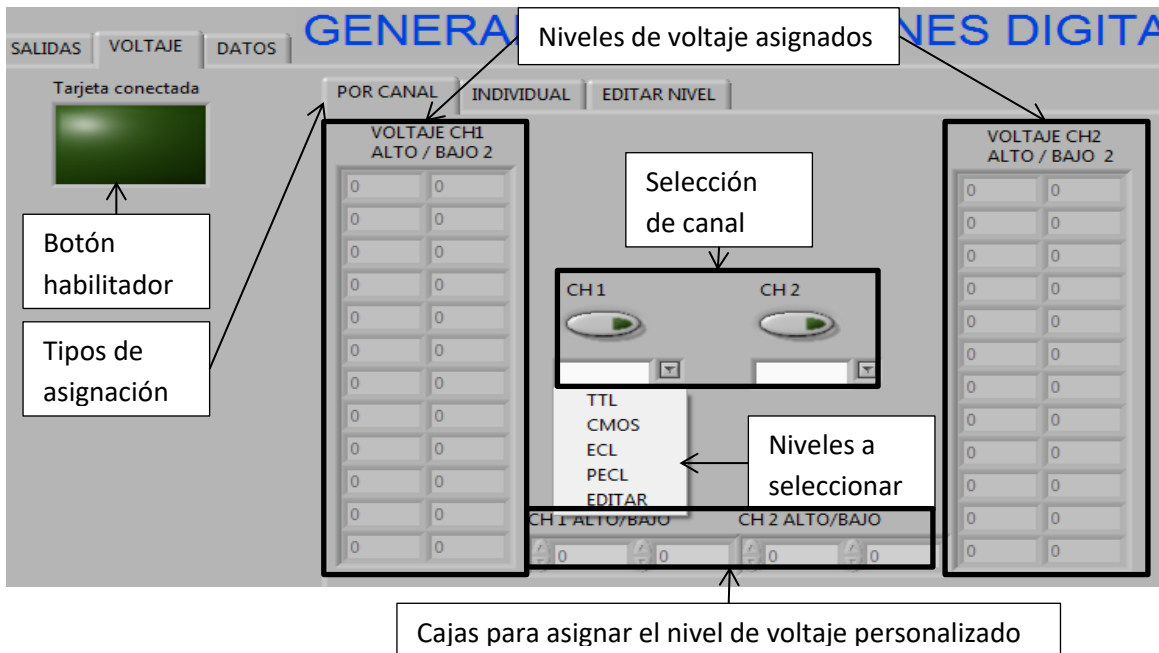
Manual figura 1.2: Pestaña SALIDAS.

Para poder seleccionar las salidas a usar, primero debe nombrar el bus de las salidas o el nombre de la salida. Después selecciona las salidas a asignar a la etiqueta del Bus, el número de salidas asignadas se muestra en el arreglo de No. Bits, y los bits asignados a la etiqueta se muestran el arreglo de Bits asignados.

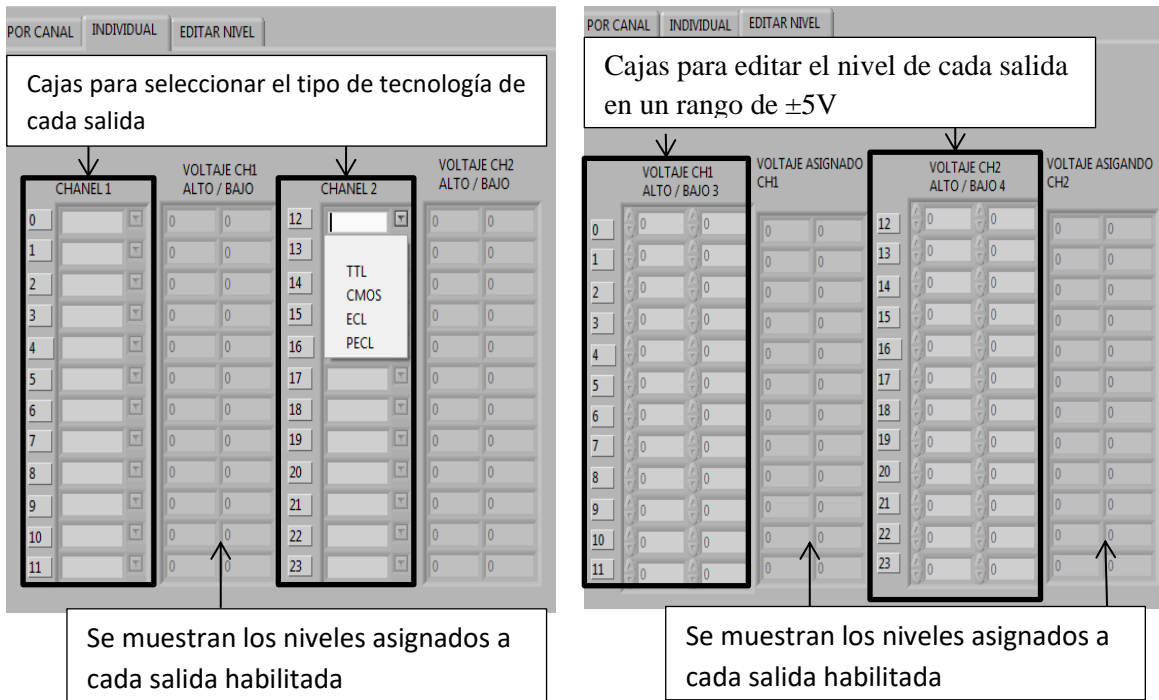
### VOLTAJE

En la pestaña Voltaje puede seleccionar los diferentes niveles de voltaje para cada canal, de manera individual o editar el nivel de voltaje de manera individual. Para poder asignar el nivel de voltaje para cada salida el usuario debe indicar que la tarjeta acopladora de voltaje se encuentra conectada.

El nivel de voltaje asignado se muestra en los arreglos de cada canal, donde se muestra el nivel alto y bajo década salida habilitada en cada canal. En la pestaña individual, cada salida puede ser configurada a una tecnología digital diferente



Manual figura 1.3: Pestaña VOLTAJE- POR CANAL.

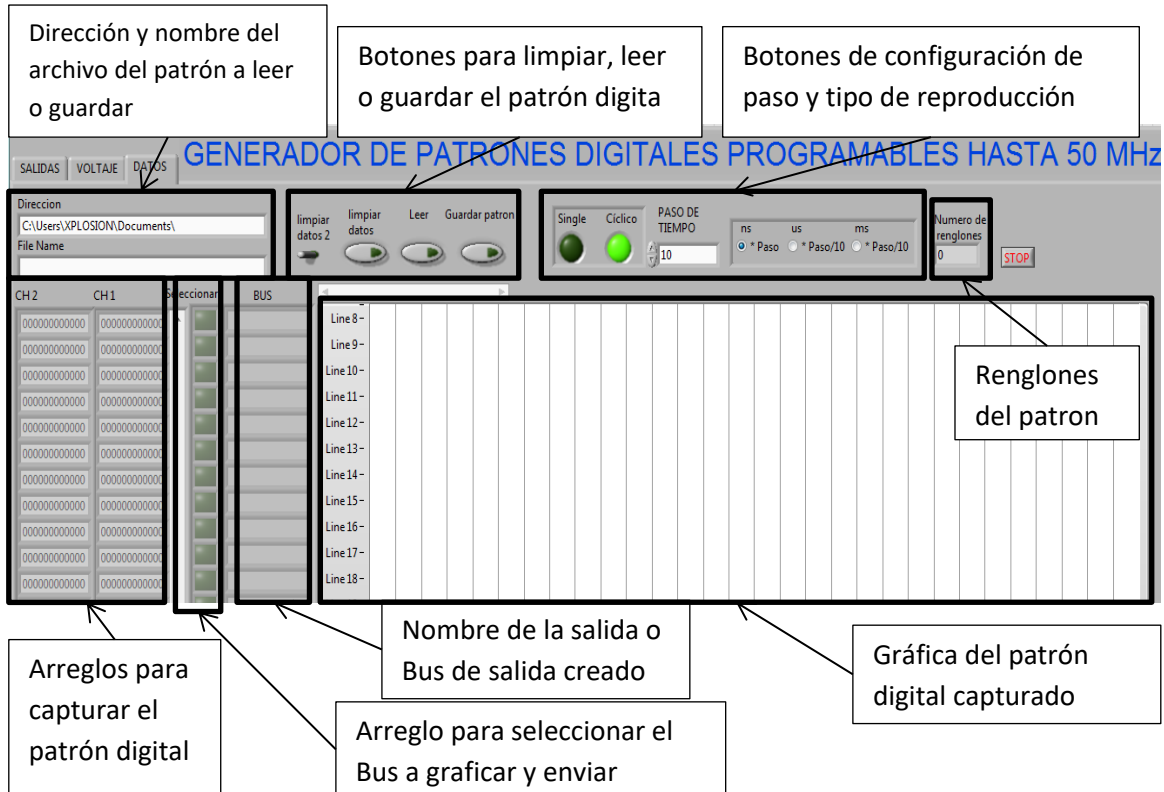


Manual figura 1.4: Pestaña VOLTAJE-INDIVIDUAL - EDITAR NIVEL.

## DATOS

Después de asignada la tecnología con la que se quiere trabajar, podemos proceder a capturar o leer un patrón digital previamente guardado. Para ello procedemos a dar clic en la pestaña DATOS.

En la pestaña DATOS, se observan dos arreglos CH1 y CH2, cada renglón del arreglo está formado por 12 bits siendo el primer bit el MSB y el último bit el LSB. La ponderización es de derecha a izquierda, en cada canal.



Manual figura 1.5: Descripción de la pestaña DATOS.

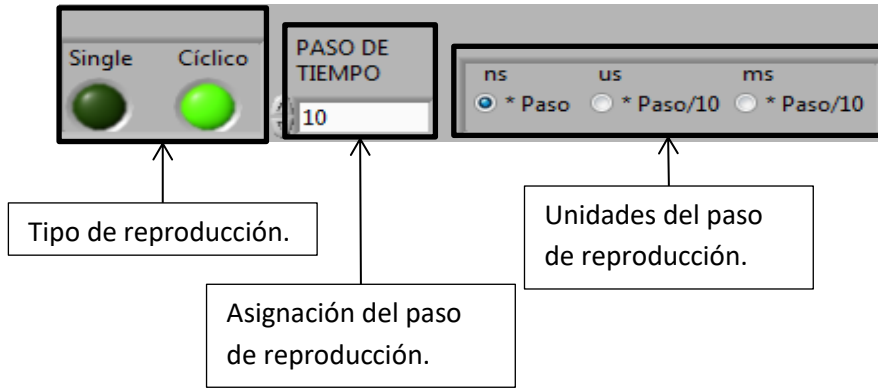
En la caja dirección, se escribe la ruta donde se va a leer o guardar el patrón digital; en la caja file name se escribe el nombre del patrón digital a leer o guardar.

En la caja de "Numero de renglones" se muestra el número de renglones que forman el patrón digital.

EL tipo de reproducción predeterminado es "CICLICO" y el paso de reproducción predeterminado es de 10 ns.

En el tipo de reproducción "CICLICO" el sistema repite indefinidamente el patrón digital, al paso de reproducción establecido por el usuario, hasta que el usuario regrese a la pestaña de configuración o presione el botón "STOP".

En el tipo de reproducción “SINGLE” el sistema solo reproduce una sola vez el patrón digital, y solo lo vuelve a hacer hasta que el usuario regrese a la pestaña configuración y pase de nuevo a la pestaña gráficas.

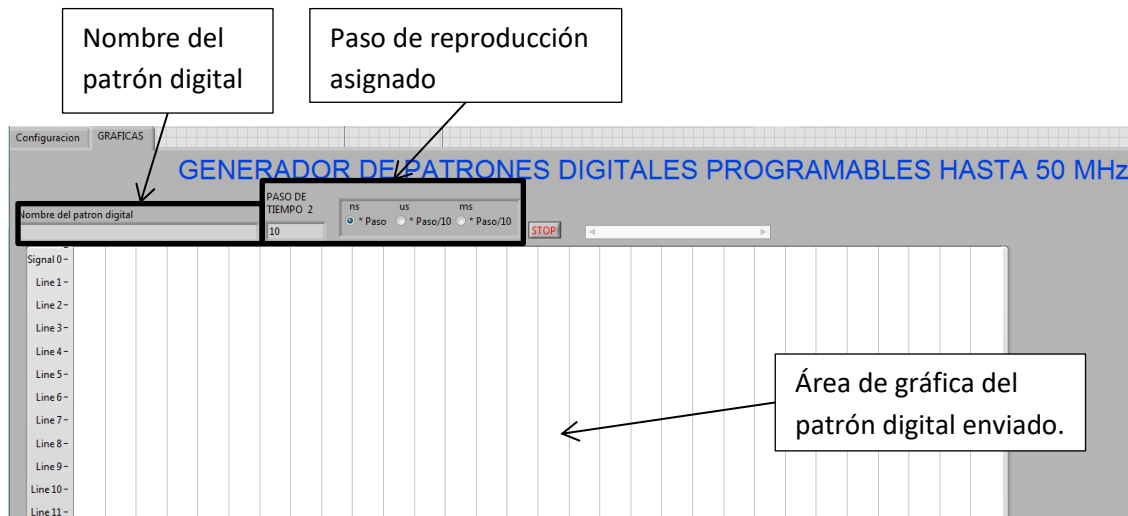


Manual figura 1.6: Botones de configuración del paso y tipo de reproducción.

La caja de “PASO DE TIEMPO” incrementa de 10 en 10 y trabaja en conjunto con la unidad de tiempo asignado: ns, us, ms.

## GRAFICA

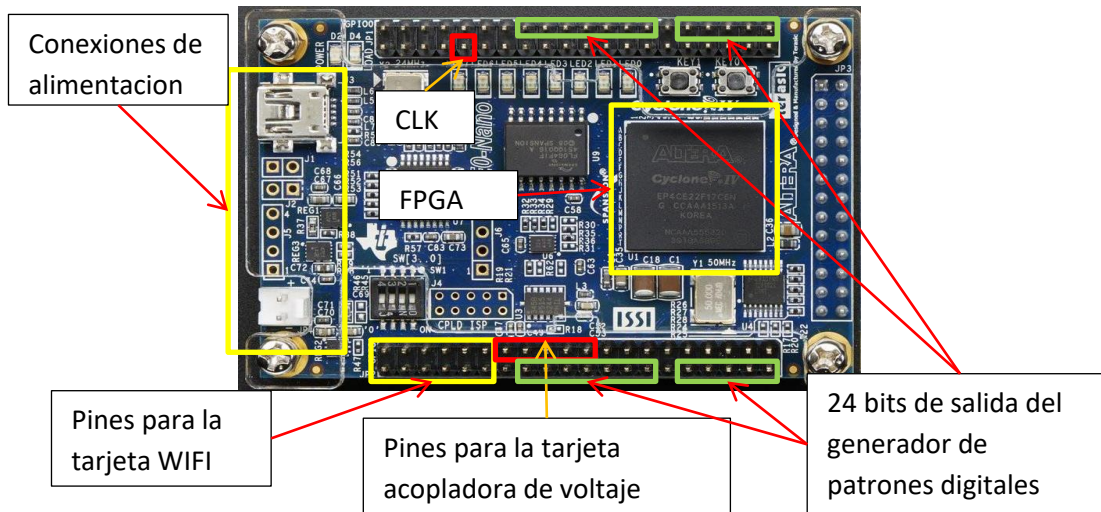
En esta pestaña se grafican y se envían los datos del patrón digital a la tarjeta electrónica que se encarga de reproducir el patrón digital según la configuración realizada por el usuario en la interfaz gráfica.



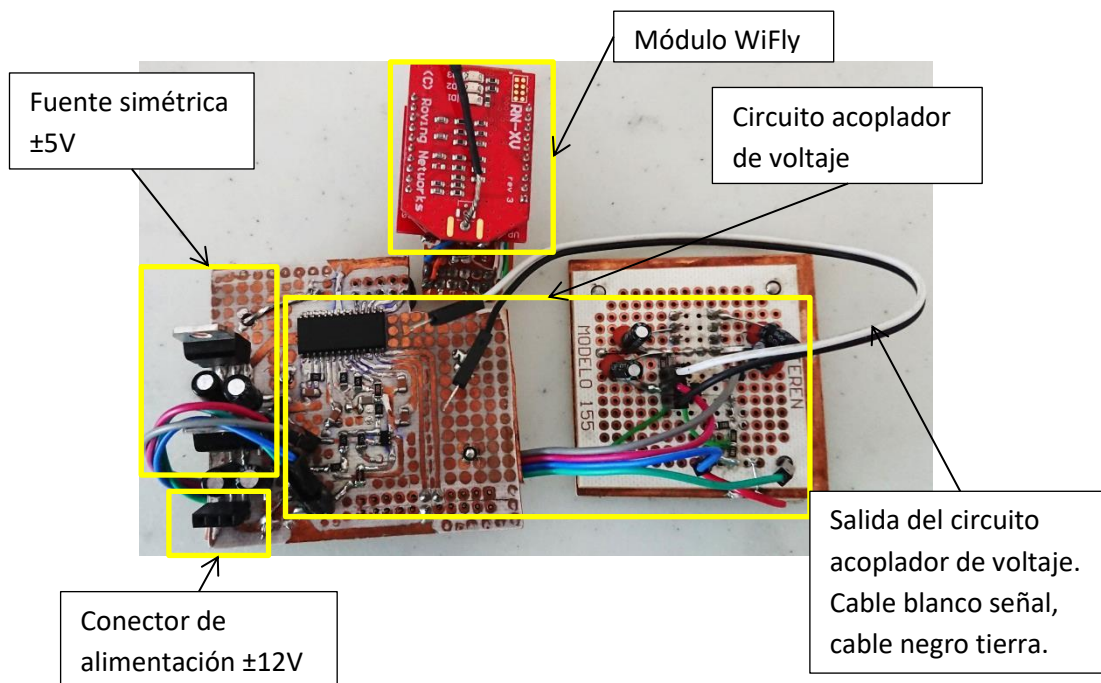
Manual figura 1.7: Pestaña de Grafica.

## DESCRIPCIÓN DEL HARDWARE

A continuación se describen las conexiones de las tarjetas que forman el generador de patrones digitales.



Manual figura 1.8: Conexiones del Hardware.



Manual figura 1.9: Descripción de las conexiones de la tarjeta acopladora de voltaje.

## **INSTRUCCIONES DE USO**

- 1.- Nombrar la etiqueta de la salida o bus de salidas.
- 2.- Seleccionar la salida o salidas de la etiqueta.
- 3.- Indicar si la tarjeta acopladora de voltaje se encuentra conectada.
- 4.- En caso de estar conectada la tarjeta acopladora de voltaje, seleccionar el tipo de tecnología digital, ya sea por canal, individual o editar nivel.
- 5.- En la pestaña "Datos" ingresa o lee un patrón digital previamente guardado.
- 6.- Selecciona el tipo de reproducción.
- 7.- Asigna el paso de reproducción.
- 8.- Selecciona la pestaña gráfica para iniciar la reproducción del patrón digital.

## Referencias

- [1] PI-5800a. Hoja de especificaciones, Pulse Instruments, Disponible en: <http://www.pulseinstruments.com/pi5800a/default.htm>
- [2] HP-8175A. Revista Hewlett-Packard Journal, Hewlett-Packard, volumen 38, numero 4, Abril 1987.
- [3] DG2020A. Hoja de especificaciones, Tektronix Disponible en: <http://www.tek.com/datasheet/dg2020a>
- [4] Wave Gen Xpress 8MB - 100 MHz 16-bit. Hoja de especificaciones, Byte Paradigm, Disponible en: <http://www.saelig.com/product/SS004.htm>
- [5] IO-3208 Logic Analyzer, Pattern Generator. Hoja de especificaciones, Link Instruments. Disponible en: [https://www.linkinstruments.com/logana32.html#pattern\\_generator\\_option](https://www.linkinstruments.com/logana32.html#pattern_generator_option)
- [6] PG3A. Hoja de especificaciones, Tektronix Disponible en: <http://www.tek.com/logic-analyzer/digital-pattern-generator>
- [7] Tocci, R. J, Widmer, N.S. Sistemas digitales: Principios y aplicaciones, México, Pearson Education, (10ª edición), (2007).
- [8] William R. Blood, Jr. MECL System Design Handbook, On Semiconductor, Rev.1ª, Four Edition, (1988).
- [9] John F. Wakerly, Diseño digital: principios y prácticas, Pearson Educación, pag. 182, (2001).
- [10] XYZs of Signal Generators, Tektronix, 17 de octubre (2007).
- [11] Microchip, "WiFly Command Reference, Advanced Features and Applications User's Guide."
- [12] DE0-NANO manual de usuario, Terasic (2015).
- [13] FPGA CPLD and ASIC from Altera, Disponible en: <http://www.altera.com/>
- [14] Torres M, Diseño e ingeniería electrónica asistida con protel dxp. México: ALFAOMEGA, (2005).
- [15] David M Pozar, Microwave Engineering, Cuarta Edición, Editorial Jhon Wiley and sons, Inc, cuarta edición, (2012).
- [16] Ultra CAD Desing, Inc's, Service Notes. PCB Impedance Control.
- [17] Rodolfo Neri Vela, Líneas de transmisión, Editorial Mc Grall-Hill, (1999).
- [18] Hoja de datos TH5651A, Texas Instruments, Convertidor digital analógico de 10 bits, obtenido de: <http://www.ti.com/lit/ds/symlink/th5651a.pdf>.
- [19] OPA 659, Hoja de datos, Texas Instrumentes, amplificador operacional, obtenido de: <http://www.ti.com/lit/ds/symlink/opa659.pdf>.
- [20] D. Ahn, et al., "A Design of the Low-Pass Filter Using the Novel Microstrip Defected Ground Structure", IEEE Transactions on microwave theory and techniques, Vol. 49, No. 1, enero (2001).
- [21] R. Pallas, *Adquisición y distribución de señales*. México: Marcombo, (2005).

[22] Javier Ruiseco López, "Control de un robot articular de 1 grado de libertad utilizando un sistema embebido vía WiFi"; Tesis de Maestría en Ciencias de la Electrónica opción Automatización, FCE-BUAP, (2012).

[23] Reese Robert, Microcontrollers From Assembly Language to C Using the PIC24, Charles River Media, Pag.284, (2005).

[24] De Gante R. A. (2016). Diseño y construcción de un sistema de adquisición de datos. México: Tesis Maestría. Benemérita Universidad Autónoma de Puebla.

[25] Alonso, J. Electrónica Digital y microprocesadores, Ediciones de la Universidad de Oviedo, (2015).

[26] Consejos para depurar sistemas digitales, con osciloscopios mixtos. Pág. 45. Obtenido de : [http://www.redeweb.com/\\_txt/656\\_657/42.pdf](http://www.redeweb.com/_txt/656_657/42.pdf). Pág.4

[27] Cecilio Blanco Viejo, Electrónica Digital, 46-48. Ediciones de la Universidad de Oviedo, (2003).

[28] San Miguel, P," Electrónica aplicada", Paraninfo 2ª edición, (2016).