

ESCUELA POLITÉCNICA NACIONAL

FACULTAD DE INGENIERÍA ELÉCTRICA Y ELECTRÓNICA

CREACIÓN DE BLOQUE DE PROPIEDAD INTELECTUAL E INTEGRACIÓN CON INTERFAZ DE USUARIO EN SISTEMA EN CHIP ZYBO PARA APLICACIÓN SPWM DE ÍNDICE DE MODULACIÓN Y FRECUENCIA VARIABLE

**TRABAJO DE TITULACIÓN PREVIO A LA OBTENCIÓN DEL TÍTULO DE
INGENIERO EN ELECTRÓNICA Y CONTROL**

ENRIQUE ALEJANDRO TOSCANO HERNÁNDEZ

enrique.toscano@epn.edu.ec

DIRECTOR: ING. PATRICIO IVÁN CHICO HIDALGO, MSc.

patricio.chico@epn.edu.ec

Quito, agosto 2019

AVAL

Certifico que el presente trabajo fue desarrollado por Enrique Alejandro Toscano Hernández, bajo mi supervisión.

ING. PATRICIO IVÁN CHICO HIDALGO, MSc.
DIRECTOR DEL TRABAJO DE TITULACIÓN

DECLARACIÓN DE AUTORÍA

Yo, Enrique Alejandro Toscano Hernández, declaro bajo juramento que el trabajo aquí descrito es de mi autoría; que no ha sido previamente presentado para ningún grado o calificación profesional; y, que he consultado las referencias bibliográficas que se incluyen en este documento.

A través de la presente declaración dejo constancia de que la Escuela Politécnica Nacional podrá hacer uso del presente trabajo según los términos estipulados en la Ley, Reglamentos y Normas vigentes.

ENRIQUE ALEJANDRO TOSCANO HERNÁNDEZ

DEDICATORIA

Este trabajo se lo dedico a toda mi familia, padres y hermana, por haberme apoyado para estudiar esta carrera universitaria.

AGRADECIMIENTO

Expreso un gran agradecimiento a Dios por prodigarme bendiciones y bienestar, a mis padres por permitirme tener una profesión, su ejemplo, esfuerzo y apoyo, han sido la base sobre la cual me he erguido como persona y estudiante.

A la Escuela Politécnica Nacional y a su planta de docentes, por formarme profesionalmente mediante la transmisión del conocimiento académico, teórico y práctico, con conciencia ética y crítica, para contribuir al bienestar de la comunidad y al desarrollo del país.

Al director del presente trabajo de titulación, Ing. Patricio Chico, MSc, por acompañarme, bajo su guía académica y profesional, en el desarrollo de este trabajo de investigación, previo a la obtención del título de Ingeniero en Electrónica y Control.

Al Dr.-Ing. Marcelo Pozo, Jefe del Laboratorio de Electrónica de Potencia, Control de Máquinas y Redes Eléctricas Inteligentes de la Escuela Politécnica Nacional, por su gentileza en autorizarme el uso de las instalaciones e instrumentos para validar este proyecto de titulación.

A los señores Jefe, profesores, asistente de investigación y tesis del Laboratorio de Electrónica de Potencia, Control de Máquinas y Redes Eléctricas Inteligentes de la Escuela Politécnica Nacional, por compartir generosamente sus conocimientos y experiencias en beneficio de mis capacidades y competencias profesionales.

A Ana Belén, por compartir gratos momentos a lo largo de este trayecto universitario.

ÍNDICE DE CONTENIDO

AVAL	I
DECLARACIÓN DE AUTORÍA.....	II
DEDICATORIA.....	III
AGRADECIMIENTO.....	IV
ÍNDICE DE CONTENIDO.....	V
RESUMEN	VII
ABSTRACT	VIII
1. INTRODUCCIÓN.....	1
1.1. OBJETIVOS.....	2
1.2. ALCANCE	3
1.3. MARCO TEÓRICO.....	4
1.3.1. SISTEMA EN CHIP ZYNQ [4] [5].....	4
1.3.2. BLOQUES DE PROPIEDAD INTELECTUAL [4].....	7
1.3.3. INVERSORES TRIFÁSICOS.....	8
1.3.4. MODULACIÓN POR ANCHO DE PULSO SENOIDAL (SPWM) [7] [8].....	8
1.3.5. ACCIONAMIENTO DE MOTOR DE INDUCCIÓN ALIMENTADO POR INVERSOR TRIFÁSICO [11] [12] [7].....	11
2. METODOLOGÍA.....	17
2.1. ZYBO [14]	18
2.1.1. OPTOACOPLAMIENTO ENTRE ZYBO Y <i>GATE-DRIVERS</i>	20
2.2. VIVADO DESING SUITE	22
2.3. CONFIGURACIÓN DE LA PARTE LÓGICA PROGRAMABLE	24
2.3.1. PROCESO DE PERÍODO DE MUESTREO DE LAS ONDAS MODULANTES.....	26
2.3.2. PROCESO DE LA CONSTRUCCIÓN DE LA ONDA PORTADORA	28
2.3.3. PROCESOS DE COMPARACIÓN DE LA ONDA PORTADORA CON LAS ONDAS MODULANTES	29
2.3.4. PROCESOS DE GENERACIÓN DE TIEMPOS MUERTOS ASIMÉTRICOS PARA SPWM	31
2.3.5. PROCESO DE ACTUALIZACIÓN SPWM.....	32

2.3.6.	CONFIGURACIÓN DE PINES DE SALIDA.....	32
2.3.7.	BLOQUE DE PROPIEDAD INTELECTUAL PARA APLICACIÓN DE SPWM.....	33
2.4.	PROGRAMACIÓN DEL SISTEMA DE PROCESAMIENTO	37
2.4.1.	ALGORITMO DE EVALUACIÓN DE NÚMERO PRIMO.....	42
2.4.2.	ALGORITMO DE DESCOMPOSICIÓN EN FACTORES PRIMOS...	43
2.4.3.	ALGORITMO PARA REDUCIR EL ERROR DEL PERIODO DE MUESTREO DE LA ONDA PORTADORA	44
2.5.	INTERFAZ GRÁFICA.....	46
2.6.	DIAGRAMAS DE FLUJO	47
3.	RESULTADOS Y DISCUSIÓN	57
3.1.	REPORTES DE IMPLEMENTACIÓN DE VIVADO DESIGN SUITE.....	57
3.1.1.	REPORTE DE TIEMPO [23] [24] [25].....	57
3.1.2.	REPORTE DE UTILIZACIÓN	64
3.1.3.	REPORTE DE POTENCIA	65
3.2.	SIMULACIÓN DE LAS SEÑALES DEL BLOQUE DE PROPIEDAD INTELECTUAL SPWM.....	66
3.3.	PRUEBAS DE LABORATORIO	73
3.3.1.	PRUEBAS DIRECTAMENTE EN LAS SALIDAS DE ZYBO.....	77
3.3.2.	PRUEBAS CON MOTOR DE INDUCCIÓN	79
3.3.3.	PRUEBAS CON CARGA TRIFÁSICA BALANCEADA RESISTIVA-INDUCTIVA EN CONEXIÓN Y	87
3.3.4.	PRUEBAS CON CARGA TRIFÁSICA BALANCEADA RESISTIVA EN CONEXIÓN Y	89
4.	CONCLUSIONES Y RECOMENDACIONES.....	91
4.1.	CONCLUSIONES.....	91
4.2.	RECOMENDACIONES	92
5.	REFERENCIAS BIBLIOGRÁFICAS	94
	ANEXOS	97

RESUMEN

Este trabajo de titulación crea un bloque de propiedad intelectual (IP) con el objeto de desempeñar aplicaciones SPWM de índice de modulación y frecuencia variable en el sistema en chip ZYBO. ZYBO puede realizar múltiples funciones, concurrentemente, debido a que dispone de un arreglo de compuertas lógicas programables en campo (FPGA) y de un sistema de procesamiento. En ZYBO se considera como maestro de las comunicaciones al sistema de procesamiento, el cual está conformado principalmente por procesadores ARM.

El sistema en chip completo, FPGA y el sistema de procesamiento, entrega señales SPWM que se utilizan para excitar un inversor trifásico. La modulación SPWM se la configura en VHDL y empaqueta en el bloque IP. El ARM actualiza las condiciones de trabajo del bloque IP usando protocolos AXI (*Advanced Extensible Interface*) y, además, interactúa con el usuario que ingresa los datos al sistema, vía comunicación serial, usando una interfaz gráfica.

Finalmente, mediante la simulación del bloque IP y experimentación con un sistema ZYBO-inversor, se realiza el análisis y validación de los resultados obtenidos al alimentar las siguientes cargas trifásicas: motor de inducción, resistencias y resistencias-inductores.

Se plantea este trabajo como alternativa a la utilización de sistemas microcontroladores clásicos que tienen la desventaja de ejecutar una instrucción a la vez. La utilización del sistema en chip ZYBO permitirá al Laboratorio de Electrónica de Potencia, Control de Máquinas y Redes Eléctricas Inteligentes de la Escuela Politécnica Nacional formar una biblioteca de bloques IP para futuras investigaciones en el campo de electrónica de potencia.

PALABRAS CLAVE: SPWM, sistema en chip, ZYBO, bloque IP, motor de inducción trifásico

ABSTRACT

This project creates an intellectual property (IP) block with the purpose of performing SPWM applications of modulation index and variable frequency on system on chip ZYBO. ZYBO can perform multiple functions concurrently, because it has a field programmable gate array (FPGA) and a processing system. The master of all communications in ZYBO is the processing system, which is mainly made up of ARM processors.

The entire system on chip, FPGA and processing system, delivers SPWM signals that are used to drive a three-phase inverter. The SPWM modulation is configured in VHDL and packaged in the intellectual property block. The ARM updates the working conditions of the IP block using AXI (Advanced Extensible Interface) protocols and interacts with the user who enters the data into the system, via serial communication, using a graphical interface.

Finally, through simulation of the IP block and experimentation with a ZYBO-inverter system, the analysis and validation of the results obtained when feeding the following three-phase loads, induction motor, resistors and resistors-inductors are performed.

This project is proposed as an alternative to the use of classic microcontroller systems that have the disadvantage of executing one instruction at a time. The use of system on chip ZYBO will enable Laboratorio de Electrónica de Potencia, Control de Máquinas y Redes Eléctricas Inteligentes of Escuela Politécnica Nacional to form a library of IP blocks for future research in the field of power electronics.

KEYWORDS: SPWM, system on chip, ZYBO, IP block, three-phase induction motor

1. INTRODUCCIÓN

En los sistemas microprocesados se puede ejecutar una acción a la vez, limitando la aplicación al tiempo de ejecución de cada instrucción programada en el procesador. Cuando se desea realizar múltiples funciones críticas de control o integrar un sistema adicional como, por ejemplo, la interacción con los usuarios, se debe trabajar con un adecuado manejo de las actividades principales y secundarias en el mismo circuito integrado, prolongando paulatinamente el tiempo de ejecución de las instrucciones del procesador hasta el punto en que no se logra cumplir con los tiempos requeridos para realizar el control principal de la aplicación, ocasionando que no sea viable utilizar un único sistema microprocesado para realizar la aplicación completa.

Una solución a la limitante de tiempos de ejecución de instrucciones en los sistemas microprocesados es implementar la aplicación de software en un Sistema en chip (SOC- *System on Chip*) Zynq de Xilinx, que tiene integrado un arreglo de compuertas lógicas programables en campo (FPGA-*Field Programmable Gate Array*) y procesadores ARM, con los cuales se explotan las capacidades de trabajar con múltiples procesos, críticos y secundarios, al mismo tiempo.

Además, los SOCs, al integrar la versatilidad de las unidades centrales de procesamiento (CPU- *Central processing unit*) y el poder de procesamiento de la FPGA, permiten integrar, por separado y al mismo tiempo, funcionalidades de interfaz de usuario, control supervisor, administración de sistema; así como, algoritmos de control, redes, etc.

En consecuencia, el mismo empaquetamiento del SOC es utilizado para ejecutar acciones de software y realizar el algoritmo de control que interactúa con hardware, de esta manera, al tener toda la aplicación en un mismo circuito integrado, se obtiene un sistema más confiable y de mejor desempeño [1].

El Laboratorio de Electrónica de Potencia, Control de Máquinas y Redes Eléctricas Inteligentes de la Escuela Politécnica Nacional ha adquirido tarjetas ZYBO (*Zynq board*) de Digilent, socio de Xilinx, que poseen arreglos de compuertas lógicas programables en campo (FPGA) equivalente a Artix-7, y procesadores dual-core ARM Cortex-A9, dentro de su Sistema en Chip, para realizar distintos tipos de control en el área de máquinas eléctricas.

Este trabajo de titulación tiene el objetivo de crear un bloque de propiedad intelectual para aplicaciones de modulación por ancho de pulso senoidal (SPWM - *Sinusoidal pulse width*

modulation) de índice de modulación y frecuencia variable, que será integrado con una interfaz de usuario, en el mismo sistema en chip ZYBO; y, de formar una base para crear una biblioteca de bloques de propiedad intelectual para futuras investigaciones en el campo de electrónica de potencia y control de máquinas.

La técnica de modulación SPWM se realiza en la FPGA y permite variar la velocidad y el voltaje, así como, reducir el contenido armónico de las corrientes en los motores de inducción, con resultados satisfactorios. Este tipo de modulación se beneficia de las características de las FPGAs, tales como: alta velocidad debido a su paralelismo, bajo consumo energético, facilidad de reconfiguración de la plataforma para revisar y ajustar algoritmos [2]; y, de su amplio potencial para aplicaciones de electrónica de potencia, debido a su mayor robustez y concurrencia [3]. Además, se hará uso del procesador ARM incluido en el SOC para interactuar con el usuario o integrar nuevos algoritmos usando otros bloques IP para trabajos futuros.

Para facilitar la comprensión de las siglas utilizadas en el presente trabajo, al final del documento se anexa una lista de abreviaturas con sus respectivos significados.

Con el propósito de poner en funcionamiento el sistema en chip ZYBO se anexa una guía de usuario con los procedimientos para: programar y utilizar la aplicación de ZYBO; simular el voltaje medio cuadrático entre líneas de un inversor trifásico excitado por la modulación SPWM de ZYBO; y, modificar el programa del sistema de procesamiento.

Con el fin de contribuir en el ámbito académico y a futuras investigaciones, la aplicación desarrollada en el SOC se entregará al Laboratorio de Electrónica de Potencia, Control de Máquinas y Redes Eléctricas Inteligentes de la Escuela Politécnica Nacional para utilizar los algoritmos y bloque de propiedad intelectual en las tarjetas ZYBO.

Los próximos trabajos de investigación que se desarrollen sobre la base de este proyecto de titulación, tendrán la capacidad de agregar más procesos a la FPGA sin comprometer los procesos originales debido al paralelismo, con la ventaja de poder aprovechar los algoritmos del software y sus versiones, independientemente de la tarjeta utilizada.

1.1. OBJETIVOS

El objetivo general de este proyecto técnico es:

Crear un bloque de propiedad intelectual e integrar con interfaz de usuario en sistema en chip ZYBO para aplicación SPWM de índice de modulación y frecuencia variable.

Los objetivos específicos del proyecto técnico son:

- Realizar la revisión bibliográfica de técnicas de modulación SPWM orientada a motores trifásicos de inducción, así como de la programación de la interfaz AXI y el procesador ARM en Zynq-7000 ARM/FPGA SoC Trainer Boards.
- Desarrollar la programación del SOC para generar señales SPWM de índice de modulación y frecuencia variables que permitan excitar un inversor trifásico.
- Simular las señales de la FPGA utilizando el software Vivado de Xilinx.
- Validar la modulación SPWM en un inversor trifásico con un motor de inducción de baja potencia.

1.2. ALCANCE

- Realizar la revisión bibliográfica de técnicas de modulación SPWM para variar la velocidad de motores de inducción.
- Realizar la revisión bibliográfica de la programación de la interfaz AXI (*advanced extensible interface*) y el procesador ARM en la tarjeta ZYBO Zynq-7000 ARM/FPGA SoC Trainer Boards con la finalidad de integrar un sistema procesador-FPGA.
- Desarrollar la configuración en VHDL de la FPGA que corresponde a la parte lógica programable (*PL-programmable logic*) del SOC.
- Crear un bloque IP del código VHDL desarrollado para la FPGA.
- Desarrollar la programación en lenguaje C del procesador ARM que corresponde a la parte del sistema de procesamiento (*PS-processing system*) del SOC.
- Integrar el PS y PL del SOC mediante la interfaz AXI, de tal manera que el sistema en conjunto entregue SPWMs con índice de modulación y frecuencia variables.
- Simular las señales de salida de la FPGA del SOC en el software Vivado de Xilinx.
- Validar la modulación SPWM proporcionada por el SOC en un inversor trifásico variando el índice de modulación y frecuencia de salida y verificando el voltaje RMS entre líneas y la frecuencia en la carga respectivamente.

1.3. MARCO TEÓRICO

Esta sección presenta una introducción a los temas que se desarrollan en el presente proyecto técnico con el objeto de permitir al lector comprender la técnica utilizada para excitar motores de inducción mediante inversores trifásicos; y, las propiedades y beneficios de los Sistemas en Chip Zynq.

1.3.1. SISTEMA EN CHIP ZYNQ [4] [5]

El sistema en chip (SOC- *System on chip*) Zynq de Xilinx es, esencialmente, la combinación de un sistema microprocesado con una estructura de arreglos de compuertas lógicas programables en campo (FPGA- *Field Programmable Gate Array*) en un solo chip. Esta característica permite combinar capacidades de programación de software y hardware en un mismo integrado, lo que implica menor costo, transferencias de datos de una manera más segura y rápida, reducción del espacio físico, mejor confiabilidad y ahorro de consumo energético, debido a que el sistema completo se encuentra en el mismo lugar.

Zynq, específicamente, combina un procesador dual-core o single-core (dependiendo del modelo) ARM Cortex-A9 con una estructura FPGA basada en la arquitectura 7-Series. Además, posee una interfaz AXI (*Advanced extensible interface*) que une las dos partes del SOC, proveyendo un alto ancho de banda y presentando una baja latencia en conexiones.

Los SOCs Zynq están integrados por dos secciones, la lógica programable (PL- *Programmable logic*) y el sistema de procesamiento (PS- *Processing system*). La Figura 1.1. presenta un diagrama de la arquitectura Zynq, donde el CPU o procesador ARM es la pieza principal del PS y la PL está esencialmente compuesta por una estructura FPGA.

El PS está integrado por los siguientes bloques centrales: unidad de procesamiento de aplicación, interfaces de periféricos, memoria cache, interfaces de memoria, interconexiones y circuito de generación de reloj [4] [5]. Es de interés citar que el SOC Zynq utiliza la revisión r3p0 del procesador ARM Cortex-A9 con base en la arquitectura ARM v7-A.

La PL de Zynq mostrada en la Figura 1.2. está constituida por una estructura FPGA, bloques de entrada y salida (IOB-*Input/output block*), bloques de memoria de acceso aleatorio (RAM - *Random access memory*) para requerimientos de memoria extensos, porciones de procesamiento digital de señales (DSP-*Digital signal processing*) para operaciones aritméticas que requieren altas velocidades dedicadas a señales de longitudes

de palabras extensas, dos convertidores análogo-digital, relojes y un set de puerto JTAG (Joint test action group).

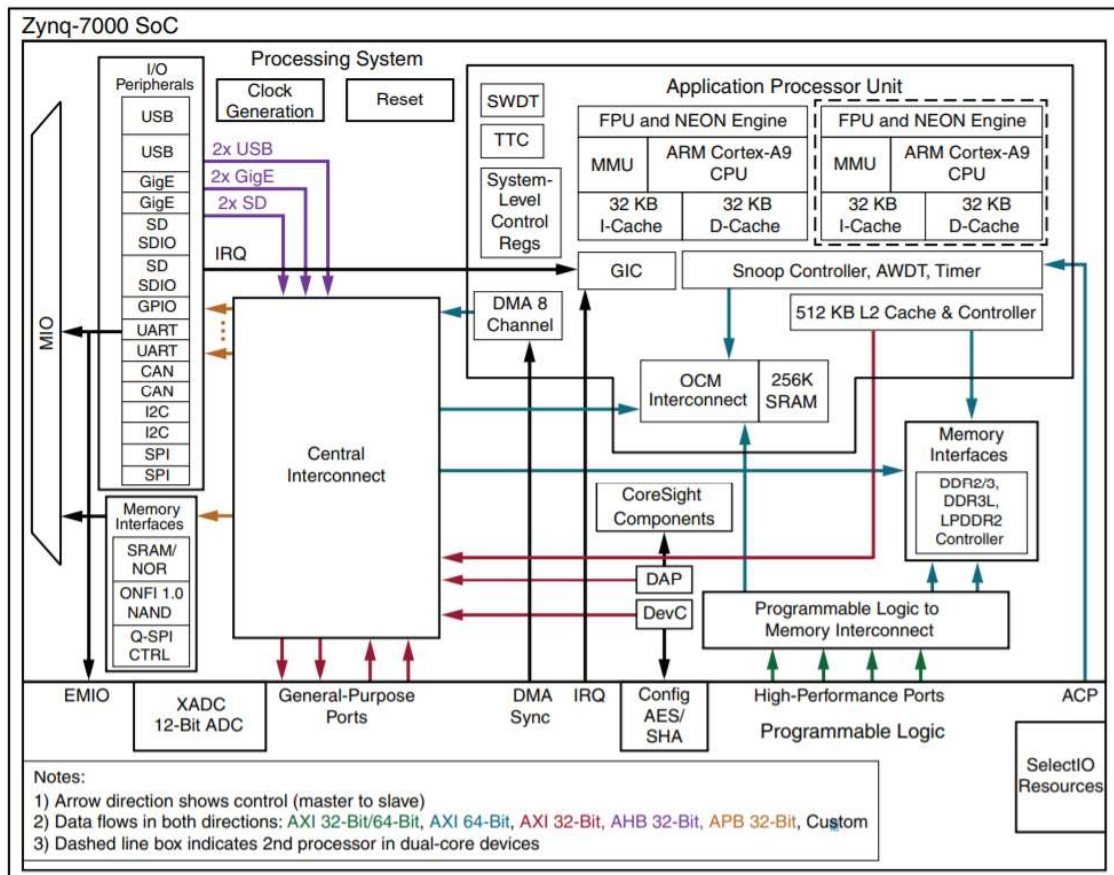


Figura 1.1. Arquitectura del SOC Zynq [5]

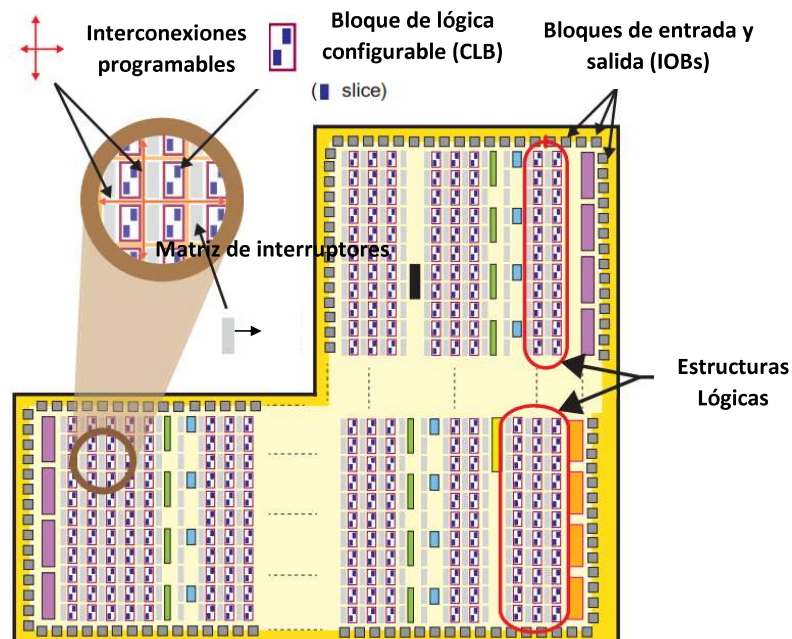


Figura 1.2. Elementos principales de la estructura lógica programable [4]

La FPGA está constituida por bloques de lógica configurable (CLB-Configurable logic block) que están conectados a otros recursos de la PL a través de interconexiones programables. La estructura de los CLBs de Zynq se indica en la Figura 1.3., donde se observa que presenta dos *slices* (porciones), cada uno con cuatro tablas de búsqueda (LUT-Lookup table) y ocho *flip-flops* (FF). Externamente y junto a los CLBs, se encuentran arreglos de interruptores que permiten realizar conexiones dentro y fuera del CLB; además, presenta entradas y salidas de *carry* para interconectar *slices*.

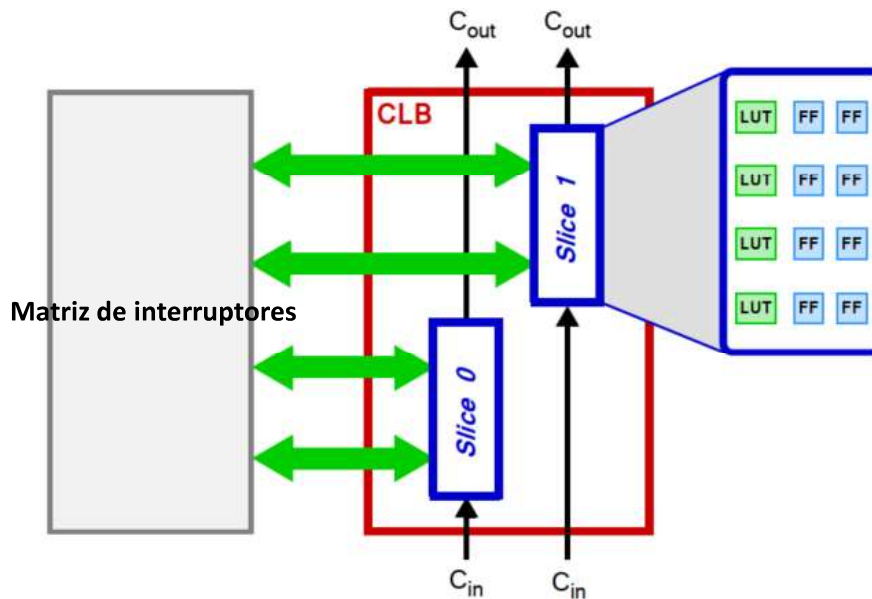


Figura 1.3. Estructura de un bloque de lógica configurable [4]

Los *slices* realizan funciones secuenciales y combinatorias con base en circuitos digitales. Es así como los LUTs pueden formar tablas de verdad de hasta 6 entradas y en conjunto con los FFs pueden prestar funciones combinatoriales y secuenciales, respectivamente.

Zynq dispone de interfaces e interconexiones AXI que permiten integrar la PL y el PS. Se entiende por interfaces a la comunicación punto a punto de mensajes en base a protocolos; y, por interconexiones a los distintos interruptores que permiten conectar diferentes interfaces con la finalidad de construir un camino para el tráfico de datos. AXI presenta tres tipos de protocolos: AXI4 (Full), AXI4-Lite y AXI4-Stream, cuyas características están detalladas en la siguiente tabla:

Tabla 1.1. Características de los protocolos AXI [4]

AXI4 (Full)	AXI4-Lite	AXI4-Stream
Mapeado en memoria	Mapeado en memoria	Sin mapeado en memoria
Transferencia de datos por ráfagas (<i>Burst</i>)	Transferencia de un dato por conexión	Transferencia de datos por ráfagas (<i>Burst</i>)
Hasta 256 palabras	Palabra por palabra	Sin restricción de tamaño

Mapeado en memoria significa que el tipo de protocolo requiere que el maestro especifique direcciones de memoria para realizar cualquier operación de transferencia de datos. El protocolo AXI4-Stream no utiliza direcciones de memoria por lo que la comunicación es directa entre el destino y la fuente.

A las interfaces AXI encargadas de unificar el PS y la PL se las puede observar en la Figura 1.1. por medio de los puertos de propósito general, de alto desempeño y el puerto de aceleración coherente (*ACP-Accelerator coherency port*). Las interfaces de propósito general y de alto desempeño se conectan con el bloque de interconexión central, a diferencia del ACP que se conecta directamente a la unidad de procesamiento de aplicaciones (*APU-Application processing unit*).

1.3.2. BLOQUES DE PROPIEDAD INTELECTUAL [4]

Un bloque de propiedad intelectual (*IP-Intellectual property*) es un componente que describe al hardware que se incorporará en un diseño. Existen dos tipos de bloques IP, los *soft-IP blocks* y los *hard-IP blocks*: los *soft-IP blocks* se caracterizan por permitir al usuario final modificar ciertas funcionalidades antes de sintetizar o implementar la aplicación; a diferencia de los *hard-IP blocks*, los cuales ya han pasado por el proceso de síntesis e implementación.

A continuación, se describen los diferentes métodos de entrega de los bloques IP:

- *Soft-IP blocks*
 - RTL sintetizable: El registro a nivel de transferencia (*RTL-Register transfer level*) es una abstracción de circuitos digitales, por esta razón, este tipo de entrega de IP es el más modificable ya que el usuario puede realizar cambios del lenguaje de descripción de hardware (*HDL-Hardware description language*), como algoritmos con lógica booleana, sentencias condicionales o secuencias de eventos, antes de sintetizar e implementar.
 - *Netlist* a nivel de compuerta: *Netlist* es el producto principal de la síntesis y contiene las conexiones entre los elementos del CLB y bloques RAM. Este entregable aún puede ser modificado para su implementación, sin embargo, ya ha pasado por un proceso parcial de síntesis [6].
- *Hard-IP blocks*
 - Firm IP: Orientada a usos para FPGAs y circuitos integrados de aplicación específica (*ASIC-Application specific integrated circuit*), esta presentación

está sintetizada e implementada, por este motivo, se restringe su empleo a dispositivos específicos o pertenecientes a una misma familia.

- Diseño de transistor: Este bloque IP está listo para producción en chips semiconductores, por lo tanto, no es aplicable para FPGAs.

1.3.3. INVERSORES TRIFÁSICOS

Los inversores son convertidores estáticos de energía que permiten la transformación de voltajes DC a AC por medio de un arreglo de interruptores de potencia. La Figura 1.4. muestra un arreglo totalmente controlado de tres ramales de transistores bipolares de compuerta aislada (IGBTs - *Insulated-gate bipolar transistors*) con sus respectivos diodos internos, la integración de este esquema con un método de control de inversión se conoce como inversor trifásico. El presente circuito está clasificado como inversor de fuente de voltaje, ya que una fuente de voltaje (V_{bus}) alimenta a la etapa de inversión. La fuente utilizada en el diagrama hace referencia al voltaje DC obtenido por una etapa de rectificación con filtrado, conocido como el voltaje del bus DC. Las salidas a, b y c identifican las fases trifásicas generadas. La topología descrita es una de las más conocidas y usadas, y puede ser implementada usando IGBTs, transistores de efecto de campo de metal-óxido-semiconductor (MOSFETs-*Metal-oxide semiconductor field effect transistors*) u otros interruptores controlados.

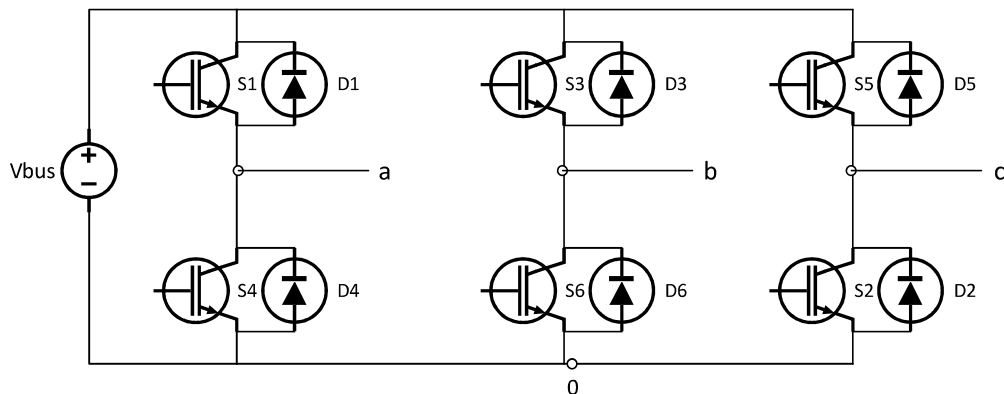


Figura 1.4. Inversor trifásico

1.3.4. MODULACIÓN POR ANCHO DE PULSO SENOIDAL (SPWM) [7] [8]

El objetivo de esta modulación es obtener una alimentación AC con características controlables para las cargas sobre la base de una entrada DC. La modulación es realizada al comparar ondas modulantes con una portadora, en este caso, ondas sinusoidales desfasadas 120 grados entre sí con dientes de sierra o triangulares, respectivamente,

como se observa en la Figura 1.6. La frecuencia de la onda portadora, la cual es aplicada idealmente a los interruptores del inversor trifásico (Figura 1.4.) para su conmutación, se caracteriza por ser mayor a la de las modulantes. Mientras que, el periodo de las ondas modulantes se refleja en voltajes y corrientes en las salidas del inversor, y representa, además, la frecuencia fundamental en la carga.

Esta técnica define dos relaciones fundamentales, el índice de modulación y la relación de modulación de frecuencia, de acuerdo con la Ecuación 1.1. y Ecuación 1.2., respetivamente. Siendo el índice de modulación la relación entre las amplitudes de la onda modulante y portadora; y, la relación de modulación de frecuencia, la relación de frecuencias entre las ondas portadora y modulante.

$$Im = \frac{A_m}{A_p} \quad (1.1)$$

$$mf = \frac{f_p}{f_m} \quad (1.2)$$

En la Figura 1.5. se presentan los accionamientos que se realizan en los interruptores del inversor al aplicar la modulación SPWM. Las señales R, S y T son las ondas modulantes que se comparan con una onda portadora triangular. Cuando la modulante es mayor que la portadora se obtiene 1L a la salida de cada comparador, y cuando es menor, se consigue 0L; estas señales controlan la conmutación de los interruptores superiores de cada ramal (S1, S3, S5), mientras que los inferiores (S4, S6, S2) son excitados por las señales negadas de cada comparación, tal como se muestra en el siguiente esquema de conexión:

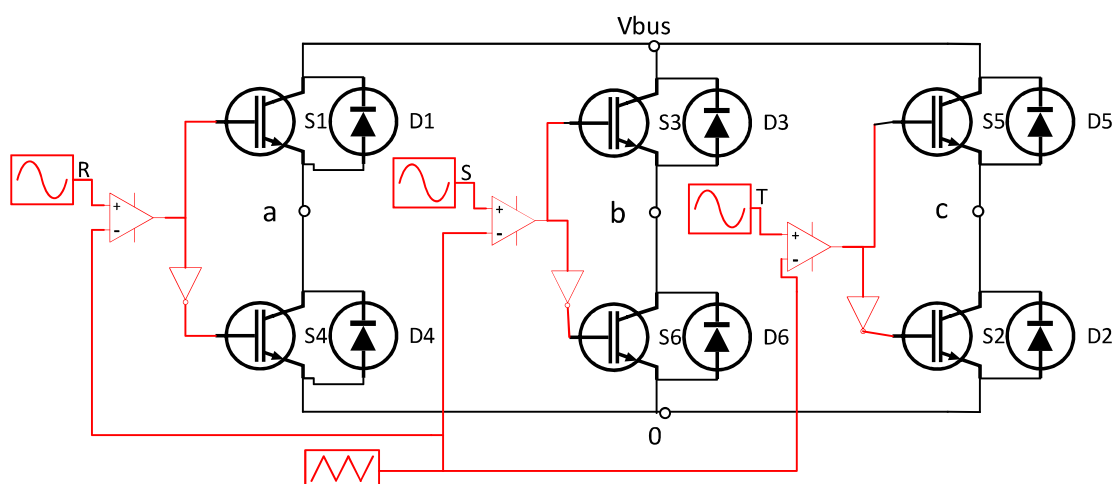


Figura 1.5. Modulación SPWM aplicada al inversor

Cuando las señales de control provocan que se conmuten los ramales, se obtienen ondas de voltaje de dos niveles en los puntos intermedios de cada interruptor (V_{a0} , V_{b0} , V_{c0}), como se muestra en la Figura 1.6. Además, se observa que el voltaje entre líneas (V_{ab}) presenta 3 niveles: el voltaje del bus dc (V_{bus}), cero y $-V_{bus}$. A partir de las líneas a, b y c del inversor se obtienen voltajes de alimentación AC para la carga.

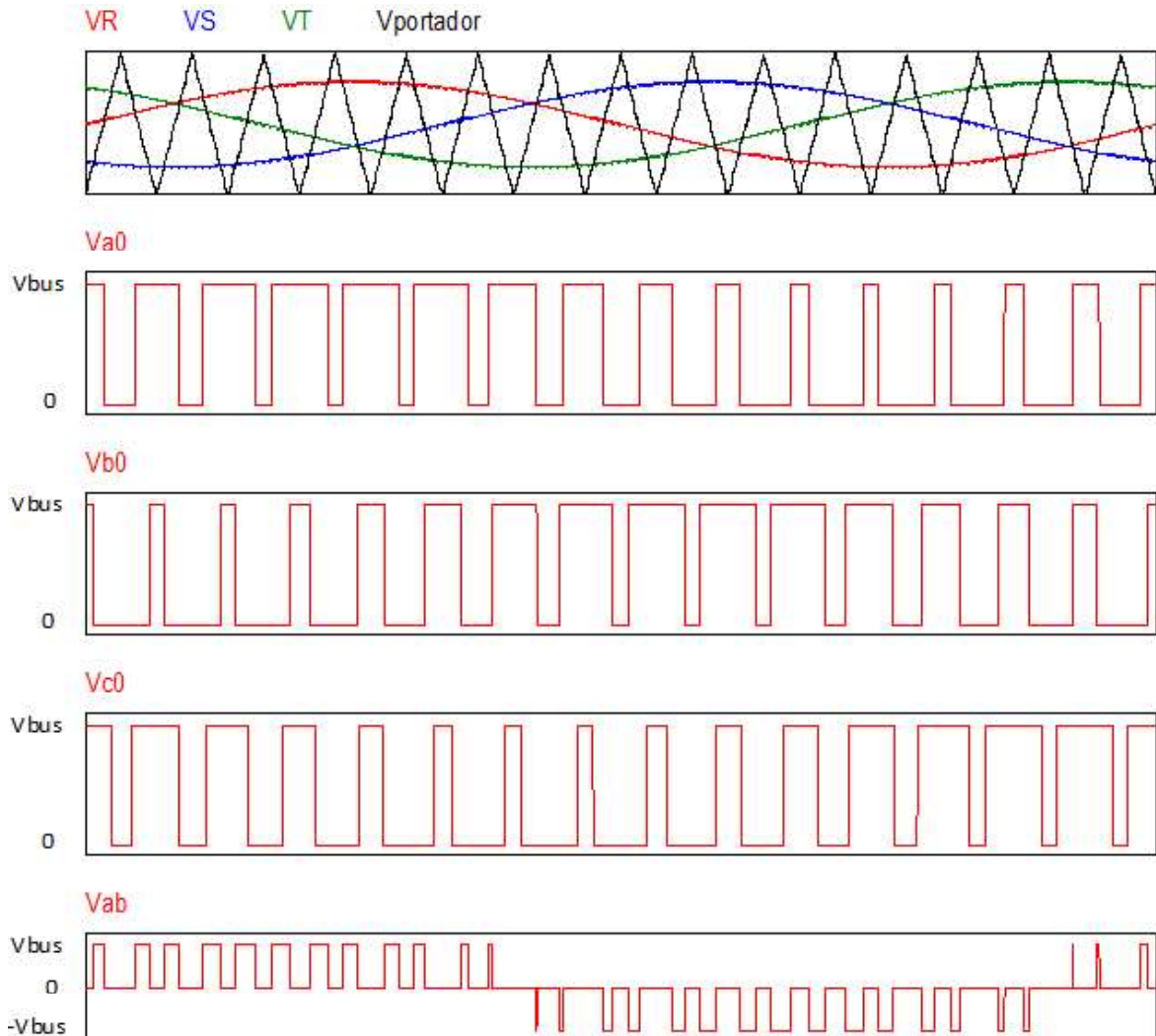


Figura 1.6. Formas de onda de voltaje SPWM

Para variar el voltaje medio cuadrático (RMS- *Root mean square*) de la fundamental entre líneas, se realizan cambios en el índice de modulación con resultados lineales cuando este no sobrepasa el valor de 1 ($m \leq 1$), este rango es conocido como la región lineal. Adicionalmente, si el índice de modulación supera 1, se ingresa a una zona de sobremodulación, donde la fundamental de voltaje varía en función de m de una manera no lineal. La zona de sobremodulación permite incrementar el voltaje de salida del inversor y produce un aumento de armónicos de bandas laterales en comparación con la región

lineal; dependiendo de la carga, la sobremodulación puede tener menores pérdidas que en la región lineal, debido a la amplitud de armónicos dominantes.

Las Ecuaciones, 1.3 y 1.4, proporcionan el voltaje RMS entre líneas y su respectiva fundamental, únicamente para la región lineal ($Im \leq 1$):

$$V_{ab} = \sqrt{\frac{2 Im}{\pi}} V_{bus} \quad (1.3) [9]$$

$$V_{ab1} = \frac{\sqrt{3}}{2\sqrt{2}} Im V_{bus} \quad (1.4) [8]$$

Existen modulaciones SPWM síncronas y asíncronas. La síncrona se caracteriza por tener una relación entre frecuencias portadora y modulante entera, lo que implica que la portadora y modulante iniciarán en el mismo instante; la asíncrona, por el contrario, posee una relación que no es entera. La asincronía de la modulación SPWM genera la presencia indeseable de subarmónicos de la frecuencia fundamental [8].

Es recomendable utilizar una relación de modulación de frecuencia impar múltiplo de 3. Al realizar esta selección, se eliminan todos los armónicos múltiplos de 3 en el voltaje entre líneas; además, por ser impar, se puede observar que los armónicos pares en los voltajes de fase son eliminados y, adicionalmente, los voltajes en las fases entre líneas del inversor mantienen una simetría en los patrones de las formas de onda trifásicas [7], [10].

El voltaje entre líneas del inversor presenta armónicos en las bandas laterales de la frecuencia de conmutación y sus múltiplos, según la Ecuación 1.5., donde h representa el orden del armónico, j y k son combinaciones de enteros. Cuando j es impar los valores de k son pares; y, cuando j es par, entonces k adquiere valores impares. El orden de los armónicos presenta una restricción, los armónicos múltiplos de 3 no están presentes.

$$h = j(mf) \pm k \quad (1.5)$$

1.3.5. ACCIONAMIENTO DE MOTOR DE INDUCCIÓN ALIMENTADO POR INVERSOR TRIFÁSICO [11] [12] [7]

Al alimentar a los devanados del estator con voltajes trifásicos se producen los siguientes fenómenos físicos:

- Se originan corrientes en cada fase del estator. Al tener una alimentación AC las corrientes cambian de polaridad.
- Las corrientes variables AC en el estator generan un campo magnético giratorio.

- El campo magnético giratorio provoca la inducción de fuerzas electromotrices (*emf-electromotive force*) en el rotor.
- Las emfs causan que se generen corrientes en el rotor.
- La interacción de las corrientes del rotor y el campo magnético del estator producen una fuerza electromagnética que ocasiona que el rotor gire.

La velocidad de rotación del campo magnético giratorio del estator es la velocidad síncrona, sin embargo, el rotor gira a una velocidad inferior debido a que cuando ambas velocidades se igualan ya no existen corrientes en el rotor, de esta manera el rotor desacelera.

La dirección del flujo magnético del estator puede ser modificada según la configuración de los interruptores del inversor. De esta manera el rotor seguirá el flujo, controlando así, el movimiento del motor. La Figura 1.7., elaborada en base a [12], presenta los 8 estados válidos para la SPWM en inversores trifásicos; se muestra:

- Con color rojo las activaciones de los interruptores (S_n).
- Con flechas de color azul, la dirección de las corrientes en las fases del estator del motor.
- Con una flecha negra, la dirección del flujo magnético en el estator.
- Con color verde, la activación de alguno de los diodos internos (D_n) de los interruptores.

Los interruptores (S_n) y sus diodos internos (D_n) conducen de manera unidireccional, siguiendo la dirección de las flechas en sus esquemáticos, por esta razón, se observa la dirección que adquieren las corrientes de las fases para los estados del 1 al 6. Las direcciones de las corrientes en los estados 7 y 8, por presentar características inductivas, mantienen la misma dirección que su estado anterior, activando algún diodo para mantener la circulación ininterrumpida de corriente. Además, se observa que los dos últimos estados, 7 y 8, no tienen una alimentación del bus dc, motivo por el cual no se crea flujo magnético. Finalmente, se debe mencionar que las direcciones de las corrientes para las condiciones 7 y 8 no se muestran debido a que no tienen una orientación general, pero sí circulan en los devanados.

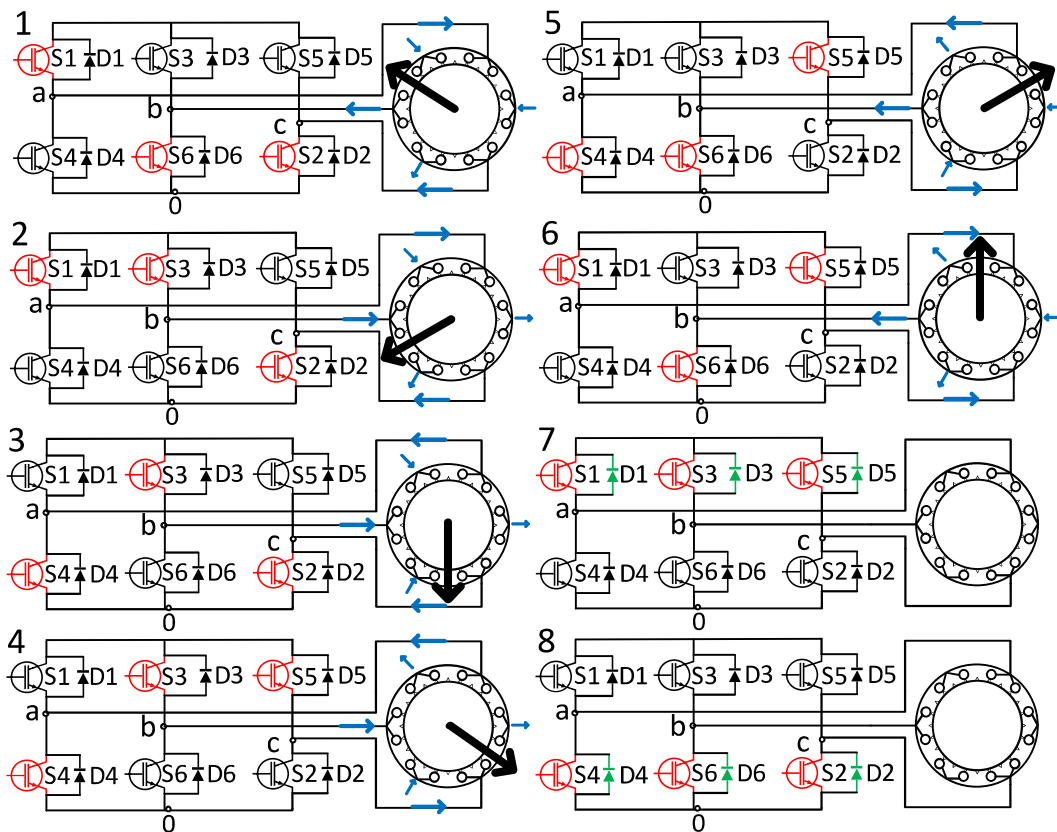


Figura 1.7. Estados válidos del inversor para la modulación SPWM

Aunque la onda entregada por el inversor a la carga sea pulsante y presente distorsión armónica, se considera que en el funcionamiento del motor es predominante la fundamental del voltaje, como se muestra en Figura 1.8.; además, las corrientes del estator son prácticamente sinusoidales, debido a que el estator por estar compuesto de devanados presenta características inductivas. Son estas inductancias las que filtrarán a las corrientes para que adquieran una forma casi senoidal y no cuadradas, tal como se muestran en las gráficas del voltaje entre líneas (V_{ab}) y de la corriente en una fase del motor (I_a) en la Figura 1.9.

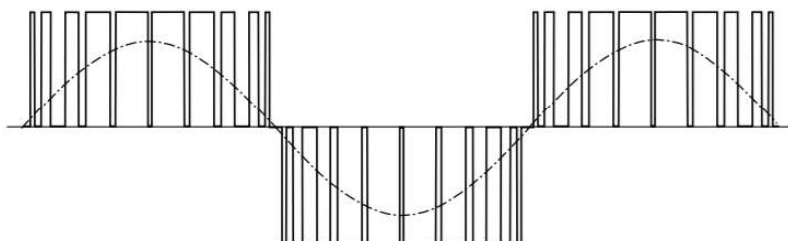


Figura 1.8. Fundamental del voltaje entre líneas mostrada por la línea entrecortada [11]

Las gráficas de la Figura 1.9. se las obtuvieron bajo las mismas condiciones de modulación y carga, con excepción de la frecuencia de la onda portadora. Se observa que al incrementar la frecuencia de la portadora se mejora el rizado de la corriente i_a , de esta manera, al tener una onda más parecida a una sinusoidal pura, se reducen los armónicos de corriente. Se observa que, a mayor frecuencia de conmutación, menor distorsión armónica total (THD-*Total harmonic distortion*) de corriente. Sin embargo, el aumento de la frecuencia de la portadora y el hecho de utilizar un control PWM también ocasiona problemas de interferencias electromagnéticas y degradación en los aislamientos de los devanados del motor, provocando una reducción de su vida útil [13].

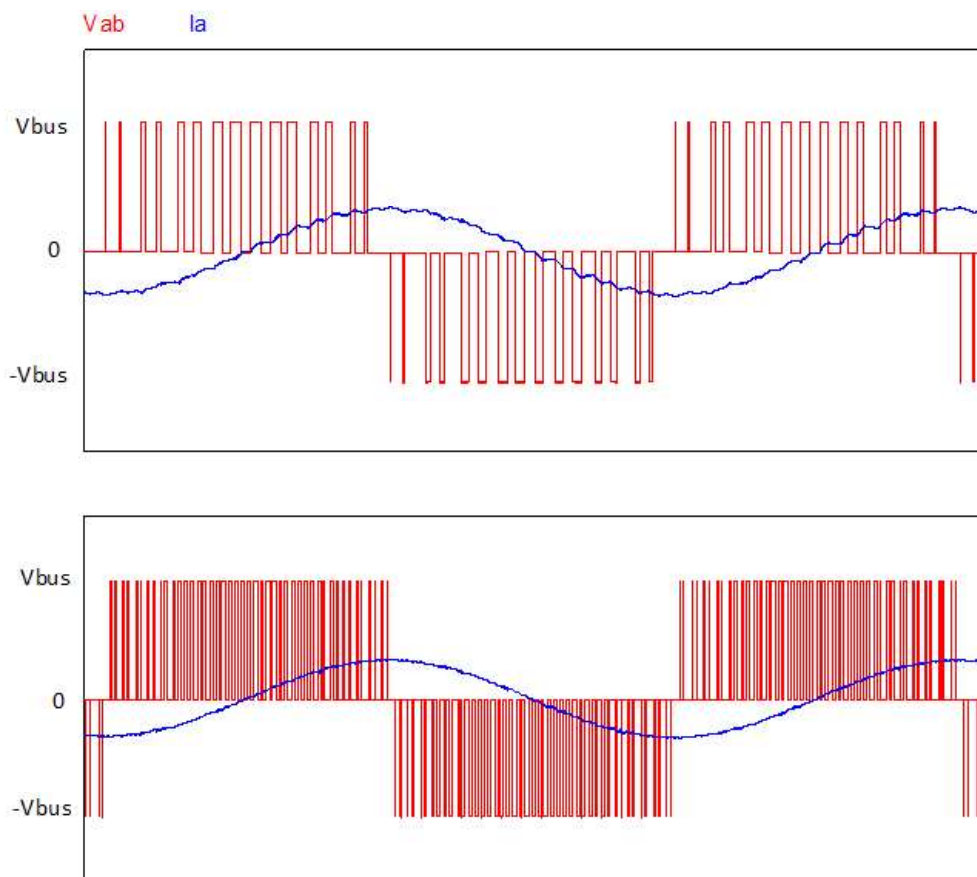


Figura 1.9. Rizado de corriente del motor con diferentes frecuencias de la onda portadora

La velocidad del motor de inducción puede ser modificada al variar la frecuencia de las ondas modulantes, de esta manera, se varía la frecuencia de salida de las corrientes en el estator. También se puede aumentar o reducir la velocidad modificando el índice de modulación, ya que varía de manera proporcional el voltaje de entrada al estator, en consecuencia, se modifica la velocidad del rotor.

En la sección 2, Metodología, se expone el puerto de ZYBO que genera las señales SPWMs, que permiten excitar un inversor trifásico mediante optoaisladores como medida

de protección y, aislamiento del control y la electrónica de potencia; se configura en VHDL el sistema digital que contiene la modulación SPWM y se crea un bloque de propiedad intelectual SPWM; y, se programa el sistema de procesamiento que es el maestro que controla al bloque de propiedad intelectual y la interfaz del usuario con ZYBO.

La modulación SPWM es síncrona y para realizarla se crea, mediante contadores, una onda periódica triangular como onda portadora con amplitud y frecuencia variable; y, por medio de tablas sinusoidales de 8 bits desfasadas 120 grados con respecto de cada una, tres ondas periódicas modulantes sinusoidales de frecuencia variable y amplitud constante.

Para crear el bloque de propiedad intelectual, las configuraciones para la modulación SPWM en VHDL son empaquetadas en conjunto con los protocolos AXI para el intercambio de información entre el sistema de procesamiento y la lógica programable.

El sistema de procesamiento o PS (ARM) es programado en C y se encarga de recibir los datos de frecuencia fundamental e índice de modulación, ingresados por el usuario, reafirmar al usuario el dato que ha sido ingresado o solicitarle que se mantenga dentro del rango de funcionamiento establecido para la aplicación y, finalmente, calcular y enviar las entradas que necesita el bloque de propiedad intelectual para actualizar la modulación SPWM en función de las entradas del usuario.

La interfaz de Matlab es el vínculo entre el usuario, la computadora y ZYBO, esta interfaz es una alternativa al uso de terminales como PUTTY, TERA TERM, etc, puesto que brinda una conexión y utilización más amigable con el usuario y está diseñada específicamente para mejorar el intercambio de información con ZYBO. Esta interfaz se conecta con ZYBO, vía comunicación serial, para enviar los datos digitados por el usuario en el teclado de la computadora; y, presenta la respuesta de ZYBO en la pantalla para verificar el dato que se ha guardado en el Sistema en Chip.

Posteriormente, en la sección 3, Resultados y Discusión, se indican los reportes de utilización, tiempos y potencia del sistema digital completo en ZYBO, con esta información se puede comprender de manera cuantitativa el espacio disponible que se puede usar para añadir más algoritmos en la lógica programable, y verificar que no se violen restricciones de diseño. También se expone la simulación del bloque de propiedad intelectual SPWM con tres condiciones específicas de 60 Hz para la frecuencia fundamental (f) y 96% para el índice de modulación (I_m), $f=42$ Hz e $I_m=38\%$, $f=22$ Hz e $I_m=10\%$; y 87 como relación de modulación de frecuencia, que permiten comprobar el cumplimiento de cada uno de los citados parámetros en las formas de onda, amplitudes y frecuencias de las modulantes y portadora, así como en las 6 señales SPWMs. Por último, se presentan las pruebas de

laboratorio en cargas trifásicas R, RL y motor de inducción; verificando en el voltaje entre líneas, los niveles, formas de onda, variaciones de la frecuencia fundamental y voltaje RMS, en función de las entradas del usuario.

Finalmente, en la sección 4, Conclusiones y Recomendaciones, se presentan las conclusiones y recomendaciones obtenidas como resultado del proceso de investigación, metodología, experimentación y validación.

2. METODOLOGÍA

Este proyecto técnico utiliza una investigación aplicada con la que busca desarrollar una modulación SPWM en el Sistema en Chip ZYBO, beneficiándose de sus características de paralelismo, validándola con el funcionamiento de un motor de inducción. Se realizan consultas en libros, artículos científicos y hojas de datos de fabricantes para recolectar la información de la fase teórica, la cual consta en la sección 1., Introducción, con el objetivo de explicar cómo excitar motores mediante un inversor y las características y beneficios que presentan los Sistemas en Chip Zynq, así como el intercambio de información mediante protocolos AXI entre el procesador del sistema de procesamiento y la FPGA.

Esta sección, Metodología, constituye la fase de implementación metodológica, la misma que tiene el objeto de incorporar el conocimiento de la Introducción para crear un bloque de propiedad intelectual con la modulación SPWM y una interfaz de usuario en ZYBO aprovechando las características de Zynq.

Las fases de simulación y validación mostradas en la sección 3., Resultados y Discusión, muestran las simulaciones de las señales de ZYBO, y las validaciones de la modulación SPWM con distintos tipos de carga obtenidos a partir de la aplicación metodológica de los conocimientos y técnicas expuestas en la Introducción.

Esta sección, fase de implementación metodológica, presenta una introducción de la tarjeta ZYBO y el software Vivado Design Suite, en las subsecciones 2.1. y 2.2., cuya tecnología servirá para programar todos los algoritmos. Los procesos configurados dentro del bloque de propiedad intelectual son explicados en la subsección 2.3. La programación del sistema de procesamiento que controla al bloque IP mediante una interfaz de usuario se detalla en las subsecciones 2.4. y 2.5. Finalmente se resume el trabajo realizado mediante diagramas de flujo en la subsección 2.6.

Se ha realizado una partición entre el hardware y el software, entendiéndose esta como el desarrollo de algoritmos en la PL y en el PS, respectivamente, que permiten demostrar la característica de paralelismo de la FPGA y explotar la capacidad de la unidad de punto flotante del procesador para realizar cálculos con números flotantes.

La PL detalla los algoritmos utilizados en la descripción del hardware necesario para la realización de la modulación SPWM; mientras que, el PS envía las características de las ondas modulantes y portadora a la FPGA en función de la interacción con el usuario.

2.1. ZYBO [14]

ZYBO es una plataforma de desarrollo, producida por la compañía Digilent, en la que se ha instalado el hardware necesario para tener acceso a los periféricos del SOC Zynq XC7Z10 de Xilinx, el cual tiene una arquitectura dual-core ARM Cortex-A9, combinado con un equivalente de FPGA Artix-7. En la Figura 2.1. se muestra la tarjeta ZYBO y se reconoce rápidamente el hardware principal integrado por interruptores, conectores, puertos USB, etc., que permiten que esta tarjeta Zynq (ZYBO) esté lista para ser usada y programada.

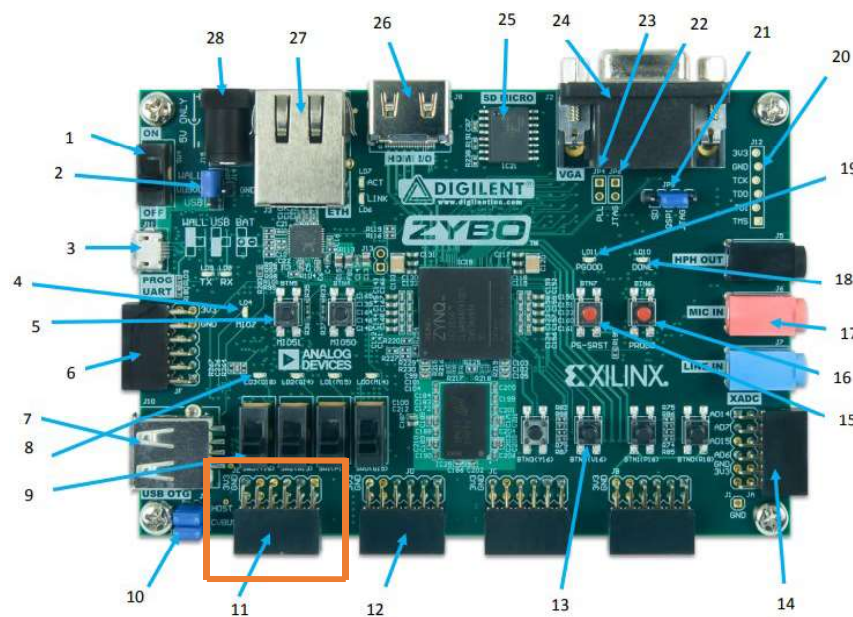


Figura 2.1. ZYBO y sus periféricos [14]

La frecuencia por defecto de la FPGA que se necesita conocer para desarrollar los algoritmos de la PL es 125 MHz, este es el reloj con el que se sincroniza el hardware que se configura con el lenguaje de descripción de hardware de circuito integrado de muy alta velocidad (VHDL-*Very high-speed integrated circuit hardware description language*).

Se necesitan 6 señales SPWM para que exciten a los interruptores del inversor trifásico (Figura 1.5., pág. 9), la tarjeta ZYBO presenta puertos de módulos periféricos (Pmod-*Peripheral module*) de donde se pueden obtener dichas señales. El puerto Pmod estándar se puede observar en la Figura 2.1., dentro del recuadro anaranjado, el mismo que está conectado a la PL de Zynq y sirve como un puerto de entradas y salidas de propósito general. En la Tabla 2.1. se presentan las características del citado puerto:

Tabla 2.1. Características del puerto Pmod estándar [14]

Puerto Pmod estándar
Protección contra cortocircuito
Entradas o salidas de un solo extremo (<i>single ended</i>)

La Figura 2.2. evidencia que se pueden utilizar 8 señales en el puerto que se está analizando, además, tiene la característica de que a cada uno de los 8 pines se los puede referenciar ya sea a la tierra o al voltaje de alimentación (GND o VCC), de esta manera se pueden obtener las 6 señales SPWMs del mismo puerto.

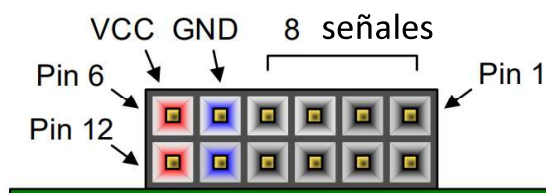


Figura 2.2. Diagrama de puerto Pmod estándar [14]

Las frecuencias de conmutación nominales utilizadas en inversores trifásico normalmente fluctúan entre los 5 kHz [15], [16], [17], [18], [19]. Debido a las capacidades que poseen los IOBs de Zynq de entregar señales de altas velocidades, el puerto Pmod estándar presenta buenas características dinámicas para proporcionar ondas con frecuencias de conmutación de este rango, por lo que es utilizado para generar las señales SPWMs.

ZYBO es compatible con los entornos de desarrollo Vivado Desing Suite, SDSoC y con las herramientas de ISE/EDK, todos pertenecientes a Xilinx. Se escoge trabajar con Vivado Desing Suite porque permite programar a bajos niveles de abstracción (VHDL) en comparación a SDSoC (lenguajes de programación basados en C). Además, otra razón para utilizar Vivado Design Suite, es que Digilent a la par con Xilinx ofrecen un extenso soporte para ZYBO en este entorno de desarrollo, a diferencia de ISE/EDK.

Es necesario explicar cómo funciona la aplicación a un nivel global más detallado en ZYBO, porque a partir de esos criterios se configura la aplicación en VHDL, luego se crea el bloque de propiedad intelectual y finalmente se programa el código del PS. Las funciones de la interfaz con el usuario son realizadas en el PS, independientemente de las que ejecuta la PL, obteniéndose la separación entre las actividades principales de control SPWM y las secundarias de interfaz con el usuario. En la Figura 2.3. se explica cómo se fusionan las dos partes fundamentales de Zynq: PL y PS, para recibir la información del usuario, realizar cálculos y enviar registros actualizados a la PL, sin comprometer el control principal de modulación.

Las entradas a ZYBO son los parámetros de frecuencia fundamental de salida del inversor para la carga y el índice de modulación de la SPWM que son digitados en una interfaz gráfica en la computadora y enviados por comunicación serial a través del controlador receptor/transmisor asíncrono universal (UART-*Universal Asynchronous*

Receiver/Transmitter) al CPU del PS, estos datos son convertidos por el CPU en parámetros que tienen el objeto de actualizar las propiedades de la modulación SPWM en la PL, finalmente se escriben estos parámetros calculados previamente en el bloque IP de la PL por medio de la central de interconexiones AXI que es el nexo entre el PS y la PL en Zynq. Una vez actualizado el punto de operación ingresado por el usuario en el bloque IP, sus señales de salidas PWM y SPWM son actualizadas en el puerto estándar JE y el led 0 (LD0), como se describe en la sección 2.3.

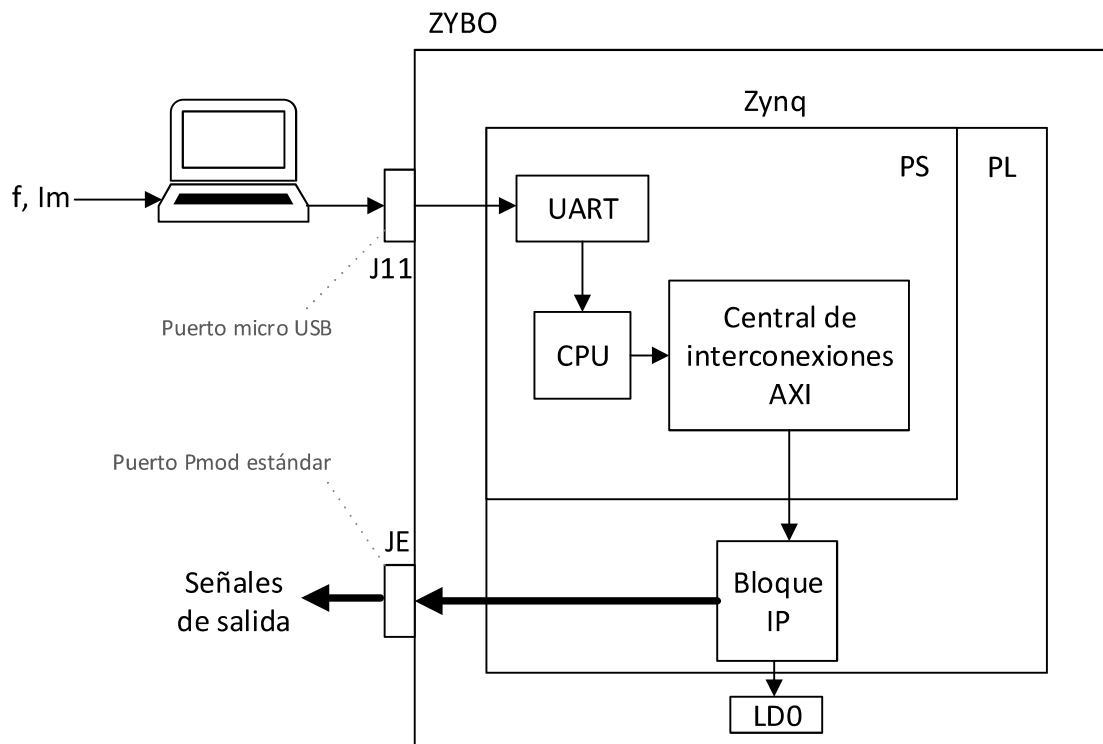


Figura 2.3. Esquema general del flujo de la aplicación en el sistema en chip

2.1.1. OPTOACOPLAMIENTO ENTRE ZYBO Y GATE-DRIVERS

Las señales a la salida del puerto Pmod estándar pasan por un proceso de aislamiento óptico para conectarse a los *gate-drivers* que excitan a los interruptores de los ramales del inversor trifásico, con el fin de evitar dañar a la tarjeta ZYBO en el caso de que se produzca una falla del sistema de potencia.

En la Figura 2.4. se muestran los circuitos de aislamiento utilizados en la excitación de un ramal del inversor trifásico. Se utilizan opto-nands 6N137 para desacoplar el circuito de potencia del circuito de control (ZYBO) con la configuración sugerida por la hoja de datos del optoaislador [20].

La resistencia limitadora de corriente conectada al ánodo del led (pin 2), de los optoacopladores, se dimensiona para 10 mA aproximadamente. Se calcula con la ley de Ohm, el valor de la resistencia, conociendo que el voltaje entregado por el puerto Pmod estándar de ZYBO es de 3.3 V (V_{ZYBO}), y la caída de voltaje de conducción típico de los leds (V_f) es 1.4 V. La corriente esperada (I) y los voltajes mencionados son remplazados en la Ecuación 2.1., obteniéndose un valor de 190 ohmios que es normalizado a 180 ohmios acorde a las resistencias comerciales.

La potencia que disipa la resistencia, conectada en serie a los leds, alimentada continuamente con una corriente de 10 mA es 0.018 W como se muestra en el cálculo de potencia de la Ecuación 2.2. El análisis se lo realiza asumiendo que la corriente es constante, sin embargo, estas resistencias serán alimentadas con voltajes SPWM, por lo tanto, la potencia real que disiparán es menor a 0.018 W. Según el análisis realizado se opta por utilizar resistencias de un cuarto de vatio, ya que están sobredimensionadas para la potencia de la aplicación mencionada.

$$R = \frac{V_{ZYBO} - V_f}{I} = \frac{3.3V - 1.4V}{10mA} = 190\ \Omega \rightarrow 180\ \Omega \quad (2.1)$$

$$P = I^2 R = (0.01^2)(180) = 0.018\ W < 0.25\ W \quad (2.2)$$

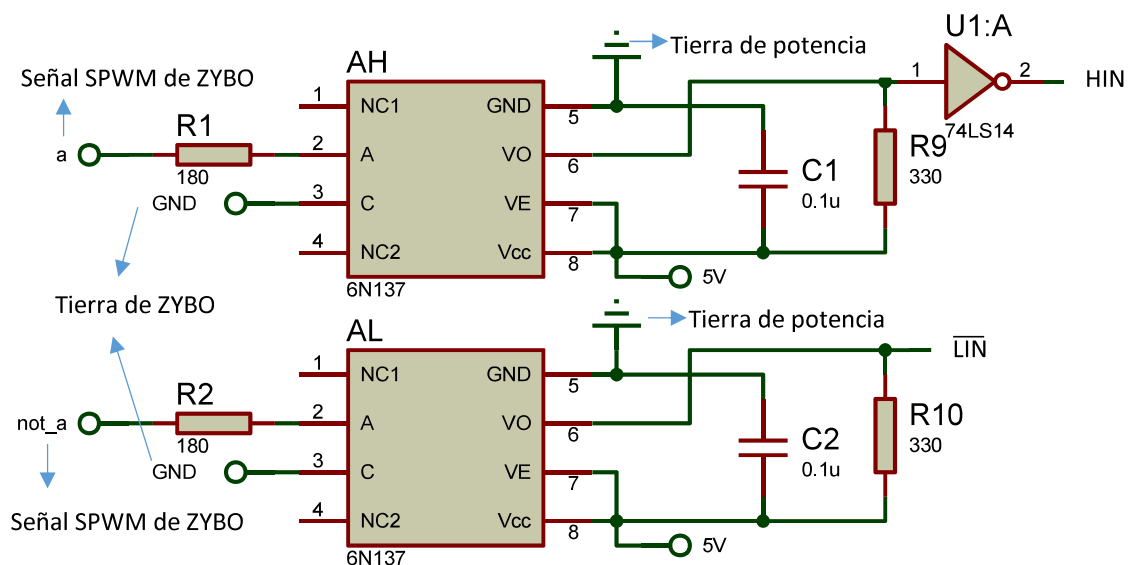


Figura 2.4. Aislamiento óptico entre las señales de salida de ZYBO para el ramal 'a' y las entradas de un *gate-driver*

El funcionamiento del *gate-driver* L6390D [21] utilizado en el inversor, con el que se procede a validar el proyecto en la sección 3, toma dos señales SPWMs de entrada, “HIN” para alimentar al interruptor superior de un ramal, y “ $\overline{\text{LIN}}$ ” para alimentar al inferior, como se muestra en las señales de salida de la Figura 2.4. y en las entradas señaladas en la Figura 2.5. Este *gate-driver* presenta la particularidad de que su entrada “ $\overline{\text{LIN}}$ ” está negada, por lo tanto, todas las SPWMs que proporciona ZYBO para el accionamiento de los interruptores inferiores del inversor deben ser negados para establecer un correcto funcionamiento.

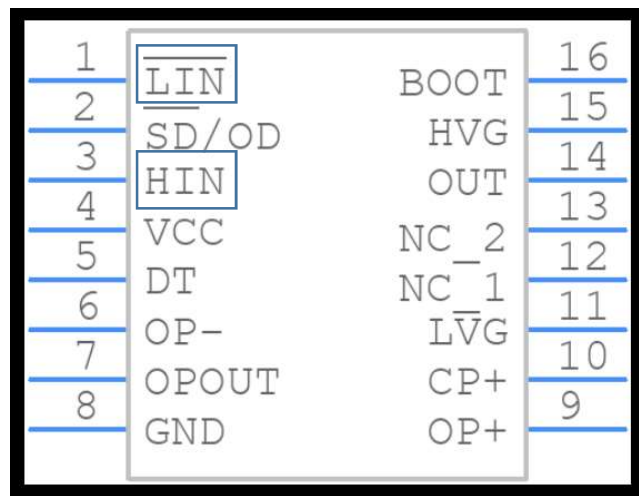


Figura 2.5. Periféricos de *gate-driver* L6390D [22]

La salida de la opto-nand (V_o), con la configuración mostrada en el circuito inferior de la Figura 2.4. invierte la señal de entrada (not_a), de tal manera que se puede alimentar directamente a la entrada “ $\overline{\text{LIN}}$ ” del *gate-driver*.

La salida de la opto-nand (V_o), del circuito superior también invierte a su señal de entrada (a). Para alimentar a “HIN” se niega la señal de salida de la opto-nand utilizando un *Schmitt trigger*, de esta manera la misma forma de onda de entrada del optoacoplador (a) es alimentada a la entrada “HIN” del *gate-driver*.

Los dos ramales adicionales del inversor trifásico tienen la misma configuración que los circuitos mostrados en la Figura 2.4.

2.2. VIVADO DESING SUITE

ZYBO puede ser programado en distintos softwares tanto para la PL como para el PS, esta subsección tiene el fin de indicar las herramientas que se utilizan para programarlo y como integrar un sistema digital completo utilizando Vivado Design Suite.

Vivado es un entorno de desarrollo que crea automatizaciones de diseños electrónicos a partir de un código de registro a nivel de transferencia (RTL). VHDL se utiliza para describir al RTL, modelando su estructura lógica y la función del circuito digital completo, como se observa en la Figura 2.6., donde se crean circuitos secuenciales como FFs y combinacionales como un conjunto de compuertas lógicas.

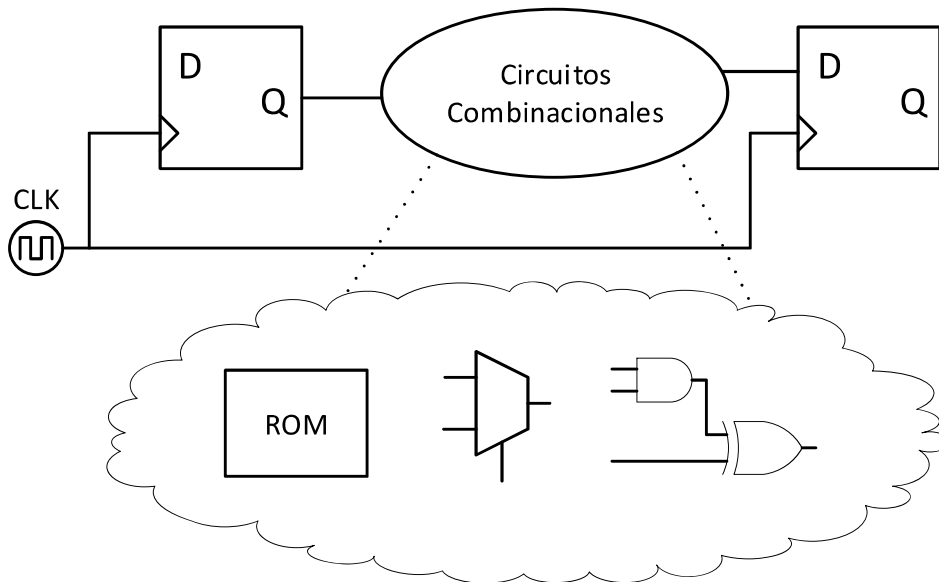


Figura 2.6. Esquema de un circuito digital

La síntesis, en Vivado, analiza el modelo RTL revisando que el código sea escrito conforme a los estilos requeridos para VHDL y realizando controles de reglas de diseño, por ejemplo, verificar que cada señal sea utilizada en las respectivas entradas o salidas. Luego, Vivado infiere el hardware del modelo utilizando circuitos principales básicos como FFs, multiplexores, *carries* [23]. En la Figura 2.7. se indican las entradas de la herramienta de síntesis, constituidas por el código escrito en VHDL y sus librerías, además de restricciones para modificar el *netlist* a nivel de compuerta; el principal objetivo de la síntesis es convertir el modelo de RTL de Vivado en un *netlist* a nivel de compuerta; y, como información secundaria, se tiene acceso a reportes originados por esta operación de síntesis [24].

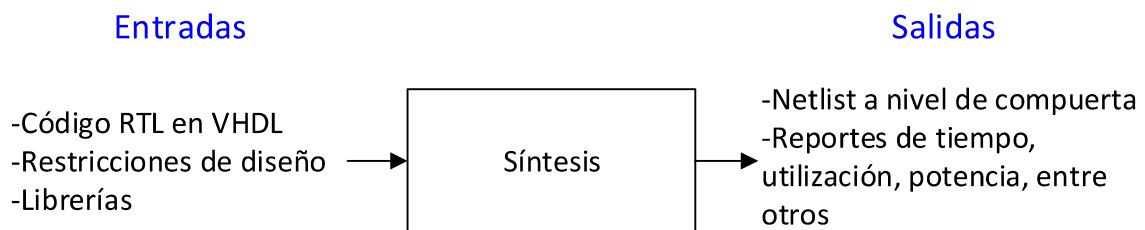


Figura 2.7. Proceso de síntesis

La herramienta de implementación de Vivado permite implementar el circuito digital de una manera física en las FPGAs. A partir del *netlist* generado por la síntesis se comienza con el proceso de implementación que está conformado por tres pasos principales:

- Optimización: Optimiza el diseño del *netlist* para que los circuitos entren en la FPGA de una manera más sencilla.
- *Place*: Coloca a los elementos como FFs, LUTs y *carries* en los arreglos de la FPGA, considerando la geometría y el espacio físico.
- Ruteo: Conecta a todos los elementos que han sido colocados en la FPGA [25].

En la Figura 2.8. se indican las entradas y salidas del proceso de implementación que realiza Vivado. Al igual que en la síntesis, se pueden agregar condiciones de diseño que restringen a la implementación para que se obtenga un circuito distinto con otro tipo de conexiones, priorizando el área, velocidad o manejo de potencia de la FPGA. La salida principal de la implementación es el circuito digital que se encuentra ubicado y conectado dentro de los arreglos de la FPGA; además, se puede acceder a reportes del proceso cuando ha finalizado la implementación.

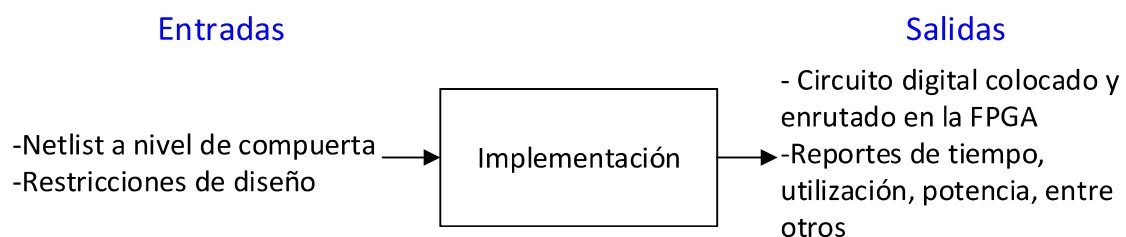


Figura 2.8. Proceso de implementación

Finalmente, para programar al dispositivo se genera un archivo *bitstream*, concluyendo así con el proceso de implementación.

2.3. CONFIGURACIÓN DE LA PARTE LÓGICA PROGRAMABLE

Se desea crear una configuración en hardware utilizando VHDL, de manera que presente las capacidades modulares que permitan ampliar el programa a futuro y también concedan una facilidad de lectura del código al tener tareas separadas. Por las razones mencionadas se decide utilizar procesos, y debido a que desarrollan varias tareas específicas de manera concurrente, se pueden agregar más procesos, que además, brindan una estructura parecida a lenguajes de alto nivel.

Se crean SPWMs síncronas, de tal manera que el número de portadoras por cada onda sinusoidal esté en función de la frecuencia fundamental de salida. La configuración de la PL construye señales discretas de las ondas modulantes y portadora, que serán comparadas para obtener SPWMs con tiempos muertos. Estas reconstrucciones mantienen una armonía y orden entre ellas, lográndose producir las ondas de la Figura 1.6. (VR, VS, VT, Vportador), presentada en la página 10, para distintos índices de modulación y frecuencia; en tales casos, la amplitud de la onda portadora varía de manera inversamente proporcional al I_m y las frecuencias de las modulantes adquieren el valor de la frecuencia fundamental de la carga.

El criterio de Nyquist establece que la frecuencia de muestreo de una señal periódica es por lo menos el doble de la frecuencia más alta de la señal a muestrear. Para construir las ondas discretas modulantes y portadora, se utiliza el criterio de Nyquist como punto de partida. A continuación, se analizan las características de las frecuencias de las ondas periódicas que se quieren digitalizar:

1. Se desea muestrear señales sinusoidales puras, lo que implica que, existe una única composición armónica para cada modulante, por consiguiente, estas señales tienen una única frecuencia.
2. La onda portadora está compuesta por infinitas formas de ondas a distintas frecuencias, sin embargo, el interés está en su frecuencia fundamental.

Por las dos características mencionadas, se concluye que, la frecuencia mínima de muestreo, según Nyquist, para las ondas portadora y modulantes es el doble de sus frecuencias fundamentales, sin embargo, para utilizar este criterio de una manera práctica en este diseño, se utilizan por lo menos diez veces la frecuencia fundamental. Los procesos: PROCESO DE PERÍODO DE MUESTREO DE LAS ONDAS MODULANTES y PROCESO DE LA CONSTRUCCIÓN DE LA ONDA PORTADORA, que se analizan en las dos siguientes secciones, están encargados de crear las señales discretas y profundizan la explicación del número adecuado y funcional de puntos de construcción de las ondas.

En la Figura 2.9. se explica con un diagrama de bloques simplificado los procesos que se ejecutan en la PL concernientes a la modulación SPWM. Todos los procesos están sincronizados con los flancos del reloj de la FPGA, en el proceso 12, adicionalmente, se realimenta a la señal Desciende obtenida de la salida del proceso 2 para sincronizar una interrupción interna. Las salidas de los procesos configuran salidas físicas en los periféricos de ZYBO, excepto la señal Desciende que es utilizada internamente.

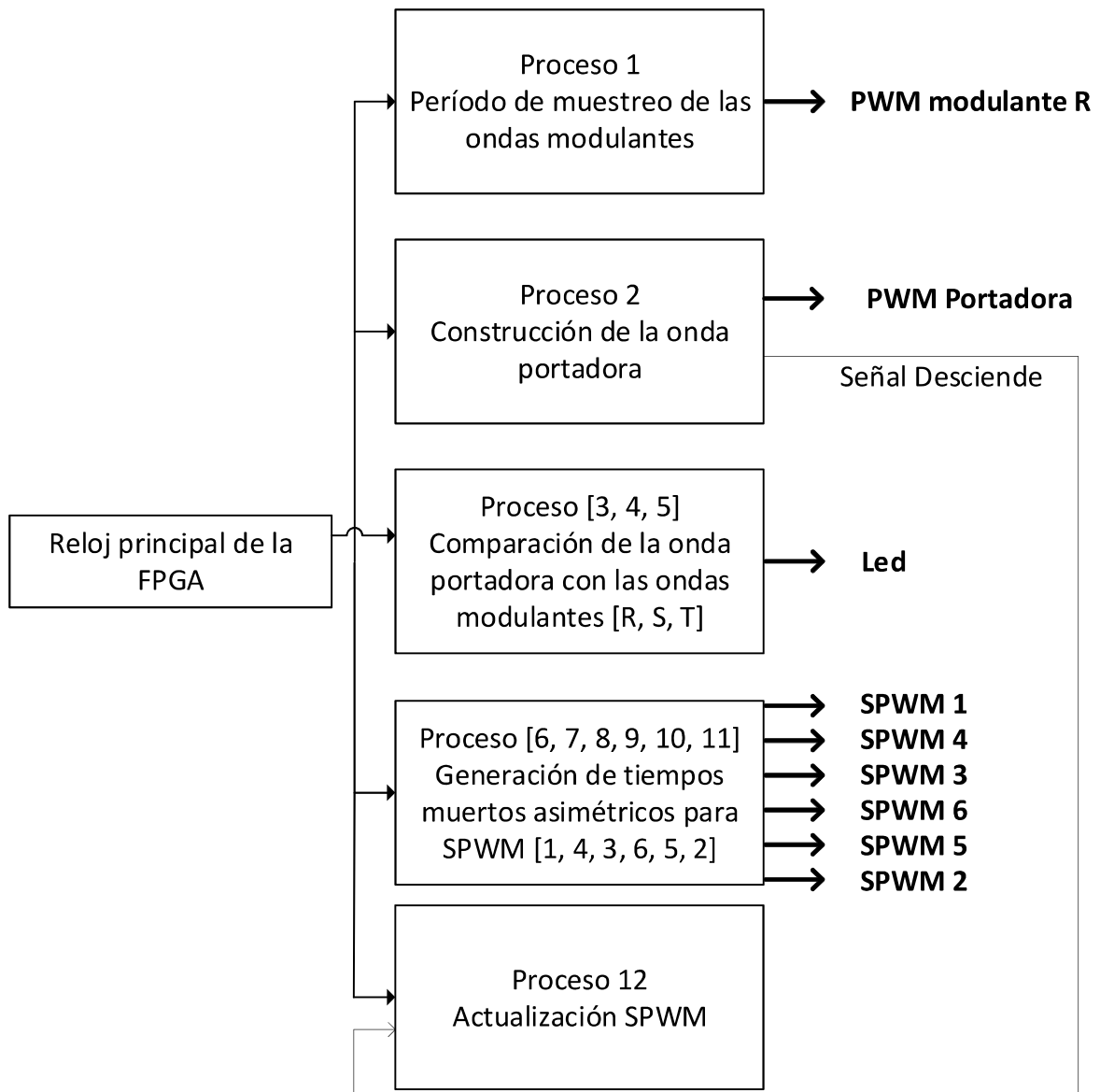


Figura 2.9. Esquema general de configuración de la PL para la modulación SPWM

En los siguientes cinco títulos (2.3.1. - 2.3.5.) de esta subsección 2.3. se explica a detalle los procesos mostrados en la Figura 2.9.

2.3.1. PROCESO DE PERÍODO DE MUESTREO DE LAS ONDAS MODULANTES

Para muestrear una onda análoga y convertirla a digital se debe discretizar en amplitud y tiempo. La digitalización por amplitud se la realiza con una codificación de 8 bits a partir de señales sinusoidales DC trifásicas desfasadas 120° entre ellas y con secuencia positiva, produciendo 256 niveles para cada modulante, y variaciones entre un mínimo de 0 y un máximo de 255 unidades enteras en amplitud. Esta digitalización se la realiza de una manera manual para obtener tablas de ondas sinusoidales de 256 niveles con 361 puntos.

El proceso del período de muestreo de las ondas modulantes trabaja con los puntos de las 3 tablas senoidales; el objetivo de este proceso es indicar que punto de la tabla debe usarse y por cuanto tiempo, para realizar los procesos de comparación con la onda portadora. Se utiliza una frecuencia de muestreo de 360 veces la fundamental para las ondas modulantes, de esta manera, se obtienen los 361 puntos de las tablas para cada una de las fases (R, S, T), con esta cantidad de puntos se cumple con la frecuencia de Nyquist para señales discretas periódicas; y, también agrega la funcionalidad de representar a las ondas sinusoidales por grados, de 0° a 360°.

Este proceso configura una salida por el puerto Pmod estándar de ZYBO mediante una señal PWM de la modulante R, la importancia de esta señal es observar en que momentos se repite la onda modulante R, con el objeto de utilizarse como una herramienta de depuración que permita mostrar el comportamiento interno en tiempo real de la modulación SPWM en ZYBO. Cada vez que se reinicia la modulante se cambia el estado de la PWM, si originalmente estaba en 0L, cambiará a 1L y viceversa, como se muestra en la Figura 2.10., esto implica que la PWM tiene la mitad de la frecuencia de la modulante R.

La señal PWM modulante R es desacoplada ópticamente utilizando la configuración mostrada en el circuito inferior de la Figura 2.4.

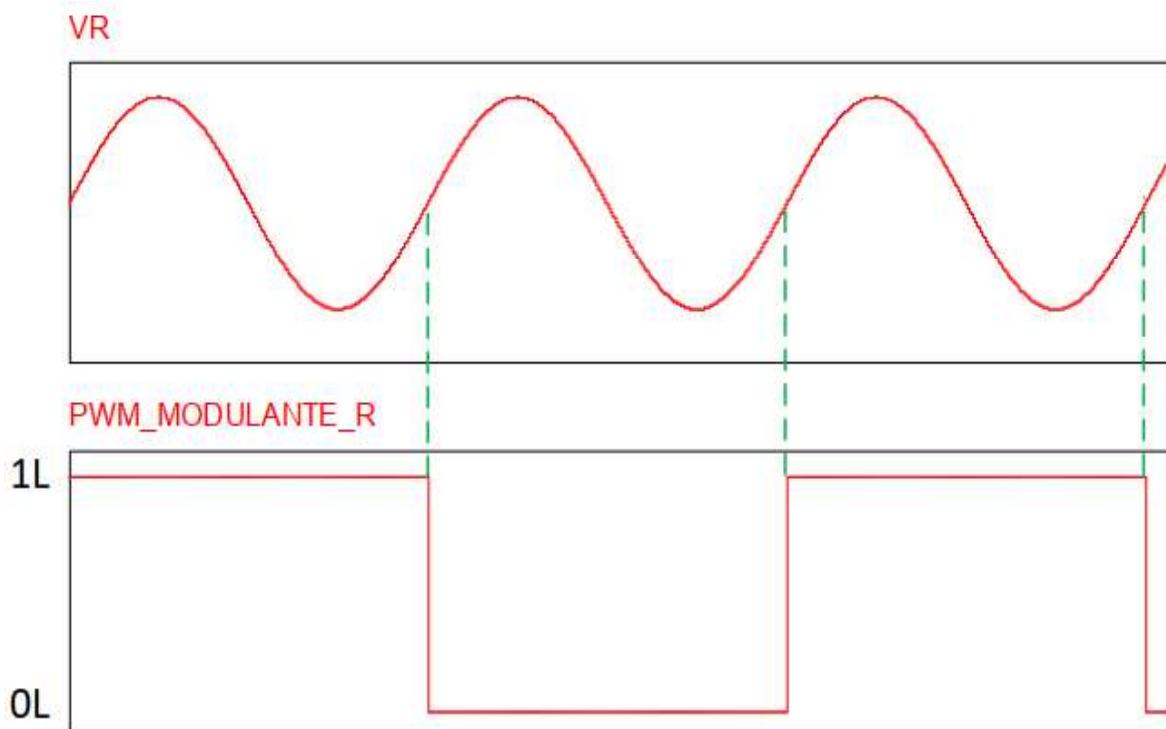


Figura 2.10. PWM de la mitad de la frecuencia de la modulante R

2.3.2. PROCESO DE LA CONSTRUCCIÓN DE LA ONDA PORTADORA

Este proceso construye a la onda portadora en función de la relación de modulación de frecuencia (mf), esta relación es entera para que la modulación trifásica sea síncrona y su valor indica el número de portadoras presentes en un período de cualquiera de las 3 ondas modulantes. La portadora es la onda que cambia de amplitud al requerirse otra condición de modulación, como la amplitud de las ondas modulantes es constante, la amplitud de la portadora varía de una manera inversamente proporcional al índice de modulación. La señal portadora se codifica con 12 bits y además se conoce que las ondas modulantes son representadas con 8 bits, por lo tanto, el mínimo valor del índice de modulación que se puede obtener es 0.0622.

Uno de los resultados del proceso de la construcción de la onda portadora es la señal de salida llamada Desciende que sirve como entrada para el proceso de Actualización SPWM y esta misma señal configura a la salida PWM Portadora, de tal manera que se obtienen las gráficas observadas en la Figura 2.11., donde se indica con los cursores que la PWM es 1L en el punto máximo de la portadora y 0L en los puntos mínimos. La importancia de la onda PWM portadora es que tiene la misma frecuencia de la onda portadora triangular y, además, identifica implícitamente si la portadora crece o decrece y en qué momentos se alcanza sus puntos máximos y mínimos; así, se puede analizar el comportamiento de la portadora desde un osciloscopio, o bien, se puede utilizar esta salida para sincronizar elementos electrónicos externos a ZYBO. Se aclara que la señal Desciende presenta el mismo comportamiento que la PWM portadora, indicada en la citada figura.

La señal PWM portadora es desacoplada ópticamente utilizando la configuración mostrada en el circuito inferior de la Figura 2.4.

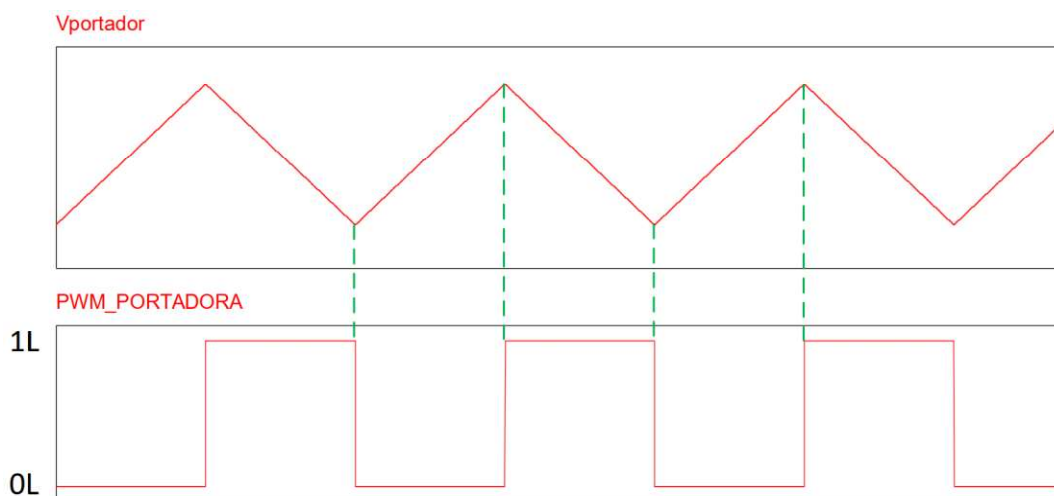


Figura 2.11. PWM y onda Portadora de la misma frecuencia

Este proceso está estrechamente ligado al programa del PS que se explica en el título 2.4., bajo los subtítulos: 2.4.1. ALGORITMO DE EVALUACIÓN DE NÚMERO PRIMO, 2.4.2. ALGORITMO DE DESCOMPOSICIÓN EN FACTORES PRIMOS y 2.4.3. ALGORITMO PARA REDUCIR EL ERROR DEL PERIODO DE MUESTREO DE LA ONDA PORTADORA; en las respectivas secciones mencionadas se profundizarán las interacciones con variables y los algoritmos desarrollados.

2.3.3. PROCESOS DE COMPARACIÓN DE LA ONDA PORTADORA CON LAS ONDAS MODULANTES

Estos procesos realizan la función de los comparadores de la Figura 1.5., presentada en la página 9, dependiendo de si la onda modulante es mayor que la portadora, en un instante de tiempo, se asigna 1L o 0L a las señales SPWM. La Tabla 2.2. presenta los resultados obtenidos en la creación de las SPWMs S1 y S4, sin tiempos muertos; al intersecar la fila de la comparación con la columna de la SPWM, se obtiene un resultado. Existen tres procesos que comparan a la onda portadora con la modulante R, S y T, pero, en este tema se analiza únicamente la comparación de la fase R y se entiende que se obtienen resultados similares para los interruptores de los ramales b y c.

Tabla 2.2. Condiciones y resultados de las SPWMs del ramal a

Comparaciones	S1 (sin tiempo muerto)	S4 (sin tiempo muerto)
$V_R > V_{portador}$	1L	0L
$V_R < V_{portador}$	0L	1L
$V_R = V_{portador}$	Estado lógico anterior	Estado lógico anterior

En la Figura 2.12. se muestran los resultados de la Tabla 2.2. en un periodo de la onda modulante R. Es evidente que las señales SPWM S1 y S4 son complementarias, evitando de esta manera ocasionar cortocircuitos en el ramal 'a' de la topología del inversor utilizada, por lo menos en los estados estacionarios; la prevención de cortocircuitos en las transiciones es explicada en el siguiente conjunto de procesos (subsección 2.3.4.). Se observa, además, que las SPWMs tienen un valor máximo y mínimo de voltaje en sus señales, 1L y 0L, esto permite representar a las ondas de manera general, en cualquier punto en las que se observen. Las señales atraviesan circuitos y su nivel de voltaje de 1L cambia a las salidas de cada uno. La Figura 2.13. presenta una cadena de bloques de hardware, los mismos que son alimentados en cada etapa por SPWMs con diferentes niveles lógicos de voltaje.

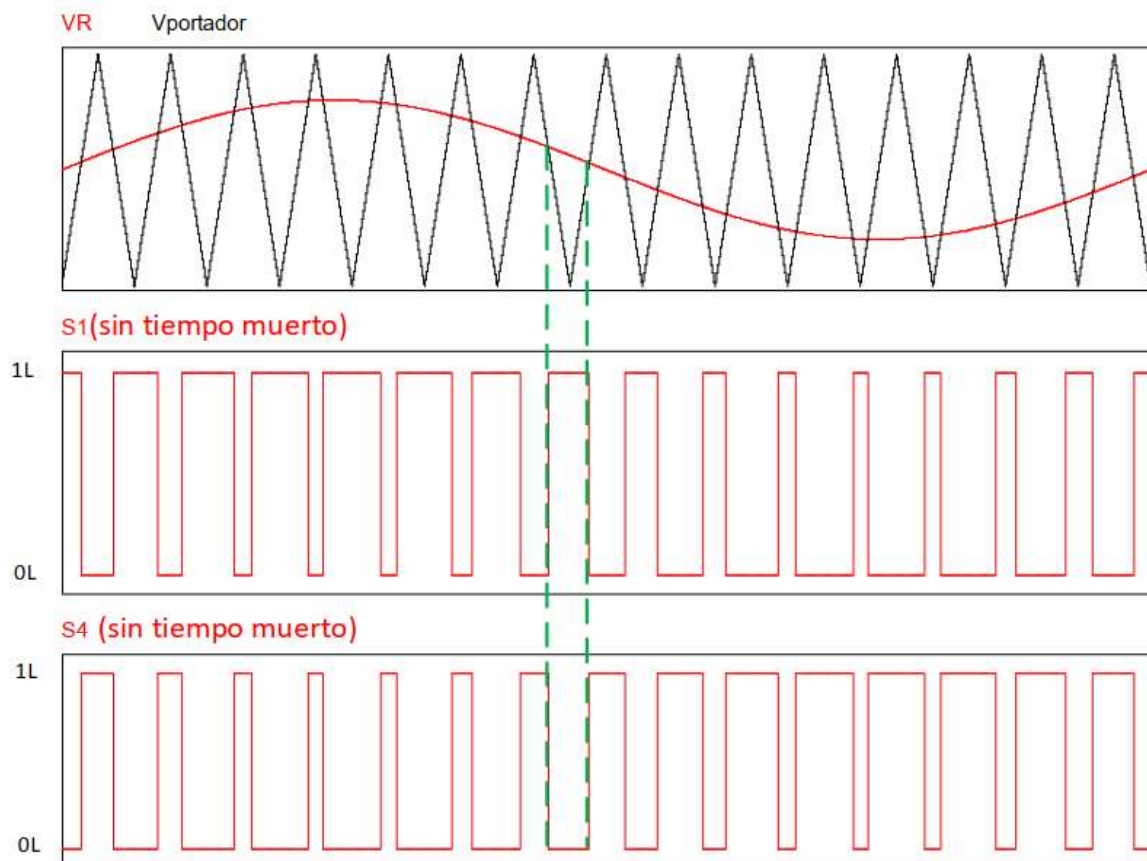


Figura 2.12. Comparación de onda modulante R con portadora y SPWMs para el ramal 'a'

En la Figura 1.5., presentada en la página 9, se observó la activación de los interruptores de potencia a partir de excitaciones de la modulación SPWM, aunque ZYBO genera estas señales, sus niveles de voltaje y potencia no son suficientes para activar los interruptores, por este motivo, se necesita acoplar el hardware necesario para alcanzar niveles adecuados de energía y hacerlo de una manera eficiente (subsección 2.1.1. OPTOACOPLAMIENTO ENTRE ZYBO Y *GATE-DRIVERS*). En la Figura 2.13. se presenta un esquema en el que se indica cómo se acoplan las señales SPWM entre las 4 etapas de hardware. Todas las etapas definen el nivel 0L como 0 V, sin embargo, no necesariamente tienen las mismas referencias; además, se observa que las salidas de las etapas cambian el nivel de voltaje de 1L en función de la entrada del siguiente bloque. Analizando de derecha a izquierda, los interruptores necesitan un *driver* que maneje sus compuertas con cantidades suficientes de corriente y niveles de voltaje apropiados para una correcta y efectiva conmutación, luego, existe riesgo en conectar el sistema de control directamente a los circuitos de potencia, por esto, se añade la etapa de aislamiento óptico revisada en la subsección 2.1.1. entre los *drivers* y la tarjeta ZYBO.

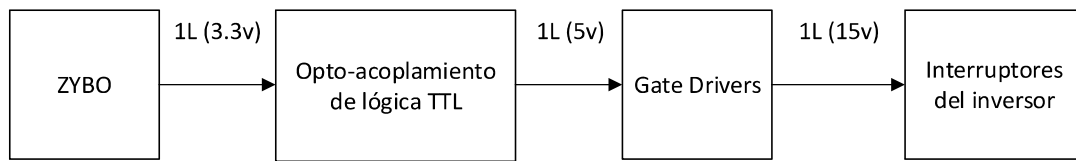


Figura 2.13. Cadena de voltajes digitales para las señales SPWM

Dependiendo de los *gate-drivers*, algunas de las señales originales SPWM de ZYBO pueden negarse como requisito de sus entradas, no obstante, las señales de salida para las compuertas de los interruptores son las mismas que se observan en la Figura 2.12. y de manera similar para el resto de los ramales del inversor trifásico.

El presente proceso analizado configura al led 0 (LD0) de ZYBO como salida, con el comportamiento de la SPWM S1 sin tiempo muerto. Esta salida sirve como indicativo de que la aplicación de la FPGA ha sido programada, cuando el usuario ingresa diferentes valores de I_m y frecuencia se observa cómo cambia la luminosidad del led 0, así como la variación de tiempo en el encendido-apagado del led 0, respectivamente.

2.3.4. PROCESOS DE GENERACIÓN DE TIEMPOS MUERTOS ASIMÉTRICOS PARA SPWM

Los interruptores de potencia necesitan un tiempo determinado para apagarse y es mayor al encendido, por esta razón, se debe asegurar que el interruptor superior de un ramal se apague completamente antes de encender su complemento inferior o viceversa.

Estos procesos crean definitivamente las señales que saldrán por el puerto Pmod estándar de ZYBO. Se configuran tiempos muertos para cada SPWM, así se evitan los cortocircuitos en los tiempos transitorios de la Figura 2.12. El tiempo muerto generado es asimétrico, es decir, se mantienen los flancos de bajada de las señales SPWM y se retrasan los flancos positivos [26].

Existen 6 procesos que añaden tiempos muertos asimétricos a cada una de las señales SPWM (S_n), individualmente. Se analizará únicamente el comportamiento dinámico de las SPWMs S1 y S4 para el ramal 'a'; los dos ramales restantes describen resultados similares. El tiempo muerto configurado es de 1.288 μs , suficiente para apagar la mayor parte de IGBTs comerciales, y se lo puede observar en la Figura 2.14., donde se ve que el proceso retrasa los flancos de subida de las SPWMs S1 y S4, a partir de las señales originales sin tiempo muerto. El tiempo muerto utilizado para esta aplicación se lo escoge en base al

rango de los tiempos muertos típicos para MOSFETs e IGBTs presentados en el *gate-driver* L6390D [21].

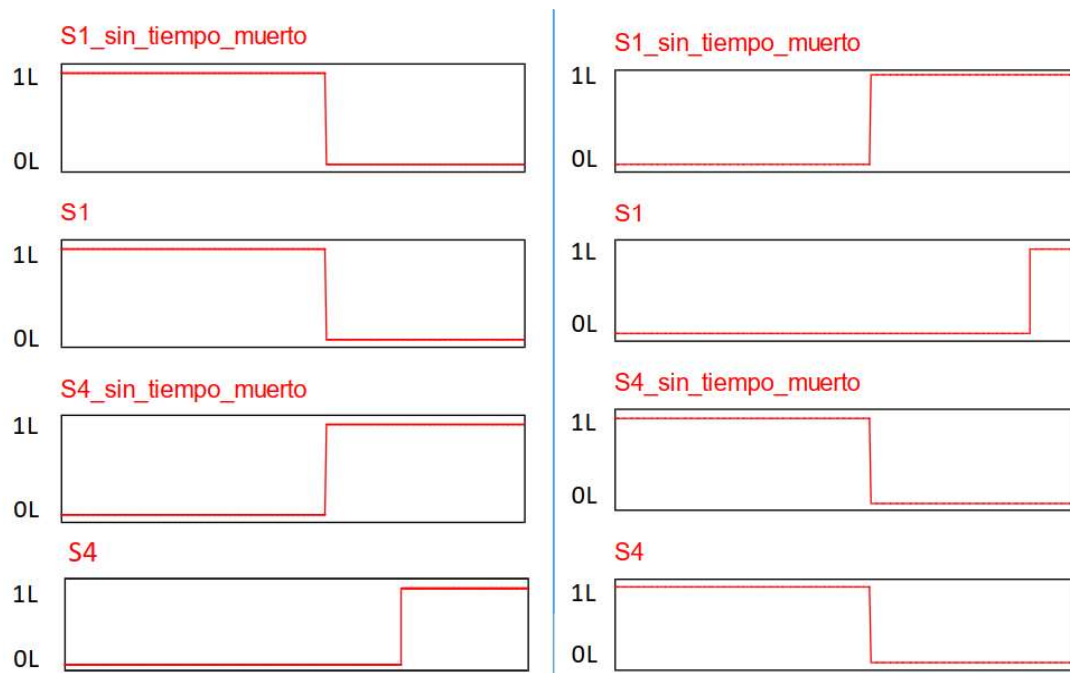


Figura 2.14. Tiempos muertos asimétricos para el ramal 'a'

2.3.5. PROCESO DE ACTUALIZACIÓN SPWM

El usuario puede ingresar nuevas características de modulación asincrónicamente, sin embargo, la FPGA debe mantener un orden de tiempos dentro del bloque IP para el correcto funcionamiento de todos los algoritmos, por esta razón los datos ingresados son aceptados de una manera síncrona.

El cambio de parámetros, realizado por el usuario, es actualizado únicamente en los puntos mínimos de la onda portadora, de esta manera se garantiza que las condiciones iniciales de todas las señales internas de la PL estén reseteadas. Este proceso está sincronizado con el flanco negativo de la señal Desciende, tiempo en el que la portadora se reinicia en su punto mínimo para empezar otro periodo con una pendiente positiva.

Con esta subsección 2.3.5. se termina de explicar los procesos de la PL de la Figura 2.9., y se continua con la explicación de las salidas de los periféricos de ZYBO y el empaquetamiento del bloque IP.

2.3.6. CONFIGURACIÓN DE PINES DE SALIDA

Aparte de configurar la aplicación en VHDL, se debe especificar qué recursos de hardware en la tarjeta ZYBO se van a utilizar como salidas. En un archivo de restricciones se

seleccionan las salidas SPWMs y PWM para que se reflejen por los pines del puerto JE, como se muestra en la Figura 2.15. También se ha asignado el valor de la SPWM S1, sin tiempo muerto, al led 0.

JE

VCC	GND	SPWM S6	SPWM S3	SPWM S4	SPWM S1
VCC	GND	PWM Portadora	PWM R	SPWM S2	SPWM S5

Figura 2.15. Pines de salida del puerto Pmod estándar (JE) con sus respectivas señales

2.3.7. BLOQUE DE PROPIEDAD INTELECTUAL PARA APLICACIÓN DE SPWM

Todos los procesos explicados en esta sección, 2.3. CONFIGURACIÓN DE LA PARTE LÓGICA PROGRAMABLE, presentan los algoritmos necesarios para realizar la modulación SPWM que permita excitar a un inversor trifásico. En este tema se explican las consideraciones que se toman para elegir el bloque IP adecuado en el cual se empaquetan los procesos, sus ventajas y finalmente se muestra un esquemático de todo el sistema que se emplea en ZYBO incluyendo el PS y la PL.

El bloque IP, localizado en la PL, necesita recibir información del PS, por lo tanto, se debe empaquetar un bloque IP de periférico AXI4 para acceder a los protocolos de intercambio de información AXI entre el PS y la PL. En la Tabla 1.1., de la página 6, se presentó las características de los 3 protocolos AXI y se decide trabajar con AXI4-Lite debido a que la aplicación no requiere enviar múltiples datos a la vez; y, además, porque presenta una interfaz simplificada al tener menos señales en sus canales de comunicación en comparación a AXI4 (Full), lo que reduce el espacio que ocupa en la PL. Adicionalmente, se escogen 5 registros de la interfaz AXI4-Lite para modificar las características de las ondas modulantes y portadora en la FPGA. En Zynq se considera al PS como el maestro, por lo tanto, el modo de interfaz del bloque IP debe ser esclavo. Vivado permite crear y empaquetar en el bloque IP archivos HDL (VHDL o Verilog) con la lógica requerida para programar a la FPGA, incluyendo las interconexiones y protocolos AXI, así como, *drivers* para escribir y leer los registros en memoria.

El bloque IP creado se muestra en la Figura 2.16., sus salidas son conexiones físicas al puerto Pmod estándar (JE) y al led 0, como se mostró en la Figura 2.3., presentada en la página 20; y, sus entradas cuentan con la interfaz esclavo S00_AXI (bus de datos por

donde se comunica el bloque IP por medio de protocolos AXI), reloj del PS (s00_axi_aclk), y una señal reset (s00_axi_aresetn).

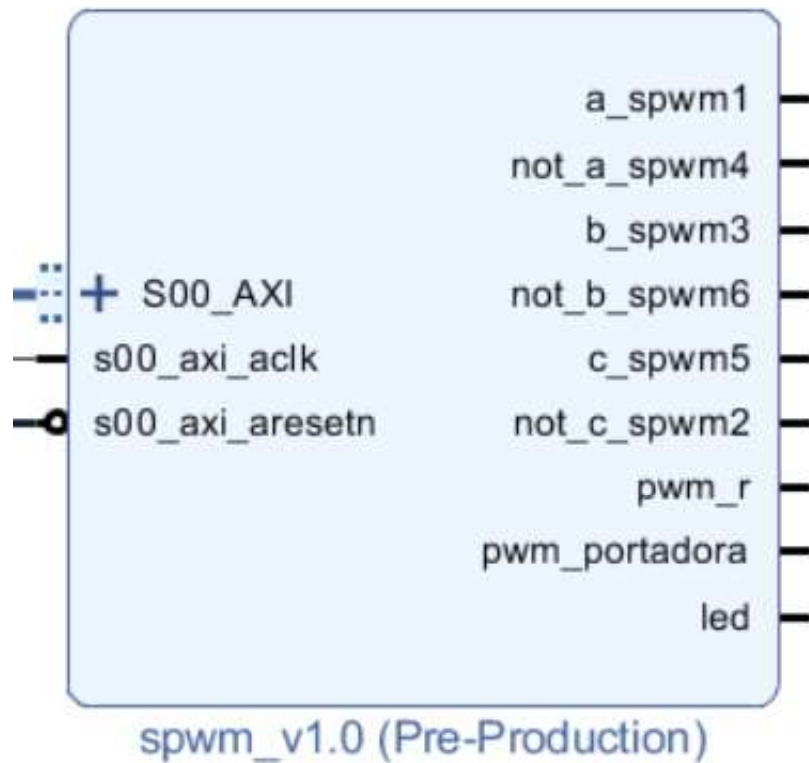


Figura 2.16. Bloque de propiedad Intelectual para aplicación SPWM

Los registros utilizados en la creación del bloque IP están definidos por su función en la Tabla 2.3. y también se puede observar gráficamente en la Figura 2.17. El nombre de cada registro está antepuesto por “slv” que quiere decir esclavo del registro (“reg n ”) correspondiente.

Tabla 2.3. Registros esclavos del bloque IP SPWM

Registros	Función
slv_reg0	Define la amplitud pico-pico de la onda portadora
slv_reg1	Define la amplitud entre pasos de la onda portadora
slv_reg2	Define el periodo de muestreo para las tablas de las ondas modulantes
slv_reg3	Define el offset para las ondas modulantes, de tal manera que siempre estén en el medio de la amplitud pico – pico de la onda portadora
slv_reg4	Define el periodo de muestreo de la onda portadora

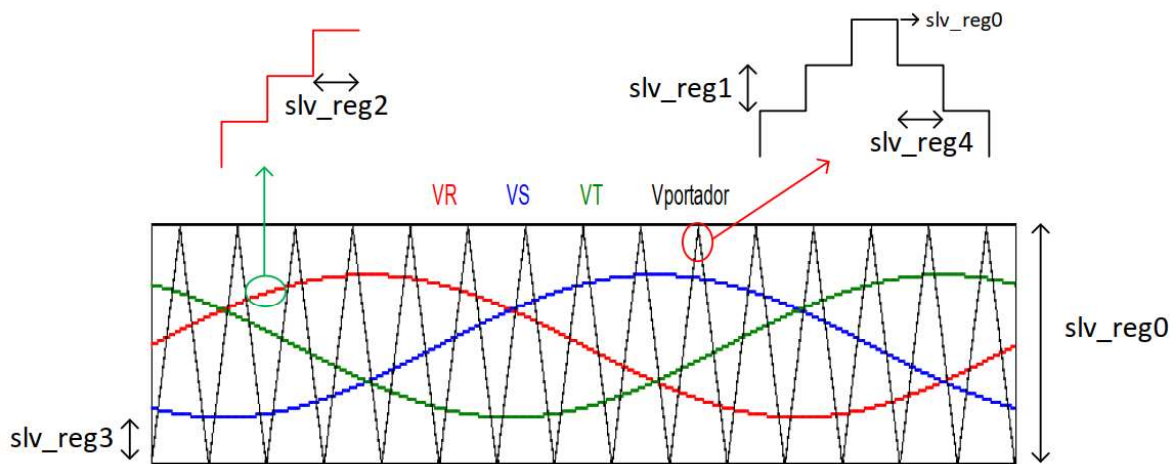


Figura 2.17. Ondas modulantes y portadora en función de los registros esclavos del bloque IP

Todas las funciones que se han explicado en los procesos de la PL se empaquetan en el bloque de propiedad intelectual junto a los procesos preconfigurados de comunicaciones por protocolos AXI, el cual presenta las siguientes ventajas:

- El bloque IP se lo utiliza como un bloque de programación en una interfaz gráfica en Vivado, lo que aumenta el nivel de abstracción para que el usuario final utilice la aplicación sin necesidad de conocer las descripciones de hardware realizadas.
- Permite reducir los tiempos de desarrollo de las aplicaciones para ser lanzadas al mercado.
- Puede ser comercializado, con licencias y regalías.

Como el usuario tiene acceso a los archivos de VHDL, el bloque de propiedad intelectual creado es clasificado como *soft-IP block* y RTL sintetizable.

Se necesita construir un sistema base de hardware del diseño en Zynq, donde se integre al PS con la PL, para crear el software de la interfaz con el usuario, por esto se debe especificar en el diseño de bloques que el sistema de procesamiento (PS) debe estar presente, así como el bloque de propiedad intelectual SPWM; ambos bloques son interconectados automáticamente por el entorno de desarrollo gráfico de Vivado para bloques IP por una interconexión AXI y un bloque de reinicio. El hardware final se observa en la Figura 2.18.

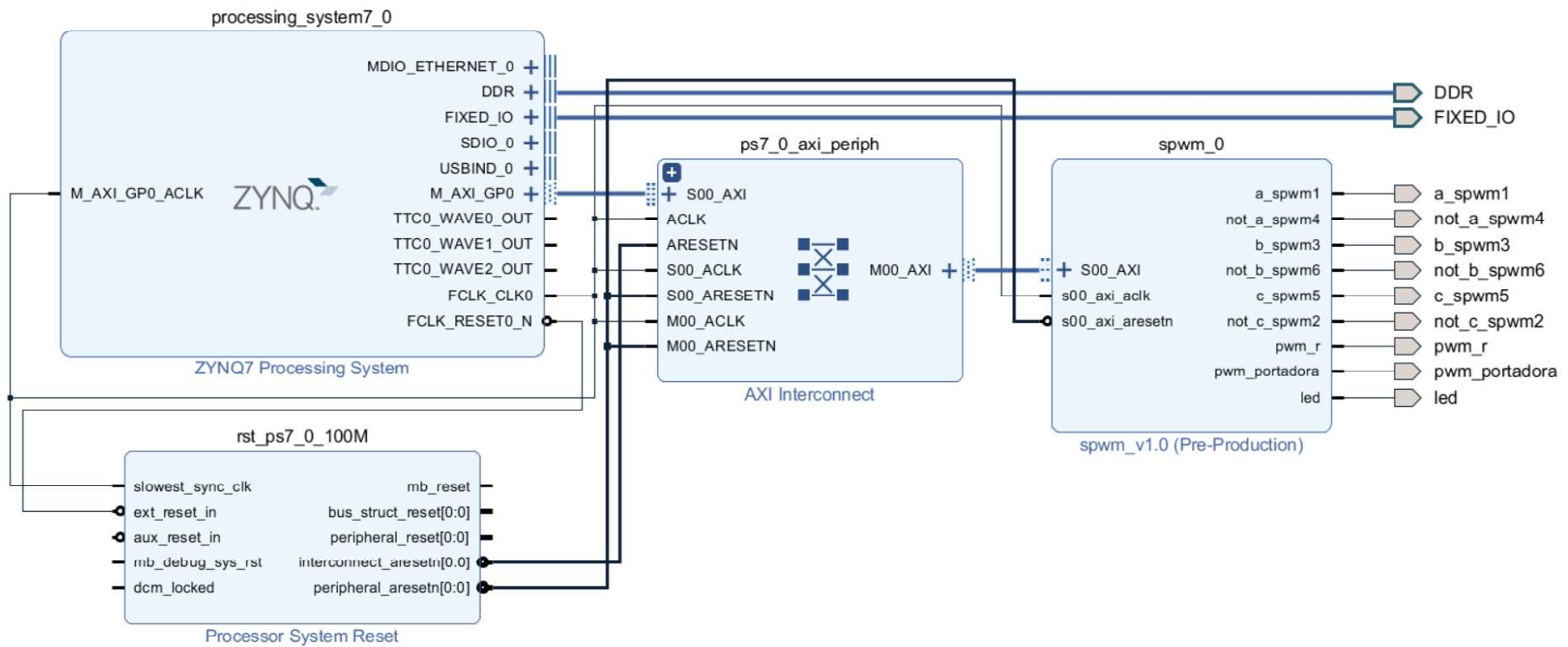


Figura 2.18. Sistema base de hardware

El diseño de bloques creado en Vivado es el sistema base de hardware del diseño en Zynq necesario para crear la aplicación de interfaz con el usuario y es sintetizado e implementado para generar el archivo *bitstream* (.bit) que programa a la PL.

2.4. PROGRAMACIÓN DEL SISTEMA DE PROCESAMIENTO

En este tema se describe como se realiza la programación del software del sistema de procesamiento de ZYBO para crear la interfaz con el usuario y el control del bloque de propiedad intelectual. A continuación, se expone el software que se utiliza en función de la estructura que crea para la aplicación; posteriormente se detalla cómo se realizan los algoritmos para controlar a ZYBO en función de las entradas del usuario.

El entorno de programación Software Development Kit (SDK) crea las capas de software para el sistema en chip Zynq, en la Figura 2.19. se muestran las 3 capas superiores para el software a desarrollar y la capa inferior para el hardware que ha sido exportado desde Vivado.

La capa inferior o sistema base de hardware, contiene el hardware sobre el cual se ejecutará el software y está constituida por todos los bloques observados en la Figura 2.18.

La capa de paquetes de soporte de plataformas (*BSP-Board support package*) es el intermediario entre el sistema operativo y el hardware, contiene librerías con funciones y drivers para los bloques IP de Vivado.

La capa de sistema operativo gestionará la ejecución de la aplicación de la capa superior con el BSP y es compatible con algunos softwares de empresas socias de Xilinx, el sistema operativo más destacable es Linux.

La capa más alta, aplicación de software, es donde se escribirán las sentencias de código de este trabajo en base a los lenguajes C/C++, con un nivel de abstracción que omite la necesidad de comprensión del hardware.

Se crea un sistema autónomo para uno de los procesadores ARM Cortex-A9 de Zynq, con el propósito de que la comunicación de la aplicación se produzca de manera directa con el hardware, esto implica que no se utilice una capa de sistema operativo, este tipo de aplicación es conocida como *bare-metal*. El archivo generado en función de este software es un formato ejecutable y vinculable (*Executable and Linkable Format*) (.elf), que es usado para programar al sistema de procesamiento (PS).

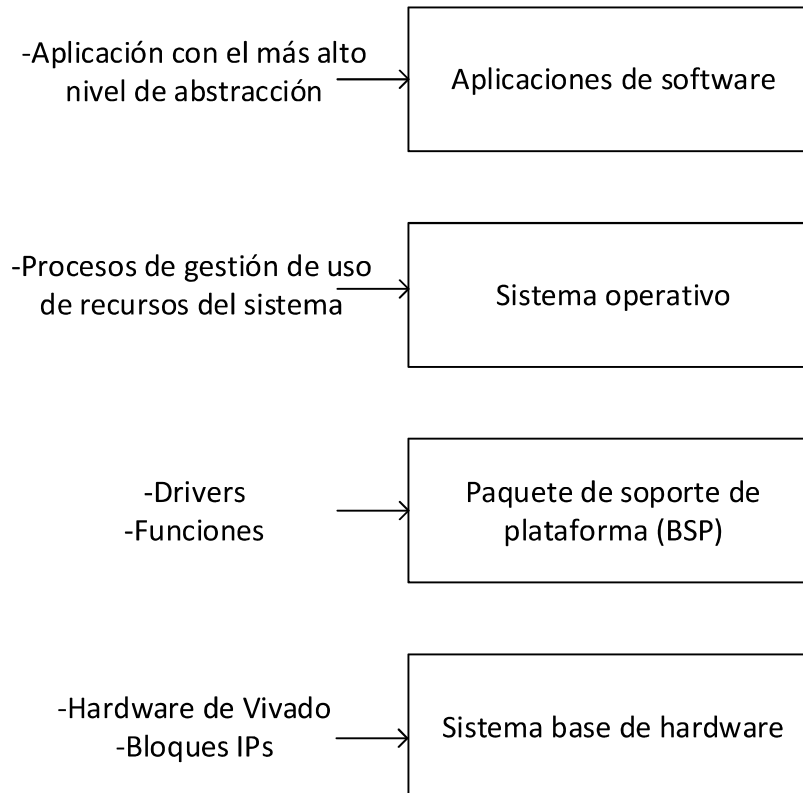


Figura 2.19. Capas de software y hardware para diseños del SOC Zynq

Vivado exporta el hardware a SDK para que se pueda crear el BSP e iniciar el desarrollo del software. Las entradas de la presente aplicación de software son la frecuencia fundamental de salida del inversor y el índice de modulación, que serán ingresados a través de una comunicación serial UART utilizando una interfaz gráfica de Matlab y el algoritmo de interfaz con el usuario desarrollado en el PS (Figura 2.20.); las salidas son las escrituras en las direcciones de memoria específicas del bloque de propiedad SPWM para los 5 registros de la Tabla 2.3., que en conjunto modifican las características de las SPWMs en la PL.

En la Figura 2.20. se observa un esquema general de los algoritmos que se procede a programar en el PS y como estos se vinculan con el bloque IP SPWM que se encuentra en la PL. Se observa que el software del PS está constituido por 6 funciones.

La primera función del PS se encarga de interactuar con el usuario para receptor los datos de frecuencia fundamental e índice de modulación. Esta función trabaja en conjunto con una interfaz gráfica. Tanto la función del PS como la interfaz gráfica son explicadas en la sección 2.5.

La segunda función del PS calcula los valores de los 5 registros esclavos que necesita el bloque IP para actualizar su punto de operación.

La tercera, cuarta y quinta función del PS se ponen en funcionamiento en función del error relativo del periodo de muestreo de la onda portadora y permiten modificar los parámetros calculados por la segunda función para crear en el bloque IP SPWM una onda portadora triangular con una adecuada resolución, estos algoritmos son explicados en las secciones 2.4.1., 2.4.2., 2.4.3.

Finalmente, la sexta función, entrega al bloque IP SPWM los valores calculados en el PS mediante protocolos AXI para modificar los 5 registros esclavos en el bloque IP SPWM.

Cuando los 5 registros esclavos del bloque IP SPWM son actualizados mediante los protocolos de comunicación AXI, los procesos de la sección 2.3. modifican el punto de operación de las señales SPWM.

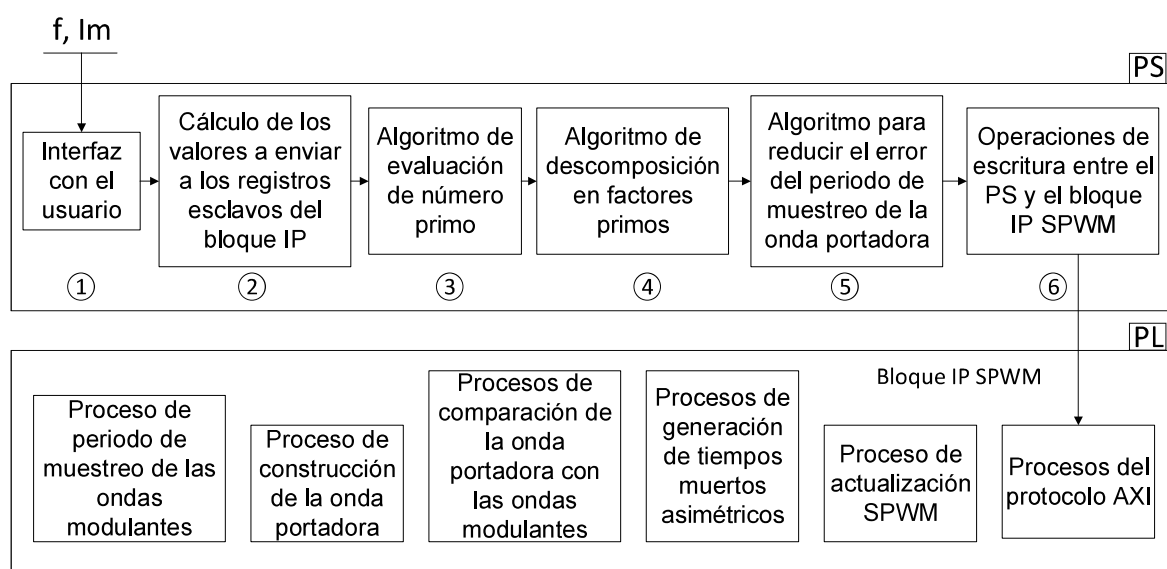


Figura 2.20. Arquitectura general de los algoritmos programados en el SOC ZYBO

Además de las entradas a ZYBO, también se puede configurar ciertos parámetros por el programador del software del SDK que pueden ser rápidamente modificables, tales como la relación de modulación de frecuencia y los límites superiores e inferiores de la frecuencia e índice de modulación que son ingresados por el usuario; esto se lo hace para proporcionar un software que sea rápidamente editable a fin de mantener la generalidad de la aplicación (el Anexo B, Guía de usuario, presenta los pasos para modificar el programa del PS). En la Tabla 2.4. se muestran los parámetros explicados, que en conjunto con las entradas al sistema definen totalmente a la modulación SPWM síncrona y sus rangos de funcionamiento.

Tabla 2.4. Constantes modificables en el software del PS

Constantes	Propósito
Relación de modulación de frecuencia (mf)	Con mf se modifica indirectamente la frecuencia de la portadora
Frecuencia de salida mínima (f_min)	La frecuencia mínima y máxima permite asegurar que la carga esté limitada dentro de los correctos rangos de funcionamiento
Frecuencia de salida máxima (f_max)	
Índice de modulación mínimo (Im_min)	Los índices de modulación mínimo y máximo crean un rango de trabajo para la aplicación, de esta manera se puede limitar el uso a una o todas las regiones (lineal, sobremodulación)
Índice de modulación máximo (Im_max)	

En base a las dos entradas del usuario, frecuencia fundamental e índice de modulación, se debe calcular el valor de cada registro a ser enviado a la PL. Los drivers del bloque IP SPWM contienen las direcciones de memoria para que el maestro, CPU del PS, inicie las transacciones de escritura y lectura definidas por los protocolos AXI, con el propósito de actualizar los registros esclavos en la PL.

Las Ecuaciones (2.3. - 2.7.) presentan los cálculos que son realizados por el maestro para asignar los valores a los registros de la Tabla 2.3. Los resultados de los cálculos de estas ecuaciones son adimensionales y sirven para configurar condiciones en la PL que equivalen a periodos de muestreo y amplitudes de voltaje:

$$A_{ppportadora} = \frac{A_{ppmodulante}}{Im} \quad (2.3)$$

$$T_mmodulantes = \frac{f_{FPGA}}{Nf_{inversor}} \quad (2.4)$$

$$Offset = \frac{A_{ppportadora} - A_{ppmodulante}}{2} \quad (2.5)$$

$$T_mportadora = \frac{f_{FPGA}}{f_{inversor} mf A_{ppportadora}} q = \frac{f_{FPGA}}{f_{inversor} mf \frac{A_{ppmodulante}}{Im}} q \quad (2.6)$$

$$Pasos = q \quad (2.7)$$

Donde:

- $A_{ppportadora}$ representa la amplitud pico-pico de la onda portadora.
- $A_{ppmodulante}$ es la amplitud pico-pico de una onda modulante.
- Im es el índice de modulación.
- f_{FPGA} es la frecuencia del reloj de la FPGA.
- $f_{inversor}$ es la frecuencia fundamental de salida del inversor.
- N es el número de muestras que tiene una tabla sinusoidal.
- $T_mmodulantes$ representa el periodo de muestreo de las ondas modulantes.
- mf es la relación de modulación de frecuencia.
- $T_mportadora$ representa el periodo de muestreo de la onda portadora.
- q es un factor que modifica al periodo de muestreo de la onda portadora cuando su error se vuelve apreciable, por defecto es 1.
- $pasos$ es el número que cuantifica la distancia en amplitud entre dos puntos de la onda portadora, por defecto es 1.

Si no se cumple con errores aceptables en el periodo de muestreo de la portadora, entonces 'q', $A_{ppportadora}$, $pasos$ y $T_mportadora$ son modificados en función de algoritmos que serán analizados posteriormente en las secciones de: ALGORITMO DE EVALUACIÓN DE NÚMERO PRIMO, ALGORITMO DE DESCOMPOSICIÓN EN FACTORES PRIMOS, ALGORITMO PARA REDUCIR EL ERROR DEL PERIODO DE MUESTREO DE LA ONDA PORTADORA; y, en el diagrama de flujo del software del PS. Para esta aplicación, se escogen como errores relativos porcentuales aceptables para el periodo de muestreo de la portadora a valores inferiores a 0.5%, en el programa del PS se puede modificar este parámetro con la constante "error_relativo_porcentual". En la Tabla 2.5. se presenta la relación que tiene cada registro con las entradas del programa.

Tabla 2.5. Relación de registros con las entradas del sistema

Parámetros para escribir en los registros esclavos del bloque IP de SPWM	Índice de modulación (Im)	Frecuencia fundamental de salida del inversor ($f_{inversor}$)
$A_{ppportadora} \rightarrow slv_reg0$	X	
$pasos \rightarrow slv_reg1$	X	X
$T_mmodulantes \rightarrow slv_reg2$		X
$Offset \rightarrow slv_reg3$	X	
$T_mportadora \rightarrow slv_reg4$	X	X

Se puede observar en la Tabla 2.5. que los parámetros relacionados con la construcción de la onda portadora están en función de las entradas, I_m y frecuencia fundamental, y que 3 de los 5 registros son utilizados para este propósito de construcción (slv_reg0, slv_reg1, slv_reg4).

Para modificar a las ondas modulantes en función de las entradas del sistema, simplemente se calculan los parámetros del periodo de muestreo y el offset, mediante la Ecuación 2.4. y Ecuación 2.5., respectivamente, y se redondea a un valor discreto. La modificación de los parámetros de las ondas modulantes es directa, es decir, lo que se calcula es lo que se escribe en los registros esclavos del bloque IP de SPWM; sin embargo, la Ecuación 2.3., Ecuación 2.6. y Ecuación 2.7. son la base para desarrollar algoritmos posteriores que modifican los resultados, en el caso de ser necesario, antes de enviar los datos.

Inevitablemente se producen errores de cuantificación en la programación, ya que los registros que se envían a la PL deben ser discretos. El objetivo de los algoritmos adicionales, que se presentan a continuación, es reducir los errores y disminuir la resolución de la onda portadora solo cuando exista un error considerable en el periodo de muestreo para la onda portadora. Se presta especial atención a esta onda, debido a que este trabajo utiliza a la portadora como señal variable en amplitud y tiempo, cuando se modifica el índice de modulación y la frecuencia, respectivamente.

2.4.1. ALGORITMO DE EVALUACIÓN DE NÚMERO PRIMO

La división en puntos para la construcción de la onda portadora está en función de su amplitud pico-pico. Si el cálculo de la Ecuación 2.3. redondeado tiene como resultado un número primo, se presenta el inconveniente de que la amplitud pico-pico de la portadora no tiene una divisibilidad que permita tener puntos exactamente equidistantes en amplitud. El problema es que, si se desea dividir a este número primo para tener puntos para representar a la portadora en su amplitud, se obtendrán resultados con fracciones y la precisión se perderá porque solo se puede representar números enteros; sin embargo, si se asegura que el resultado del punto máximo de la portadora no sea primo, se lo puede dividir en un número exacto de puntos.

El algoritmo de descomposición en factores primos del bloque 2 de la Figura 2.21. requiere tener como entrada una amplitud que no sea un número primo para poder verificar la divisibilidad para otros números enteros, por tanto, el bloque 1 debe asegurar que su entrada sea filtrada para proporcionar el requerimiento al siguiente bloque.

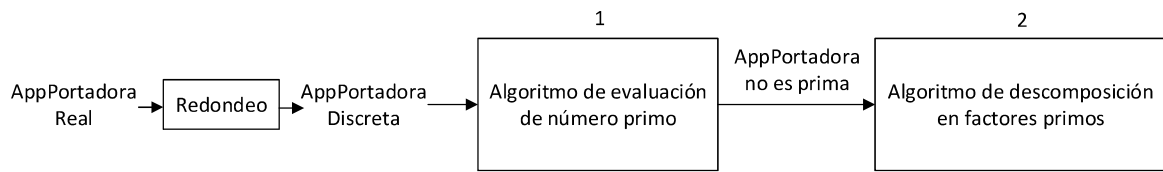


Figura 2.21. Diagrama de bloques de los algoritmos de número primo y descomposición en factores

La función del bloque 1 es identificar si su entrada es un número primo, si lo es, entonces decide incrementar o decrementar una unidad en función del número más cercano a la amplitud pico-pico de la portadora real (resultado de la Ecuación 2.3. incluyendo decimales). Como los únicos números primos consecutivos son el 2 y el 3, se garantiza que al sumar o restar una unidad no se tendrá como resultado otro número primo, debido a que el rango de operación de la amplitud pico-pico está muy alejada de dos y tres.

2.4.2. ALGORITMO DE DESCOMPOSICIÓN EN FACTORES PRIMOS

Ahora que se asegura que el número de la amplitud pico-pico de la portadora no es primo, se pueden obtener más divisiones exactas del mismo, que equivalen a tener más puntos para la construcción de la onda portadora. Este algoritmo analiza la divisibilidad del valor máximo de la portadora, guardando en un vector los factores primos que son divisibles, como se muestra en la Figura 2.22., de esta manera se proporciona un mecanismo para contar en pasos, los mismos que están en función de los factores del vector descrito, concluyendo que se tiene una representación exacta de amplitudes entre cada punto de la portadora hasta llegar al pico máximo. Es necesario recalcar que se ha llegado a este algoritmo debido a que el conteo por defecto, de puntos de uno en uno en amplitud, produce un error significativo en el periodo de muestreo de la portadora, este análisis se explicará en el siguiente algoritmo (2.4.3. ALGORITMO PARA REDUCIR EL ERROR DEL PERIODO DE MUESTREO DE LA ONDA PORTADORA).

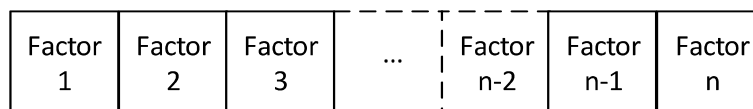


Figura 2.22. Vector de factores primos de la amplitud pico-pico de la onda portadora

En la Figura 2.23. se presenta un ejemplo en que el punto máximo de la portadora resulta ser 1050, a partir del vector de factores se puede indicar un número adecuado de pasos a utilizar que permitan alcanzar el valor requerido sin error, como se indica en la curva que está a la izquierda de la figura; y, en la curva que está a la derecha se muestra una

construcción de la portadora con pasos que no son función del vector de factores, observándose que existe un error en el resultado.

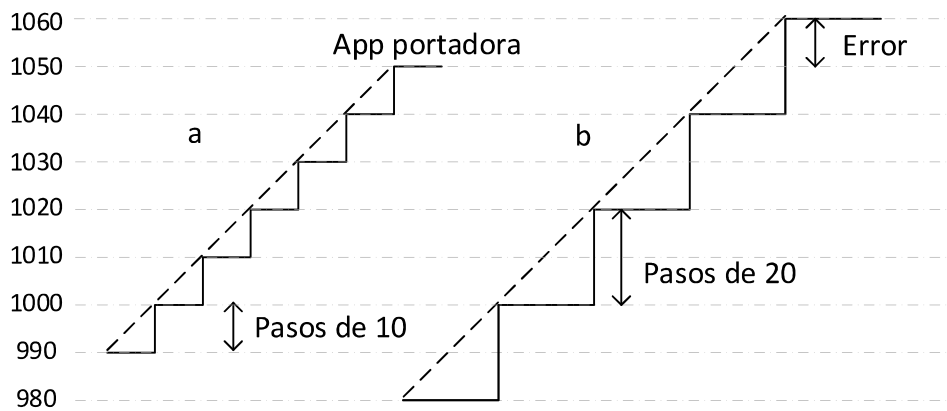


Figura 2.23. Conteo de la amplitud pico-pico de la portadora para distintos pasos

2.4.3. ALGORITMO PARA REDUCIR EL ERROR DEL PERIODO DE MUESTREO DE LA ONDA PORTADORA

La Ecuación 2.6. con el factor por defecto $q=1$ representa el periodo de muestreo de la onda portadora para cambios en amplitud de uno en uno, como se indica en la Figura 2.25. (a). La frecuencia de muestreo de la portadora es variable para cada nueva entrada del usuario y se incrementa proporcionalmente con: la frecuencia fundamental de salida del inversor, relación de modulación de frecuencia y la amplitud pico-pico de la portadora. El resultado de la Ecuación 2.6., en realidad, es el límite hasta donde la FPGA contará sus flancos positivos del reloj para indicar que se ha llegado al adecuado periodo de muestreo. El valor calculado del periodo de muestreo se redondea para ser implementable en la PL. Cuando el conjunto de todas las variables de la Ecuación 2.6., excepto 'q', provoca un resultado que se encuentra dentro de un rango cercano al periodo del reloj de la FPGA, como se muestra en la Figura 2.24., se presenta el inconveniente de que el error relativo porcentual, entre el valor implementable y el calculado, es grande, por estar tan próximo a la frecuencia del reloj de la FPGA, este mismo error significativo también se propaga a la frecuencia de la onda portadora.

Es indeseable tener un periodo de muestreo de la portadora tan cercano al periodo del reloj de la FPGA, ya que se incrementará el error por las fracciones que no pueden ser implementadas, la solución a esta desventaja es aumentar el periodo de muestreo de la portadora lo suficiente para que el error disminuya a un valor aceptable.

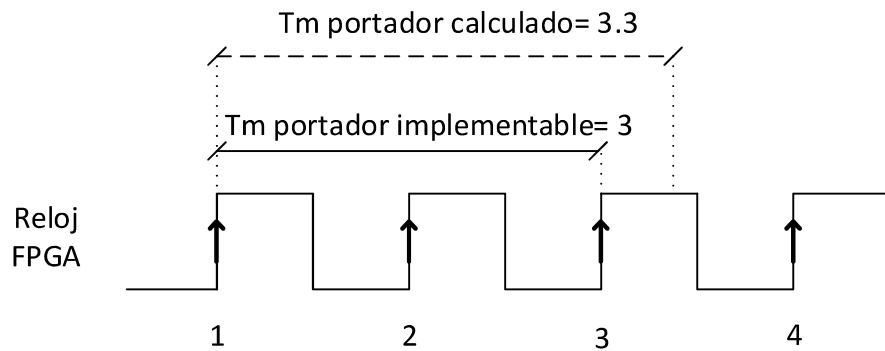


Figura 2.24. Periodo de muestreo de la onda portadora implementable

El algoritmo de esta sección utiliza el vector de factores que se obtiene de la amplitud pico-pico de la portadora para agrandar el periodo de muestreo. Se multiplican los valores de cada posición del vector por el periodo de muestreo, hasta que el error sea aceptable. El producto de cada iteración es el factor 'q' de la Ecuación 2.6. y Ecuación 2.7. En la Figura 2.25. se indica con los ejemplos (b) y (c) el incremento de los periodos de muestreo y de manera proporcional también los pasos.

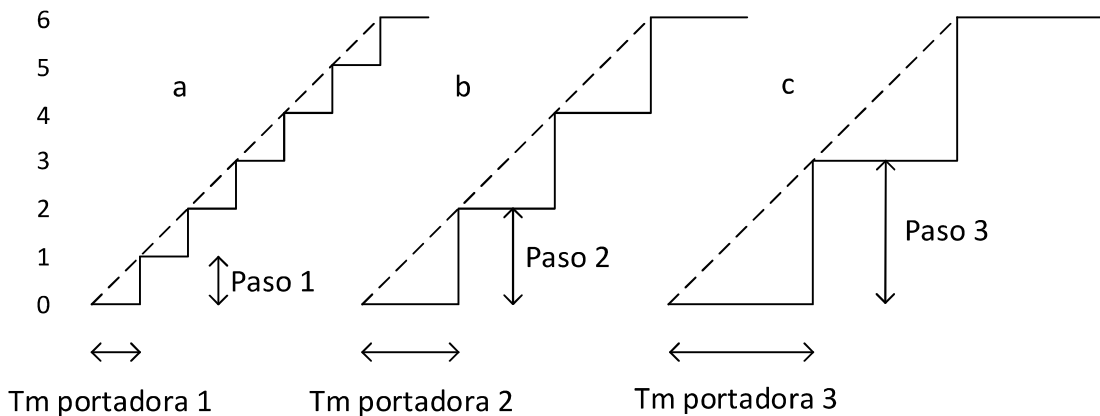


Figura 2.25. Construcción de la onda portadora mediante distintos periodos de muestreo

En esta onda periódica portadora se garantiza que por lo menos se va a tener 10 puntos en la construcción, cumpliendo con el criterio de Nyquist y con todos los algoritmos del PS explicados anteriormente; sin embargo, si no se puede cumplir el criterio mínimo de puntos con los factores primos divisibles de la amplitud pico-pico, se opta por empezar incrementando 'q' de uno en uno hasta que se cumplan las condiciones del error permitido para el periodo de muestreo y el criterio de Nyquist.

Los tres algoritmos, que modifican los parámetros de los registros de la onda portadora, actúan en ocasiones puntuales y se encuentran listos para brindar el soporte necesario.

2.5. INTERFAZ GRÁFICA

La interfaz con el usuario se realiza en conjunto entre una interfaz gráfica de Matlab, como se indica en la Figura 2.26., y el programa del PS de ZYBO. La interfaz gráfica de Matlab se conecta por comunicación serial UART a ZYBO con los siguientes parámetros de configuración:

- *Baudrate* = 115200.
- Datos = 8bits.
- Bits de parada = 1.
- Bits de paridad = ninguno.
- Control de flujo = ninguno.
- *Timeout* = 1 s.

El PS de ZYBO se programa para interactuar con el usuario, su acción es esperar a que se ingrese un dato y mostrar un resultado, ya sea una respuesta de la variable que ha sido guardada en el SOC o un mensaje indicando que el dato ingresado está fuera de los rangos predefinidos para la aplicación (Tabla 2.4.). Los datos de entrada al SOC son la frecuencia de la fundamental e índice de modulación, que pueden ser ingresados utilizando terminales configurados como comunicación UART. Se decide utilizar una interfaz gráfica, como alternativa a los terminales disponibles en la web, debido a su facilidad de uso al eliminar la configuración manual de los parámetros de comunicación serial. La presentación visual de la interfaz está diseñada bajo las siguientes características:

- Color gris de fondo para evitar cansancio en la visión del usuario.
- Utilización de colores para letras que no presenten un alto contraste con el fondo.
- Utilización de la tendencia a leer de izquierda a derecha y de arriba hacia abajo para influir en el comportamiento lector del usuario, que le permita seguir un orden lógico de conexión, ingreso y lectura de datos, y finalmente desconexión.

La interfaz gráfica interactúa con el usuario indicándole el dato que debe ingresar, si el dato ingresado no es un número entero, solicita que se ingrese un dato válido otra vez; además, indica las respuestas de ZYBO en el recuadro “Variables del sistema”.

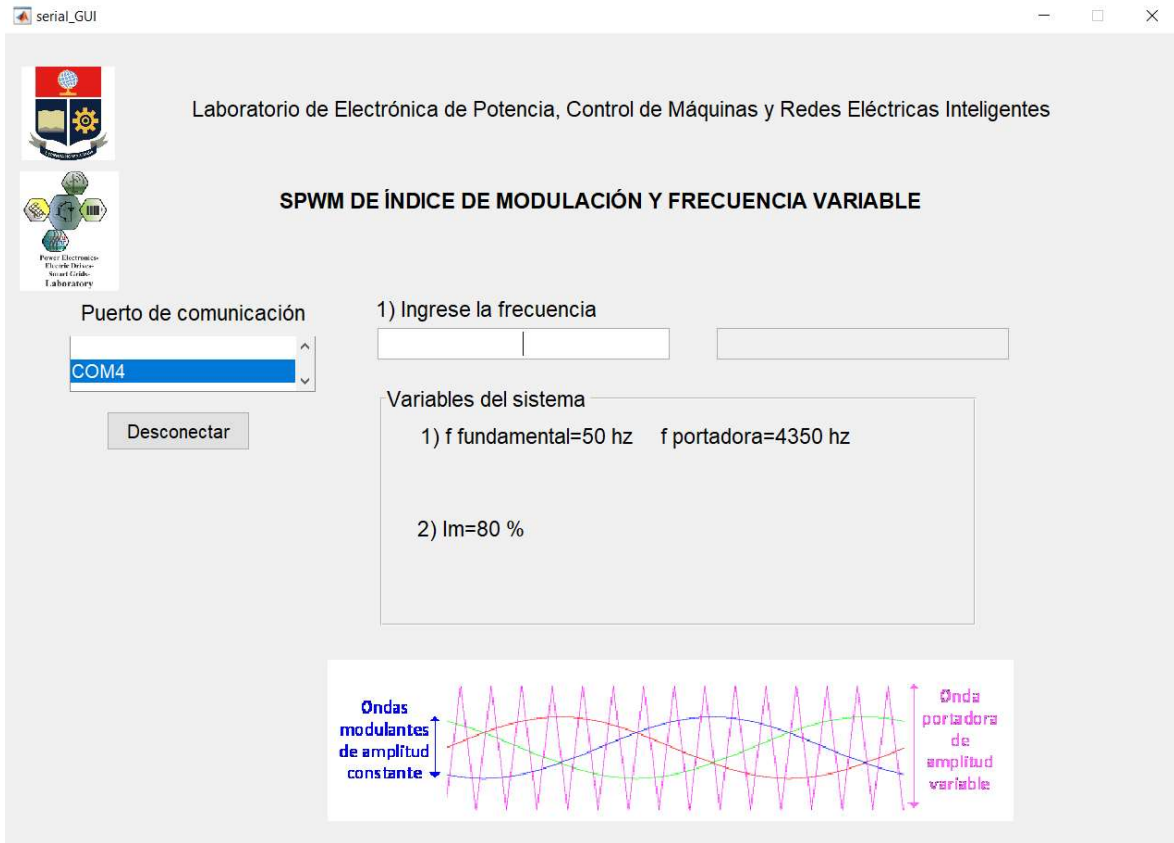


Figura 2.26. Interfaz gráfica

Existen dos recuadros de ingreso de datos, el primero identificado con el numeral 1 corresponde a la frecuencia fundamental del sistema y el segundo identificado con el numeral 2 corresponde al I_m porcentual. Luego de introducir cada dato, este es confirmado por ZYBO dentro del recuadro “Variables del sistema” que mantienen los mismos numerales de identificación de los recuadros de ingreso de datos. En el recuadro “Variables del sistema” también se presenta, junto a la frecuencia fundamental, el dato de la frecuencia de la portadora.

2.6. DIAGRAMAS DE FLUJO

Esta sección describe la secuencia de eventos que ocasiona el ingreso de datos del usuario en ZYBO; y, en diagramas de flujo, toda la programación de la PL, el PS y la interfaz gráfica de Matlab.

Previo a la presentación de los diagramas de flujo, se describe una visión general de la secuencia de eventos que se llevan a cabo, luego que la interfaz gráfica es conectada con ZYBO:

1. El CPU del PS de ZYBO está listo para recibir el dato de frecuencia fundamental.
2. La interfaz gráfica del computador solicita el ingreso de la frecuencia fundamental.
3. El usuario digita la frecuencia fundamental que desea.
4. El CPU del PS de ZYBO recibe el dato de frecuencia por comunicación serial UART, lo guarda y reafirma al usuario el dato ingresado.
5. El CPU del PS de ZYBO espera recibir el dato de índice de modulación.
6. La interfaz gráfica del computador solicita el ingreso del índice de modulación.
7. El usuario digita el índice de modulación que desea.
8. El CPU del PS de ZYBO recibe el dato de f_m por comunicación serial UART, lo guarda y reafirma al usuario el dato ingresado.
9. El CPU del PS calcula los valores a enviar a los 5 registros esclavos del bloque IP SPWM de la PL. (Ecuaciones 2.3 – 2.7)
10. El CPU del PS evalúa el error del periodo de muestreo de la portadora, si es significativo, se reduce su error con algoritmos que permitan discretizar a la onda portadora con menor resolución, modificando los valores a enviar a los registros esclavos del bloque IP SPWM.
11. El CPU del PS envía los valores calculados a los 5 registros esclavos del bloque IP SPWM de la PL a través de la central de interconexiones AXI.
12. Las siguientes acciones se desarrollan de manera concurrente dentro del bloque IP SPWM en la PL:
 - El bloque IP maneja los protocolos de comunicación AXI. Por medio de protocolos de lectura se guarda la información que envía el PS a los 5 registros esclavos.
 - Se actualizan los puntos de operación recibidos del PS en el punto mínimo de la portadora.
 - Se construye la onda portadora.
 - Se construyen las ondas modulantes.

- Se comparan las ondas modulantes con la portadora para crear las 6 señales SPWMs, acorde a las condiciones dictadas por la modulación SPWM (Figura 1.5.).
- Se añaden tiempos muertos a las 6 señales SPWM.
- El puerto JE de ZYBO es actualizado con las 6 señales SPWMs con tiempos muertos
- El puerto JE de ZYBO es actualizado con las 2 señales PWMs con la frecuencia de la portadora y la mitad de la frecuencia de la onda modulante R.
- El led 0 es actualizado con el comportamiento de la SPWM S1.

Es importante notar que, en esta secuencia de eventos, la PL y el PS trabajan de manera concurrente entre ellos y que los puntos detallados se repiten indefinidamente.

En la PL existen procesos que realizan la misma función, pero con otras señales, esto aplica para la comparación de cada una de las modulantes con la portadora y para la generación de tiempos muertos asimétricos para cada una de las 6 SPWMs, por este motivo, a continuación, se presentan todos los diagramas de flujo y se decide explicar solo un proceso de comparación y uno de tiempo muerto:

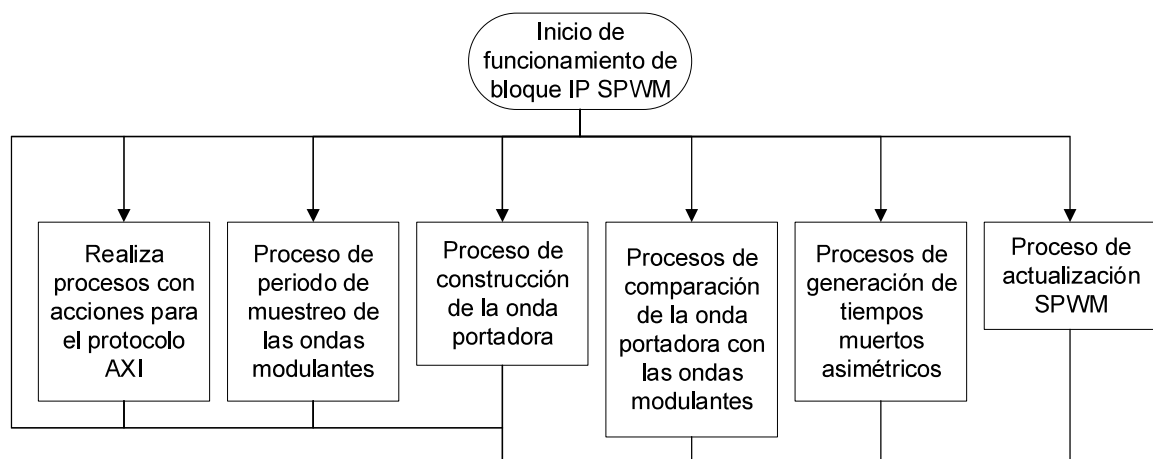


Figura 2.27. Diagrama de flujo del funcionamiento de bloque IP SPWM

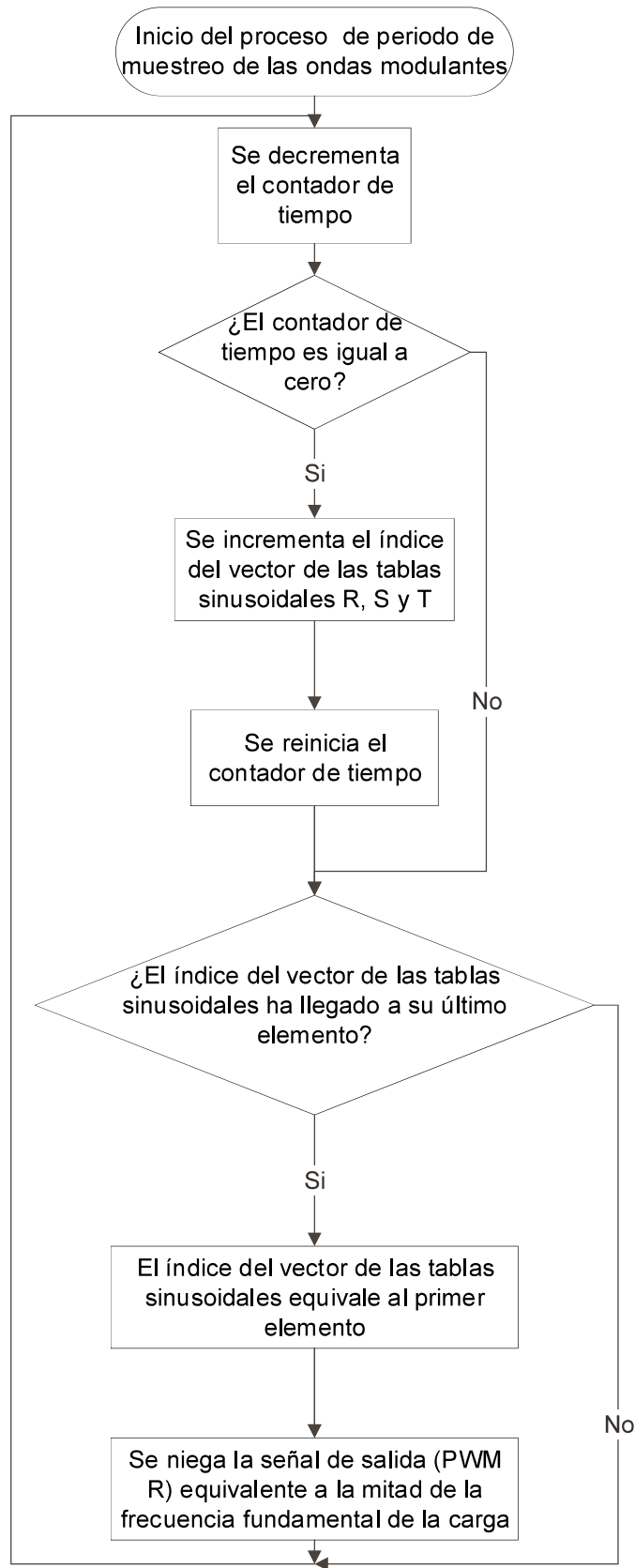


Figura 2.28. Diagrama de flujo del proceso de periodo de muestreo de las ondas modulantes

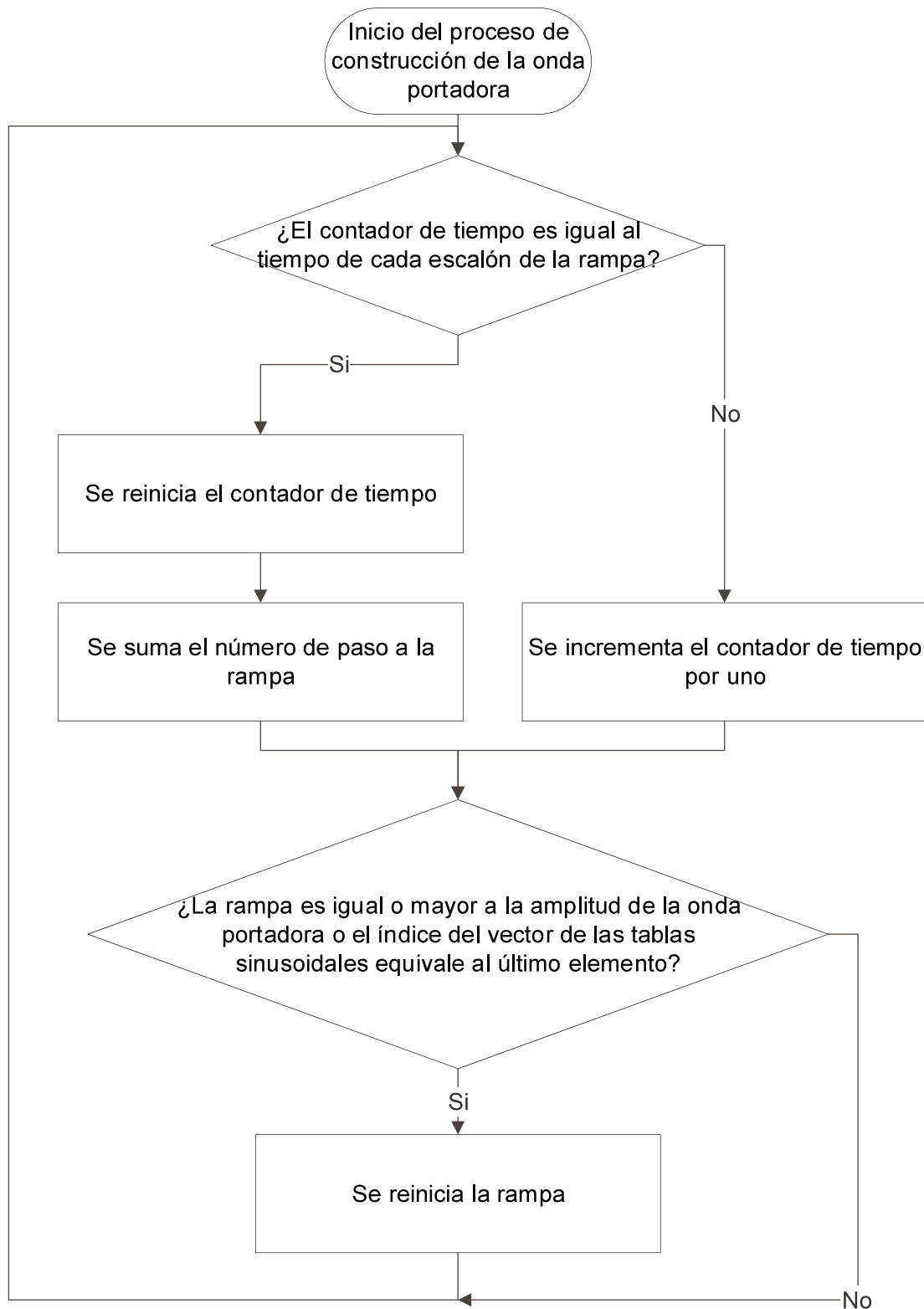


Figura 2.29. Diagrama de flujo del proceso de construcción de la onda portadora

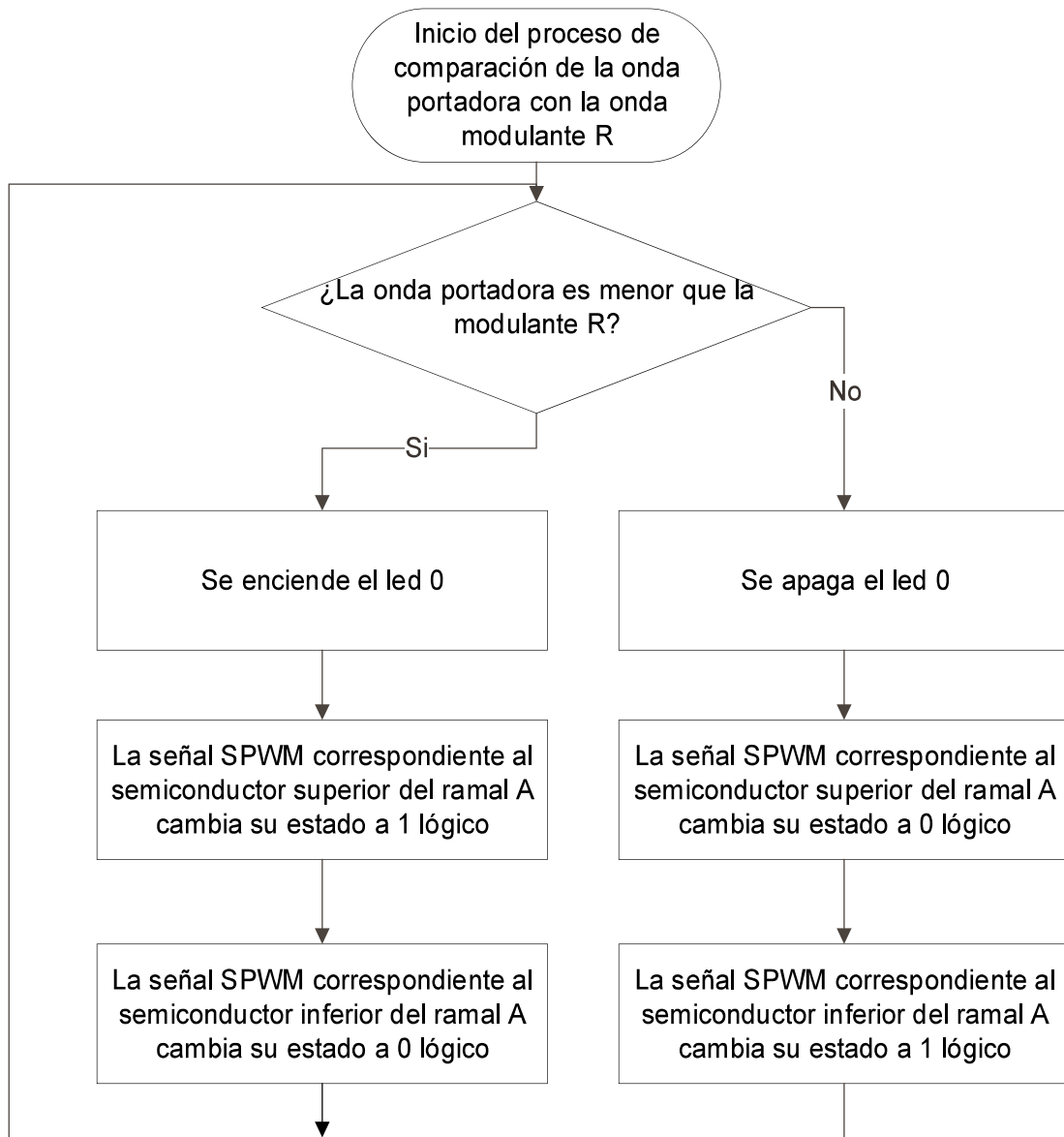


Figura 2.30. Diagrama de flujo del proceso de comparación de la onda portadora con la onda modulante R

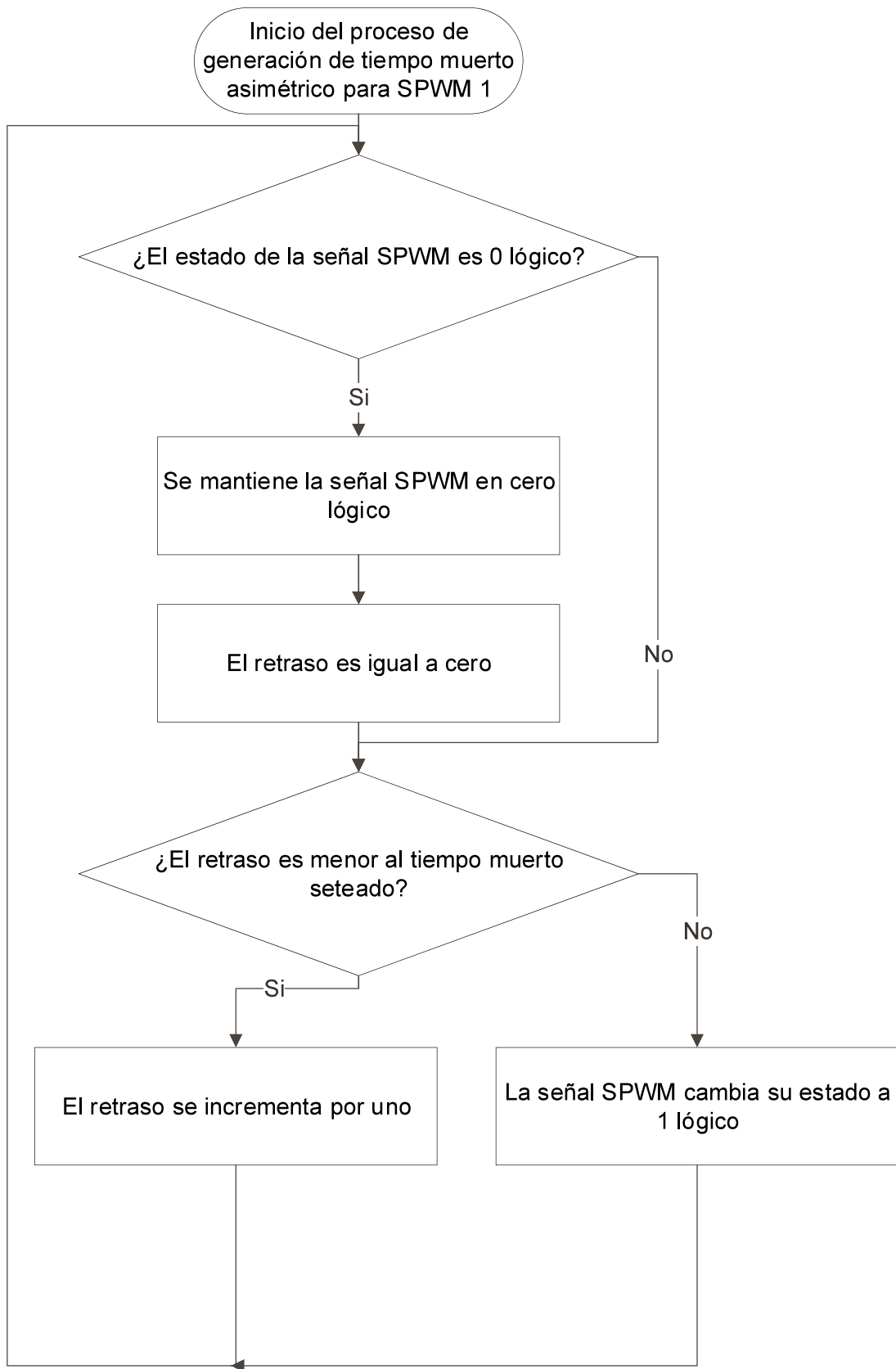


Figura 2.31. Diagrama de flujo del proceso de generación de tiempo muerto asimétrico para SPWM 1

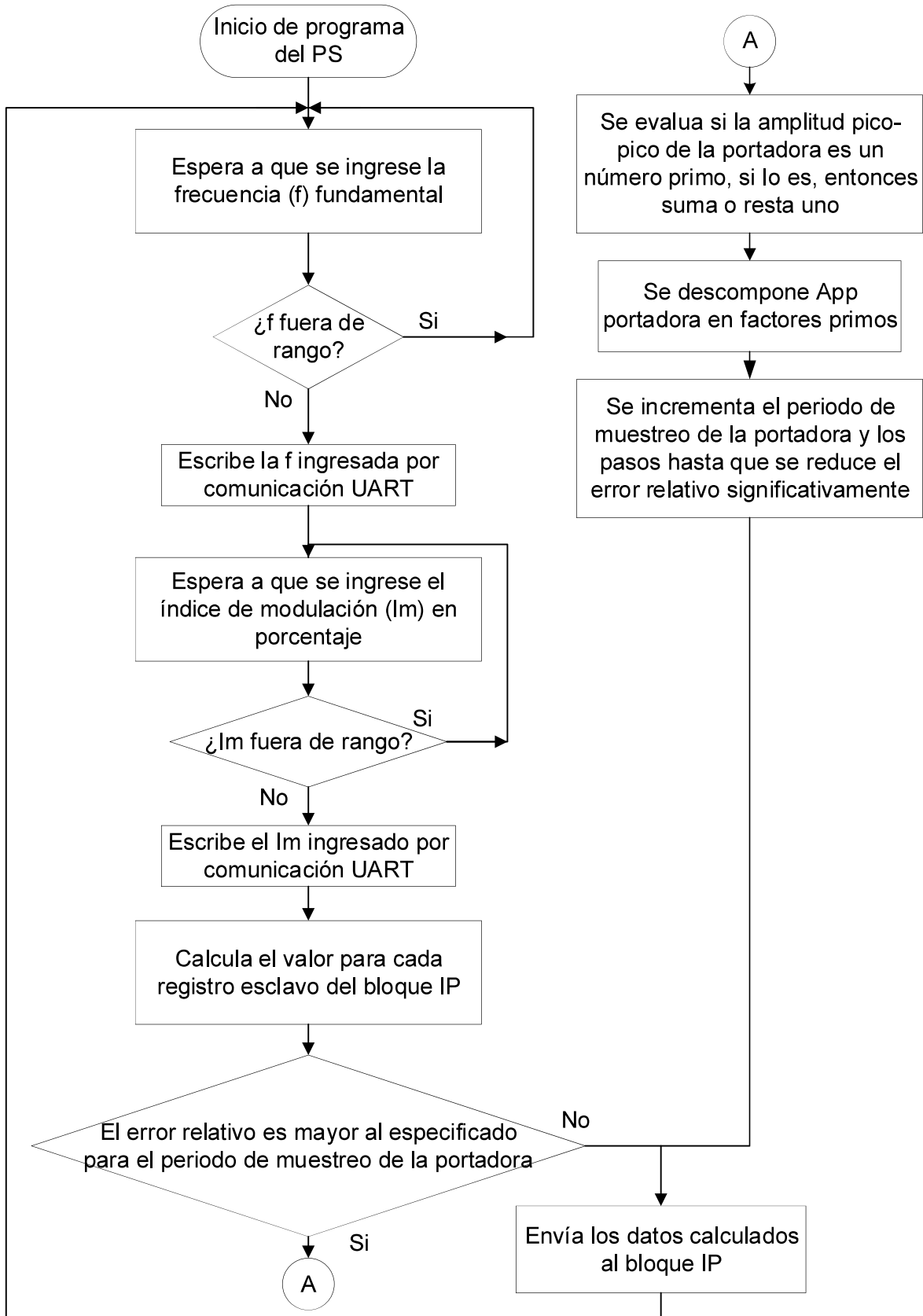


Figura 2.32. Diagrama de flujo del PS

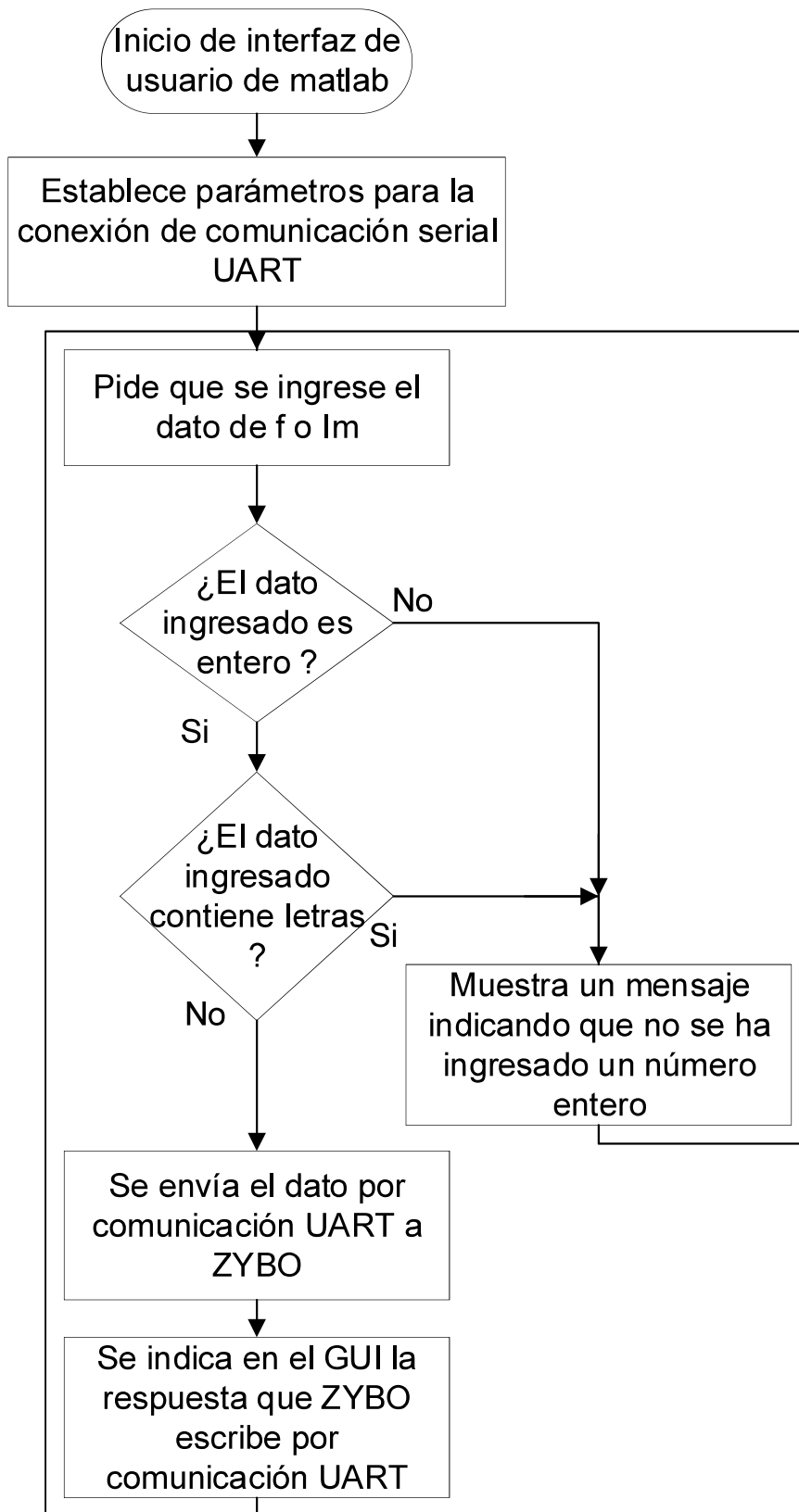


Figura 2.33. Diagrama de flujo de la interfaz de usuario de Matlab

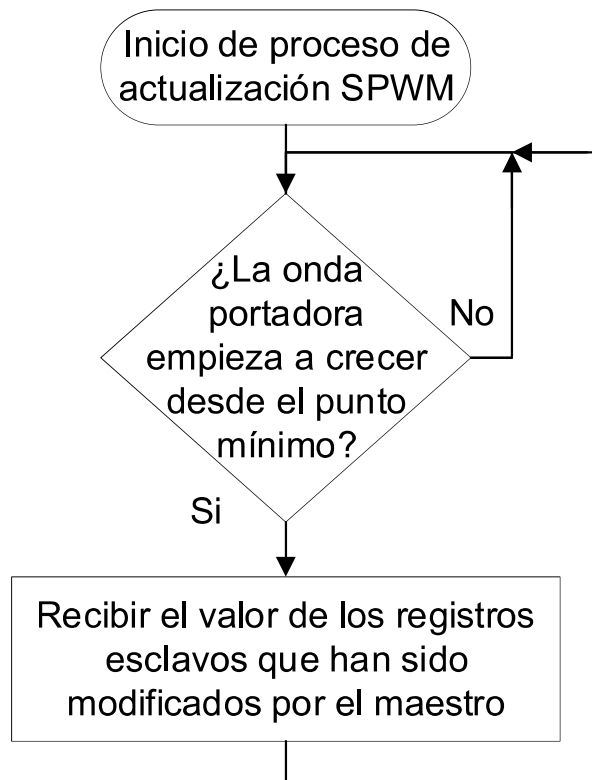


Figura 2.34. Diagrama de flujo del proceso de actualización SPWM

3. RESULTADOS Y DISCUSIÓN

En esta sección se presentan los resultados de los reportes del sistema digital de la tarjeta ZYBO, las simulaciones del comportamiento de las señales del bloque de propiedad intelectual SPWM y los resultados de las mediciones y validaciones realizadas en laboratorio utilizando un rectificador-inversor trifásico STEVAL-IHM023V3 [27] para alimentar un motor de inducción, carga RL y R.

3.1. REPORTES DE IMPLEMENTACIÓN DE VIVADO DESIGN SUITE

Se analizan los reportes de tiempo, utilización y potencia, generados a partir del proceso de implementación, para verificar que el diseño digital se comporte de la misma manera en que ha sido configurado y, además, comprobar que no se viole ninguna restricción de diseño.

3.1.1. REPORTE DE TIEMPO [23] [24] [25]

El reporte de tiempo es una herramienta que permite validar que el comportamiento del sistema digital, configurado en la PL, funcione exactamente como se lo ha diseñado sin presentar desviaciones no planificadas. En el reporte de tiempo se identifican los caminos de los circuitos que deben ser optimizados para que sus respectivos FFs no almacenen datos incorrectos. El adecuado análisis y corrección de los tiempos críticos del sistema digital aseguran que los datos guardados en los FFs siempre sean los correctos con el propósito de evitar crear condiciones de peligro para los equipos y el operador del sistema.

A continuación, se muestra cómo se realiza la validación del sistema digital configurado en ZYBO, valiéndose de las definiciones de los tiempos característicos de los FFs y de los reportes de tiempo.

3.1.1.1. *Setup time y hold time*

Para que un FF funcione adecuadamente, se requiere que el dato a ser almacenado dentro del mismo se mantenga estable dentro de un rango de tiempo. En la Figura 3.1., que muestra el almacenamiento de un dato en un FF, se puede observar el rango mínimo de tiempo en el cual el dato del FF debe mantenerse estable, esto es entre los cursores A y B, ubicados antes y después de la llegada del flanco del reloj.

Se define al *setup time* como la cantidad mínima de tiempo en la que el dato es mantenido estable en la entrada de un FF antes del flanco del reloj para que sea confiablemente almacenado. Este tiempo se extiende desde el cursor A hasta el flanco positivo del reloj y se lo representa por *tsu*. [23]

El *hold time*, representado por *th*, se define como la cantidad mínima de tiempo en la que el dato es mantenido estable en la entrada de un FF después del flanco del reloj para que sea confiablemente almacenado. Se lo mide desde el flanco positivo del reloj hasta el cursor B. [23]

Tanto el tiempo mínimo *tsu* como *th* deben ser respetados en todos los FFs del sistema digital, es decir, el dato de entrada del FF no debe fluctuar dentro del rango de tiempo delimitado por los cursores A-B.

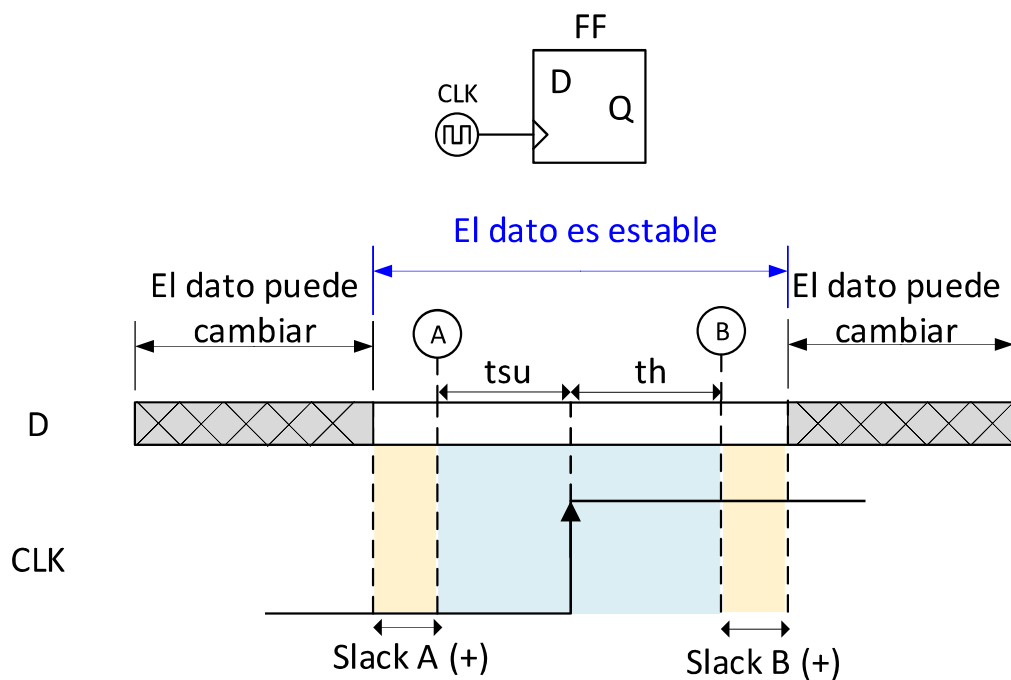


Figura 3.1. Dato almacenado en FF sincronizado con el flanco positivo del reloj

3.1.1.2. *Slack time*

Es el tiempo en que el dato del FF extiende su estabilidad a la izquierda del cursor A (*Slack A (+)*, Figura 3.1) y/o a la derecha del cursor B (*Slack B (+)*, Figura 3.1); o, es el tiempo en que el dato del FF viola la estabilidad a la derecha del cursor A (*Slack A (-)*, Figura 3.2.) y/o a la izquierda del cursor B (*Slack B (-)*, Figura 3.2.). Los tiempos *slack* son positivos cuando se respetan los tiempos *tsu* y *th*, y son negativos cuando se los viola.

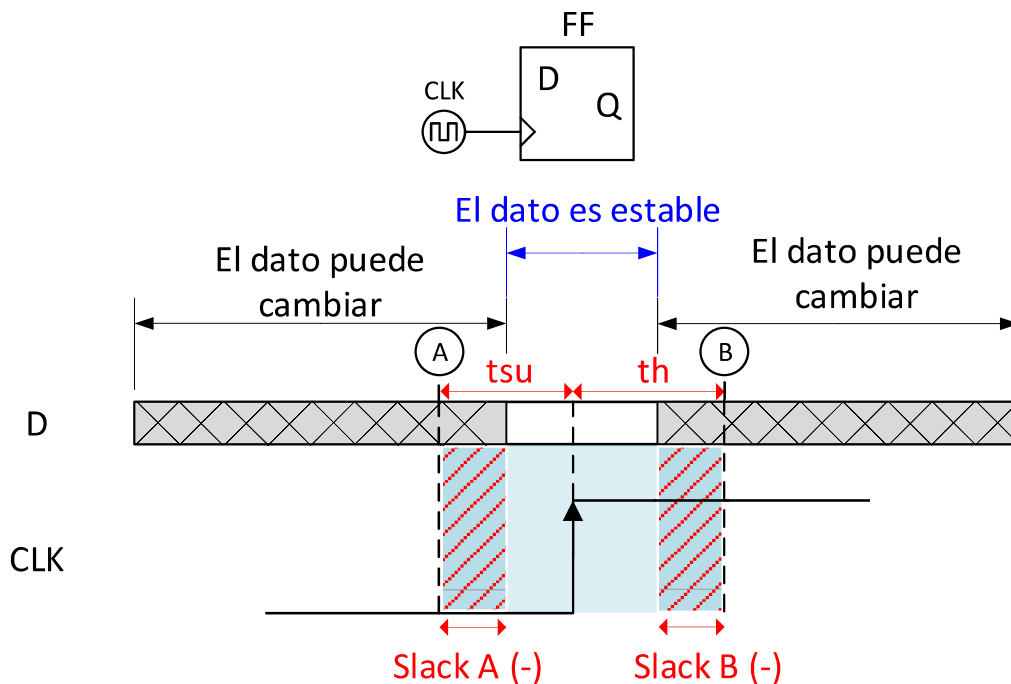


Figura 3.2. Slack time A y B que violan los tiempos t_{su} y t_h

3.1.1.3. Meta-estabilidad

La meta-estabilidad es la incapacidad de mantener al dato del FF estable durante el rango mínimo de tiempo, es decir, el tiempo total de t_{su} y t_h . Si se produce una condición de meta-estabilidad, existe la probabilidad de que el FF no almacene el dato correcto y adquiera arbitrariamente el valor de 1L o 0L, sin respetar el dato que debe ser almacenado de acuerdo con el algoritmo diseñado.

La meta-estabilidad podría generar un error en la lectura de datos y un mal funcionamiento del sistema digital configurado, ocasionando situaciones de peligro en el funcionamiento del motor de inducción manejado por ZYBO, tales como:

- La transición brusca entre dos puntos de operación.
- Cortocircuitos al alterarse las señales de los procesos de los tiempos muertos.

La razón más común para no cumplir con el *setup time* es la presencia de caminos críticos. Un camino crítico es un circuito que por su considerable cantidad de compuertas lógicas presenta una trayectoria larga de procesamiento de un dato entre dos FFs y produce un tiempo de propagación que viola o está a punto de violar el tiempo t_{su} . La Figura 3.3. presenta el esquema de un camino crítico que comienza en Q1, pasa a través del circuito combinacional A con un tiempo de propagación significativo y finalmente llega a la entrada D2.

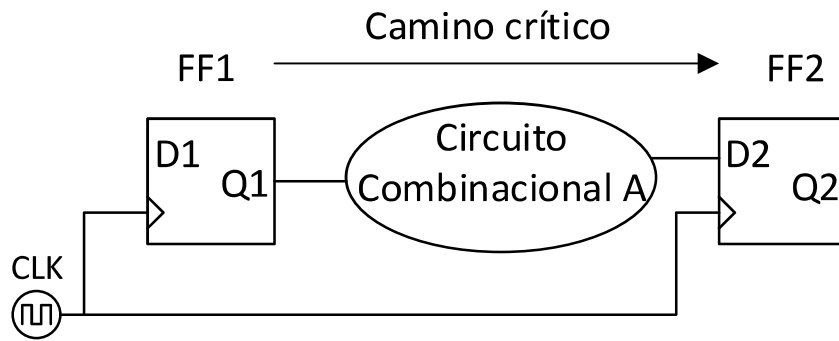


Figura 3.3. Camino crítico entre dos FFs

En la Figura 3.4. se muestra un ejemplo de un escenario en el que existe la posibilidad de no cumplir con el *hold time*. Se observa que D4 es alimentado directamente por Q3, esto implica que casi no existen retardos entre estos dos puntos. Cuando el flanco positivo del reloj llega a FF3 y FF4 al mismo tiempo, D4 debe tener un valor constante al menos por el tiempo t_h , sin embargo, Q3 empieza a variar casi instantáneamente ocasionando que D4 fluctúe antes de haber pasado el tiempo t_h , generando una condición de meta-estabilidad.

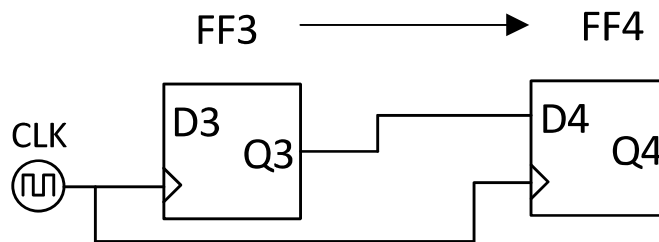


Figura 3.4. Camino directo entre la salida y entrada de FFs en serie

3.1.1.4. Soluciones a las violaciones de *setup time* y *hold time*

Los softwares que se utilizan para programar a las FPGA por lo general se encargan de hacer cumplir el *hold time*, sin embargo, las violaciones de *setup time* deben ser solucionadas por el diseñador del sistema digital. Las soluciones más comunes para eliminar una violación de *setup time* son:

- Optimizar el circuito combinacional del camino crítico para que su tiempo de propagación se reduzca.
- Aplicar la técnica de *pipelining*, que consiste en utilizar un FF intermedio que divide al circuito combinacional en dos partes, como se indica en la Figura 3.5, consiguiendo con esto la distribución del tiempo de propagación original del circuito combinacional A presentado en la Figura 3.3.

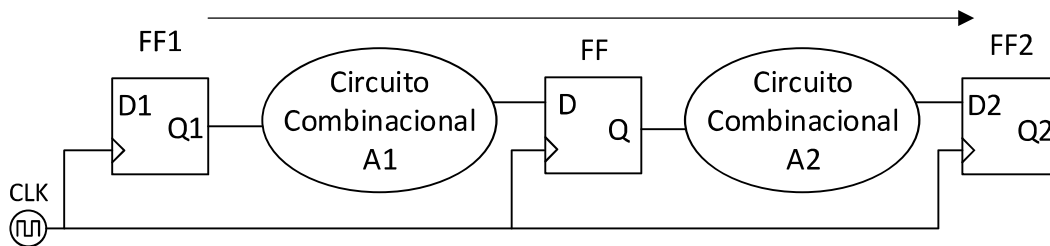


Figura 3.5. *Pipelining*

3.1.1.5. *Worst Negative Slack*

Pueden existir varios caminos críticos en el diseño del sistema digital, en la Figura 3.6. se muestra un estado de meta-estabilidad ocasionado por la violación al *setup time* por el peor camino crítico de todo el sistema digital, considerado como el trayecto que presenta el circuito combinacional con mayor tiempo de propagación, donde su salida, que a la vez es la entrada de un FF, fluctúa luego de ingresar a la zona delimitada por los cursores A-B (zona de color celeste), impidiendo cumplir con el tiempo *tsu*.

Este tiempo es conocido como *Worst Negative Slack* (WNS), zona marcada con líneas diagonales de color rojo en la Figura 3.6., el cual refleja por cuanto tiempo se ha violado a *tsu*, y es representado con signo negativo debido a que se encuentra en un estado de meta-estabilidad.

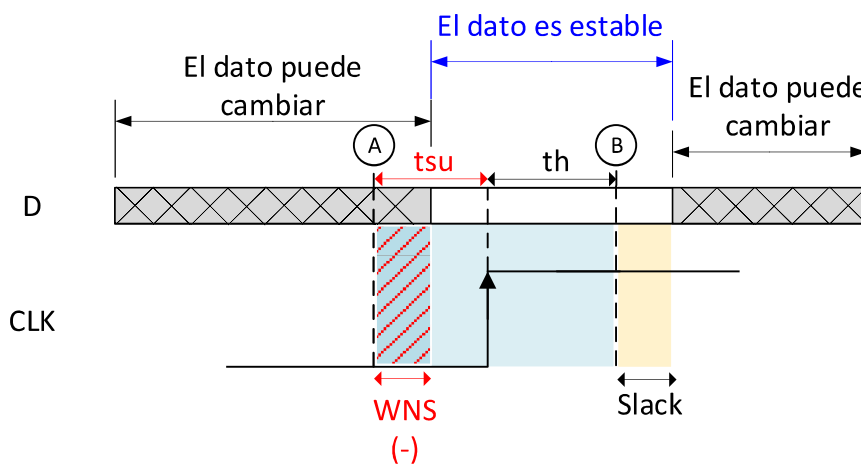


Figura 3.6. Estado de meta-estabilidad ocasionado por el peor camino crítico

3.1.1.6. *Worst Hold Slack*

En la Figura 3.7. se muestra un estado de meta-estabilidad del peor camino que incumple con el *hold time*, mayor tiempo *slack* medido desde el cursor B hacia la izquierda.

Este tiempo es conocido como *Worst Hold Slack* (WHS), zona marcada con líneas diagonales de color rojo en la Figura 3.7., el cual refleja por cuanto tiempo se ha violado a t_h , y es representado con signo negativo debido a que se encuentra en un estado de meta-estabilidad.

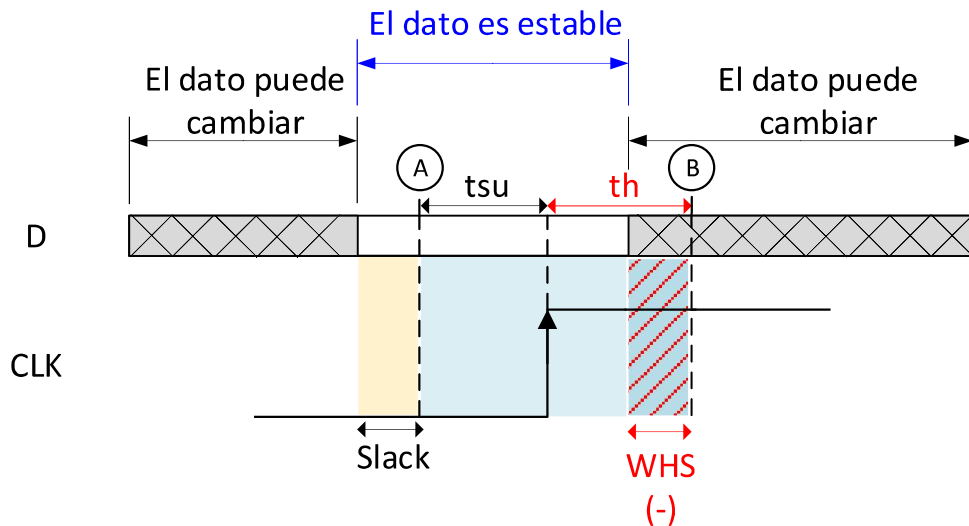


Figura 3.7. Peor estado de meta-estabilidad debido al no cumplimiento del *hold time*

3.1.1.7. *Total Negative Slack* y *Total Hold Slack*

La Figura 3.6. y la Figura 3.7. representan los peores tiempos que colocan a un camino específico en un estado de meta-estabilidad, sin embargo, pueden existir más caminos que violen la zona entre los cursores A-B en donde los datos de los FFs deben permanecer constantes. Al existir más condiciones de no estabilidad se introducen dos nuevos términos, *Total Negative Slack* (TNS) y *Total Hold Slack* (THS), que indican por cuánto tiempo total negativo se viola el *setup time* y *hold time* en todo el sistema digital, respectivamente.

3.1.1.8. Procedimiento básico de validación del comportamiento del algoritmo configurado

Los reportes de tiempo WNS, TNS, WHS, THS, producidos por el proceso de implementación, permiten validar el comportamiento del algoritmo configurado. Los siguientes pasos muestran el procedimiento básico de validación utilizado:

1. Describir a los circuitos digitales de la aplicación en VHDL.

2. Realizar el proceso de síntesis e implementación en el software Vivado Design Suite.
3. Revisar que WNS, TNS, WHS, THS presenten valores positivos o cero, es decir, que no violen ningún *setup time* y/o *hold time*. Si se llega a violar alguno de estos tiempos, rastrear el o los caminos comprometidos mediante el software e identificar la manera óptima de solucionar el problema.
4. Modificar, en el código descrito en VHDL, los caminos del circuito digital que violan los tiempos *setup time* y/o *hold time*.
5. Repetir los puntos 2 y 3. Si no se han solucionado los problemas con los tiempos, volver al paso 4.

La inobservancia de corregir las violaciones a los tiempos *setup time* y/o *hold time*, para asegurarse que no exista ningún FF en condiciones de meta-estabilidad, implica la inclusión de una o varias condiciones de peligro para el sistema inversor – motor y para el operador, ya que no se sabe en qué momento y por cuánto tiempo se desviará el algoritmo de su curso natural de funcionamiento; y, qué efectos adversos tendrá en el sistema. Por las razones mencionadas, es de extrema importancia revisar que no se viole ningún tiempo antes de proceder a conectar el diseño realizado en ZYBO a la electrónica de potencia.

3.1.1.9. Interpretación de los reportes de tiempo obtenidos en la aplicación SPWM desarrollada en ZYBO

Luego de realizar los pasos del procedimiento básico de validación del comportamiento del algoritmo, se obtienen los tiempos indicados en la Tabla 3.1., donde se presentan los reportes de tiempos del proceso de implementación del diseño digital: *Worst Negative Slack*, *Total Negative Slack*, *Worst Hold Slack* y *Total Hold Slack*.

Tabla 3.1. Reporte de tiempos de implementación

WNS	TNS	WHS	THS
0.068 ns	0 s	0.034 ns	0 s

En la Figura 3.8. y Figura 3.9. se muestran el WNS y WHS de la Tabla 3.1. para el camino crítico con mayor tiempo de propagación y camino más rápido, respectivamente. Se observa que la región de tiempo *slack* que ocupan es positiva (fuera del rango de tiempo delimitado por los cursores A-B), por lo tanto, no se viola ninguna restricción de tiempo.

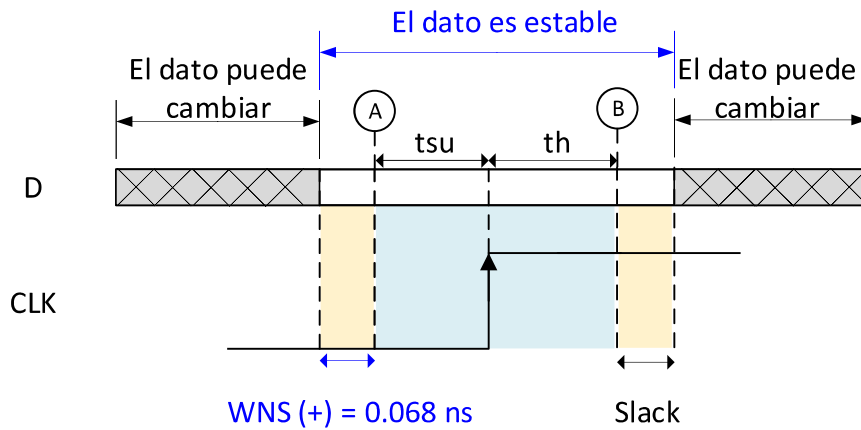


Figura 3.8. WNS del camino crítico más significativo en el sistema digital

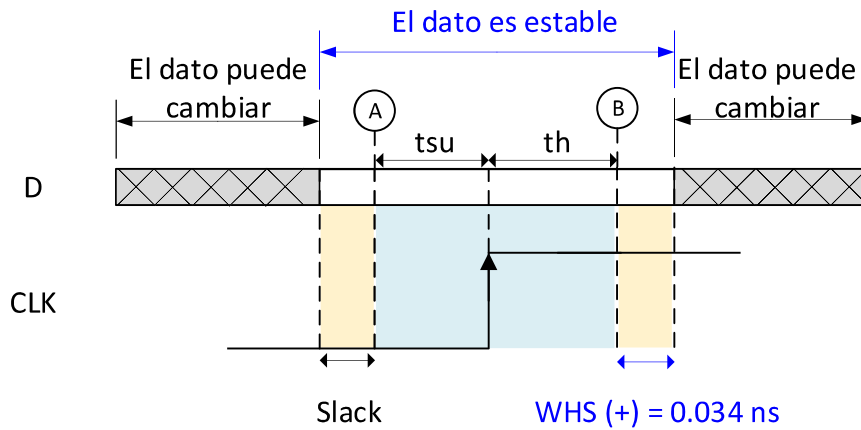


Figura 3.9. WHS del camino más rápido del sistema digital

Con los resultados obtenidos del reporte de implementación de tiempo, se garantiza que no se violan las restricciones de tiempo, debido a que TNS y THS son cero, es decir, no existe un solo camino que viole los tiempos t_{su} o t_h , garantizando así, que el flujo del algoritmo diseñado siga su curso natural sin presentar ningún desvío arbitrario.

3.1.2. REPORTE DE UTILIZACIÓN

El diseño final del sistema digital, descrito en VHDL para la PL, en conjunto con la integración del PS, bloque de reset y bloque de interconexiones AXI (Figura 2.18.), representan el porcentaje de utilización de los recursos de ZYBO, según detalle que se muestra en la Figura 3.10. Más del 90% de cada uno de los recursos (*resources*, Figura 3.11.) no son utilizados, por tanto, es notorio que para esta aplicación el diseñador del circuito digital puede obviar restricciones físicas y de tiempo para cumplir con metas de diseño, ya que tiene vastos recursos disponibles en toda la FPGA. Cuando un diseño se

vuelve muy grande y ocupa un porcentaje considerable de los recursos, el proceso de implementación se ve más limitado y se incrementa la complejidad para colocar recursos implementables físicamente en el reducido espacio disponible en la PL.

En la Figura 3.11. se presenta el reporte de utilización de manera explícita, aquí se indican exactamente cuántos elementos se han colocado y cuántos están disponibles en ZYBO. En los reportes se observa la utilización de los elementos principales de la FPGA, LUTs, FFs, IOs (entradas y salidas), BUFG (Buffers de reloj) y LUTRAMs. Las LUTRAMs son memorias de poca capacidad, también llamadas *distributed RAMs* y son formadas a base de LUTs [28].

Se puede observar en el porcentaje de utilización que existe mucho espacio disponible en ZYBO, facilitando así, la futura inclusión de nuevos procesos dentro del bloque de propiedad intelectual SPWM.

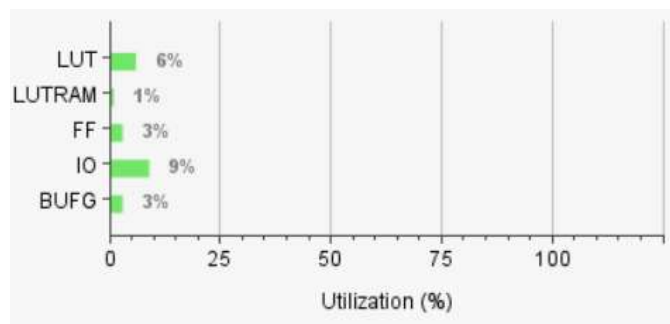


Figura 3.10. Reporte de utilización

Resource	Utilization	Available	Utilization %
LUT	1077	17600	6.12
LUTRAM	60	6000	1.00
FF	1010	35200	2.87
IO	9	100	9.00
BUFG	1	32	3.13

Figura 3.11. Reporte de utilización, disponibilidad y recursos

3.1.3. REPORTE DE POTENCIA

En la Figura 3.12. se presentan las estimaciones de las pérdidas estáticas (*Device Static*) y dinámicas (*Dynamic*) de ZYBO. La potencia total del SOC es de 1.705 W, comprendida

por 0.124 W para las pérdidas estáticas y 1.581 W para las dinámicas. La pérdida estática es producida por las corrientes de fuga de todos los transistores dentro del SOC; y, la pérdida dinámica se ocasiona por las cargas y descargas de los capacitores de entrada de los circuitos cuando existen cambios de estados lógicos en las salidas que alimentan a los mencionados circuitos, esta pérdida es función directa de la frecuencia [23].

Se nota que la potencia más representativa es la del sistema de procesamiento (PS7), esto se debe a que tiene una arquitectura definida que, aunque realice la acción más simple, utilizará el mismo número de recursos que una aplicación compleja.

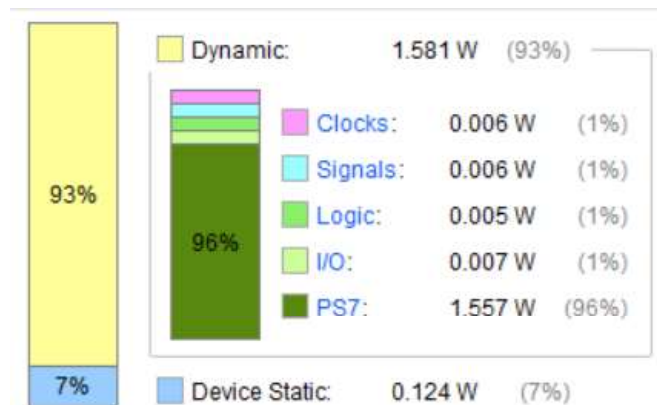


Figura 3.12. Reporte de potencia

3.2. SIMULACIÓN DE LAS SEÑALES DEL BLOQUE DE PROPIEDAD INTELECTUAL SPWM

Esta fase de simulación permite identificar como interactúan las señales dentro de la FPGA, es de interés observar estas señales ya que de esta manera se puede depurar y posteriormente verificar las salidas de ZYBO. Actualmente ZYBO trabaja con un rango de operación de frecuencia fundamental entre 5 Hz a 60 Hz e índice de modulación entre 10% a 500%.

Se simulará, en Vivado, bajo una condición específica de, frecuencia fundamental de 60 hz, índice de modulación (I_m) de 96% y relación de modulación de la frecuencia (m_f) de 87, para demostrar el correcto funcionamiento de las señales del bloque de propiedad intelectual SPWM. Es necesario notar que m_f es impar y múltiplo de 3 en cumplimiento de la recomendación analizada en la sección 1, dentro del título 1.3.4. MODULACIÓN POR ANCHO DE PULSO SENOIDAL (SPWM). Las condiciones específicas mencionadas construyen a las ondas modulantes y portadora, como se observa en la Figura 3.13., con una amplitud pico-pico de la portadora de 266 y de las modulantes de 255, es importante

mencionar que las ondas modulantes se encuentran centradas en la mitad de la amplitud pico-pico de la onda portadora. Los cursores en la figura son colocados en las transiciones de PWM R, de esta manera permiten medir el periodo de la onda modulante R y es visualizado en la regla de tiempo en la parte inferior de la figura; se observa que el periodo de la onda modulante es 16.667 ms y su frecuencia es el valor que ha sido ingresado al sistema, 60 Hz. Se aprecia que las señales SPWMs tienen una relación de trabajo variable para cada periodo, diferenciándose así de una PWM ordinaria; esta variación de la relación de trabajo en las SPWMs es fácilmente observable porque el índice de modulación está cercano a 100%, sin embargo, cuando se decrementa el I_m , la relación de trabajo de cada ciclo de las SPWMs gradualmente tiende a ser 50%. Se verifica, además, que cada par de SPWMs (S1-S4, S3-S6, S5-S2) son complementos y que el Led 0 se enciende de acuerdo con el comportamiento de la SPWM S1.

Cinco registros deben almacenar la información que envía el maestro al bloque IP, por tanto, para simular, se configuran valores específicos, en forma manual, a cada registro esclavo. La Tabla 3.2 muestra el valor configurado a los registros y también el tipo de amplitud o tiempo que representan.

Tabla 3.2. Valores configurados en la simulación del bloque IP SPWM

Registros esclavos del boque IP SPWM	Valor	Representa
<i>slv_reg0</i>	266	A_{pp} portadora
<i>slv_reg1</i>	1	pasos
<i>slv_reg2</i>	5787	T_m modulantes
<i>slv_reg3</i>	5	Offset
<i>slv_reg4</i>	45	T_m portadora

En la Figura 3.14. se realiza un acercamiento a la simulación presentada en la Figura 3.13., y se examina la frecuencia de la portadora colocando los cursores en los flancos negativos de PWM Portadora. Como m_f es 87, se espera tener una frecuencia de la portadora de 5.22 KHz. A partir del periodo de la portadora en la simulación se calcula la frecuencia y se verifica que se tiene el resultado esperado de 5.22 Khz. Además, se observa que las SPWMs se encuentran centradas con los picos máximo y mínimo de la portadora.

En la Figura 3.15. se realiza un acercamiento adicional a la simulación presentada en la Figura 3.14., para comprobar los tiempos muertos del par de SPWMs, S5 y S2. Se observa que se cumple con el tiempo muerto de 1.288 us, que fue configurado en la FPGA para evitar cortocircuitos en las transiciones de las SPWMs. La gráfica muestra el tiempo muerto de un par de SPWMs, sin embargo, este tiempo de 1.288 us está programado para todas las SPWMs.

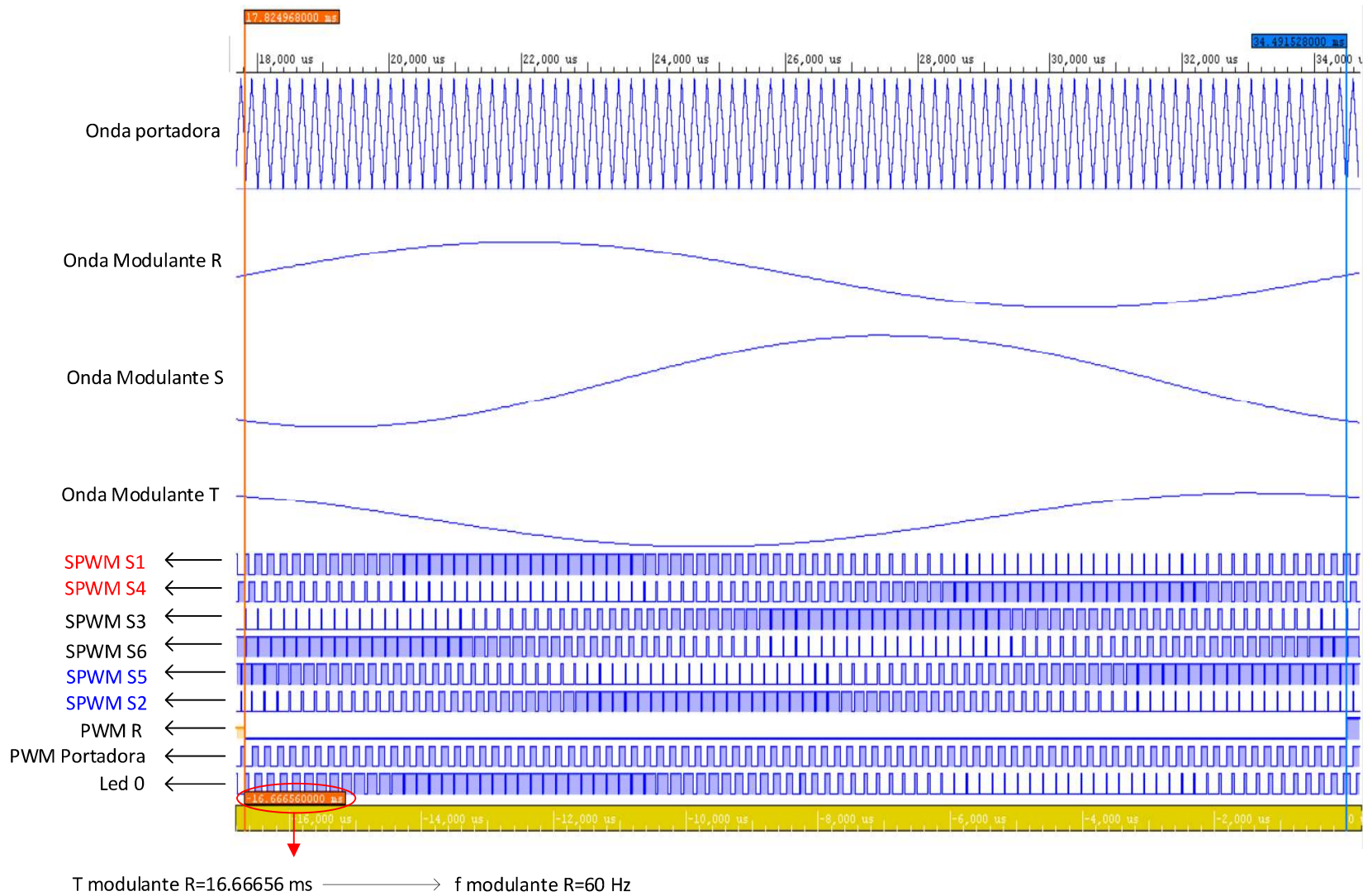


Figura 3.13. Simulación del comportamiento de las señales del bloque IP SPWM para 60 Hz e $I_m=96\%$

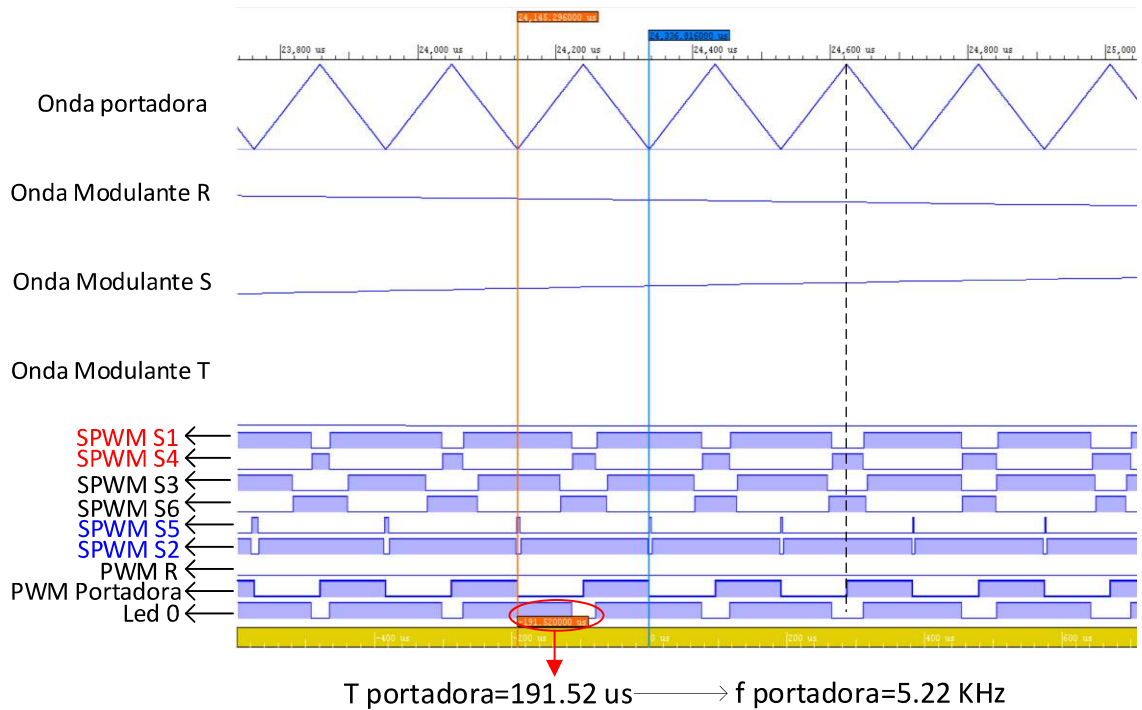


Figura 3.14. Frecuencia de portadora para punto de operación $f=60 \text{ Hz}$ e $I_m=96\%$

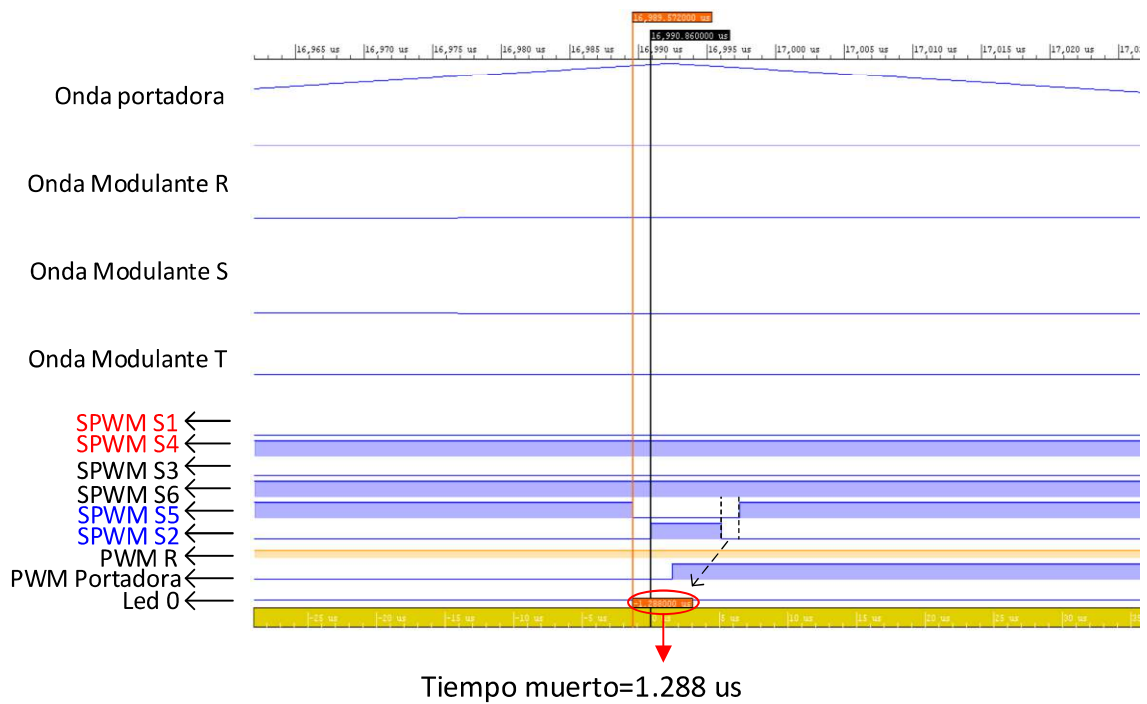


Figura 3.15. Tiempo muerto entre SPWMs S5 y S2

En la Figura 3.16. se muestra cómo se actualizan los registros de manera síncrona con el punto mínimo de la onda portadora, independientemente del instante del ingreso de parámetros por parte del usuario. Al inicio de la simulación se tiene una frecuencia

fundamental de 10 Hz, Im de 30% y mf de 84, luego son actualizados a la condición específica, frecuencia de 60 Hz, Im de 96% y mf de 87. Con la actualización de los registros se observa que las características de las SPWMs cambian al modificar las amplitudes y tiempos de muestreo de las ondas modulantes y portadora, especialmente la frecuencia de conmutación, que en la citada figura es de fácil visualización.

Las condiciones iniciales de la Figura 3.16. utilizaron un mf de 84 como un ejemplo para mostrar que el PS, maestro, puede ser modificado para cualquier mf, en este caso, par. Es importante mencionar que todas las pruebas que se presentan en esta sección utilizan un mf de 87.

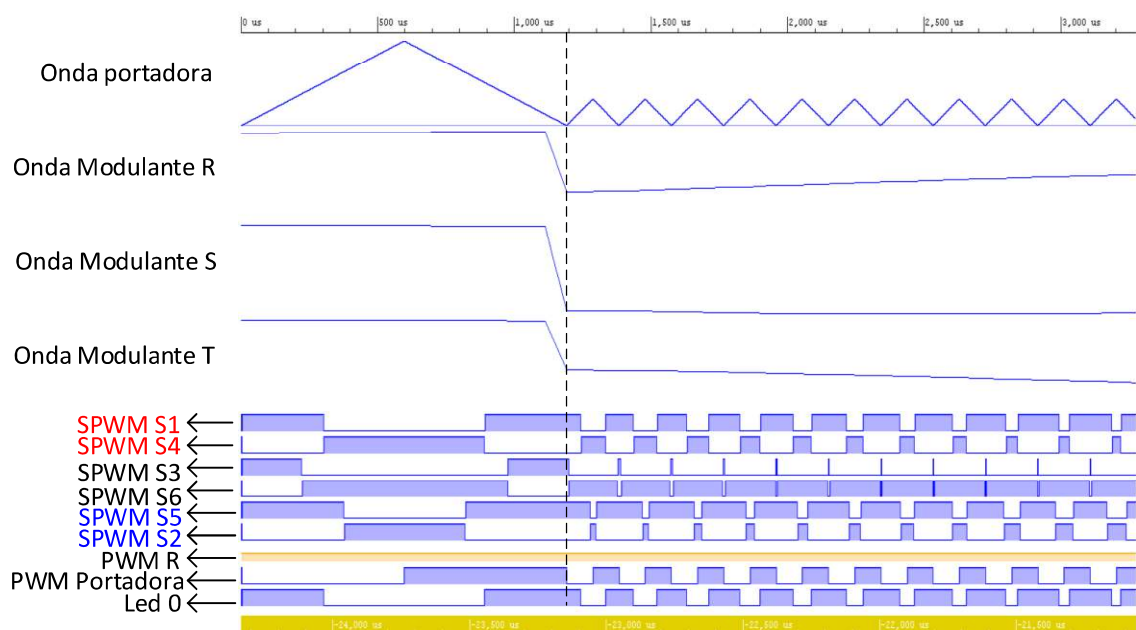


Figura 3.16. Actualización de registros en el punto mínimo de la onda portadora

Se incluye un par de puntos de operación adicionales con una frecuencia fundamental de 42 Hz e Im de 38% (Figura 3.17.); y con una frecuencia fundamental de 22 Hz y un índice de modulación de 10% (Figura 3.18.). En las gráficas se observan los puntos de operación comprobando que tanto la frecuencia como el índice de modulación son variables.

Se observa que se mantiene la relación de modulación de la frecuencia de 87 al realizar el producto con la frecuencia fundamental de los nuevos puntos de operación (42 Hz y 22 Hz) y obtener las frecuencias de las portadoras observadas en la Figura 3.19. y Figura 3.20., teniendo una desviación despreciable en la Figura 3.19.

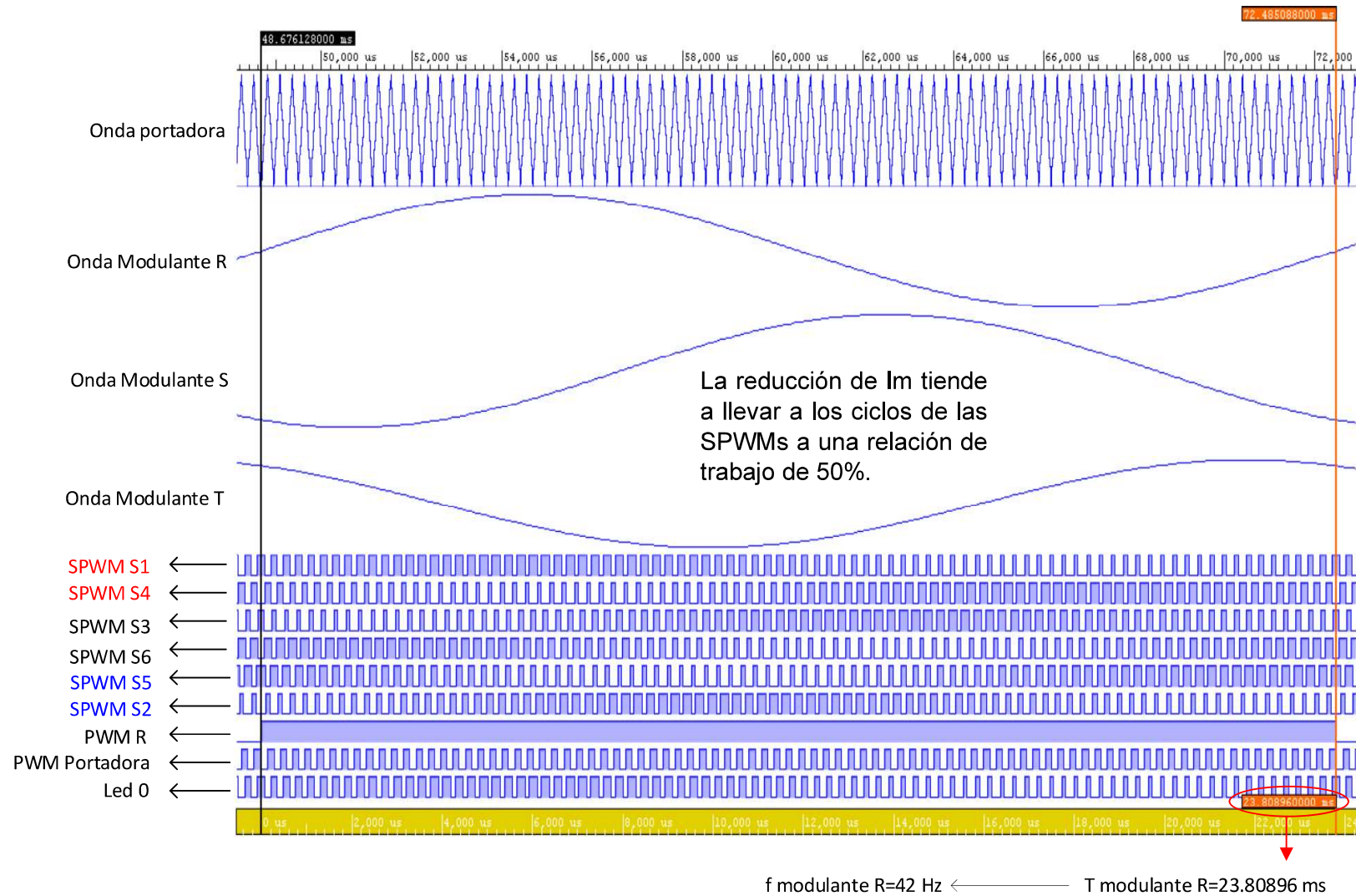


Figura 3.17. Simulación del comportamiento de las señales del bloque IP SPWM para 42 Hz e $I_m=38\%$

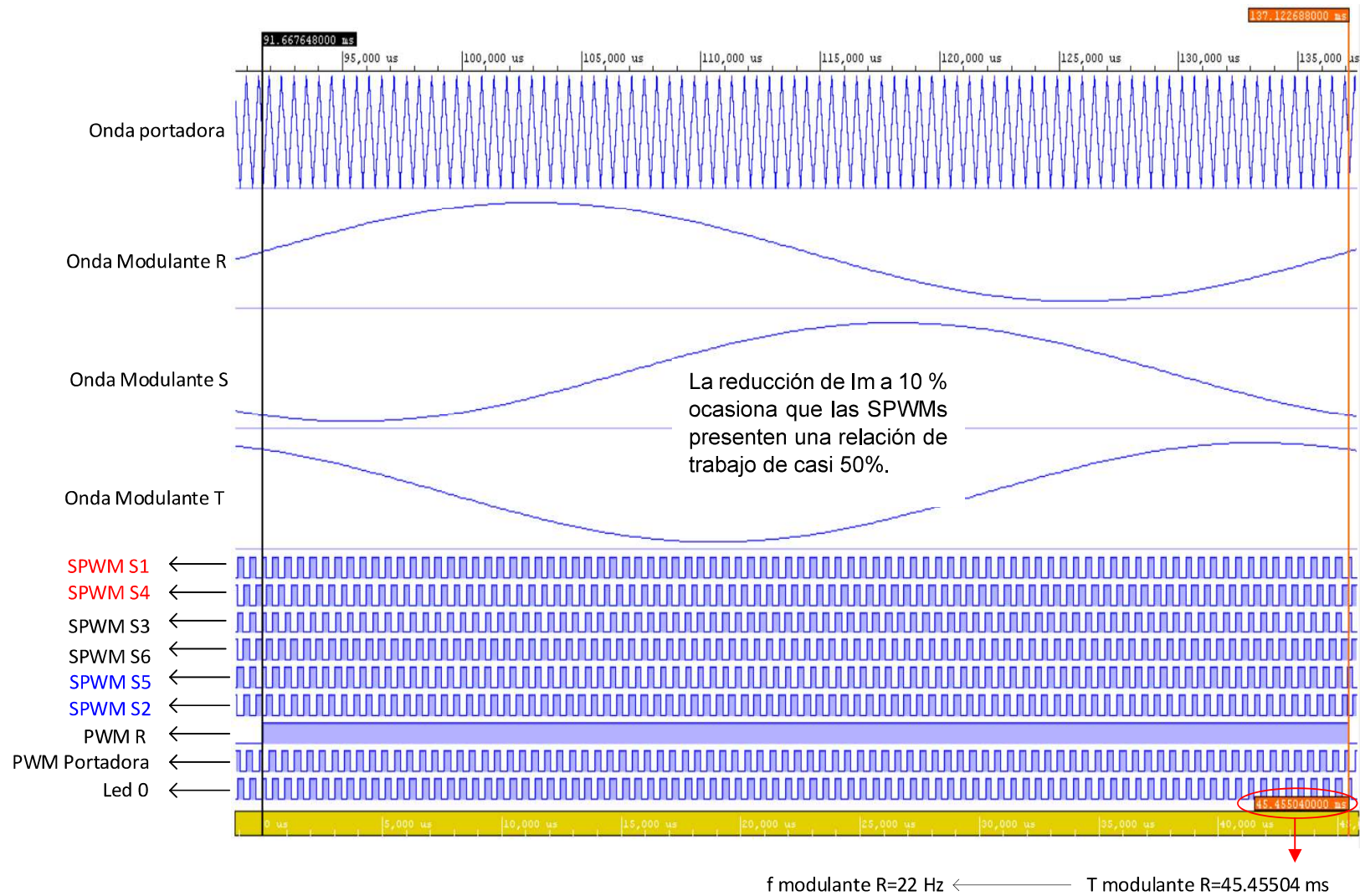


Figura 3.18. Simulación del comportamiento de las señales del bloque IP SPWM para 22 Hz e $I_m=10\%$

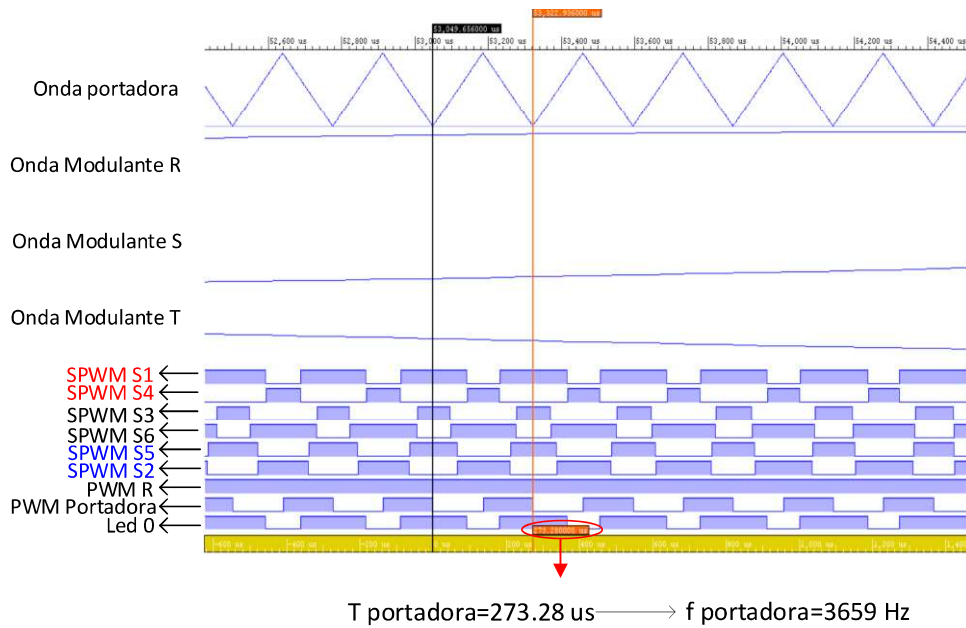


Figura 3.19. Frecuencia de portadora para punto de operación $f=42$ Hz e $I_m=38\%$

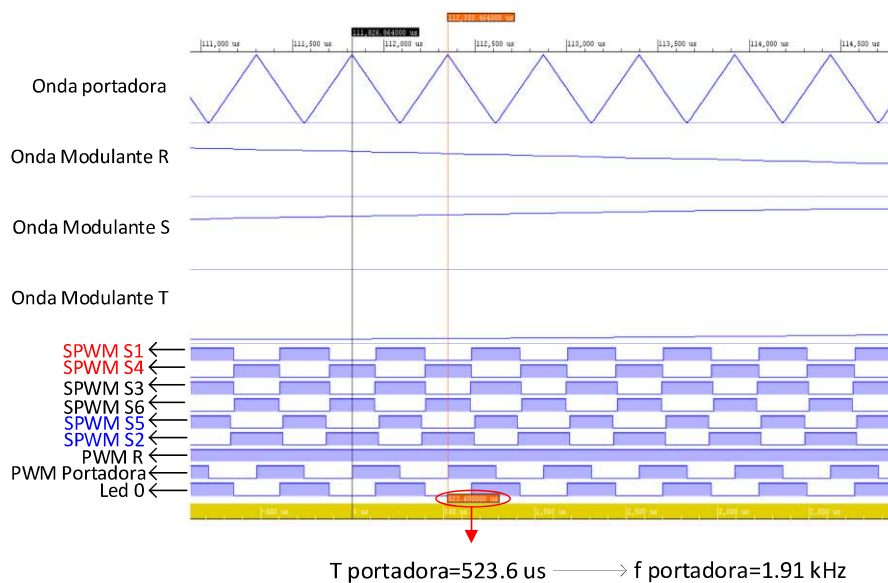


Figura 3.20. Frecuencia de portadora para punto de operación $f=22$ Hz e $I_m=10\%$

3.3. PRUEBAS DE LABORATORIO

Para validar el bloque de propiedad intelectual y el software del PS se procede a realizar pruebas de la modulación SPWM en las salidas de las señales del SOC ZYBO y en tres tipos de carga RLE (motor), RL y R, en distintos puntos de operación. Para las pruebas prácticas se utilizaron:

1. Un Sistema ZYBO-opto acoplamiento-inversor trifásico de IGBTs (Figura 3.21.).
2. Cargas para el inversor.

- a. Ventilador-motor de inducción trifásico Baldor, con número de catálogo 10142 en conexión Y (Figura 3.24.).
 - b. Circuito RL balanceado en conexión Y, aproximadamente 330 ohmios y 250 mH (Figura 3.25.).
 - c. Resistencias balanceadas en conexión Y, aproximadamente 330 ohmios (Figura 3.26.).
3. Multímetros Fluke 87 V para obtener mediciones del voltaje del bus dc (V_{bus}).
 4. Multímetros Fluke 87 V para obtener mediciones del voltaje RMS en las líneas de la carga del inversor.
 5. Un analizador de armónicos Fluke 41B.
 6. Un osciloscopio Tektronix TDS 2022C.
 7. Un osciloscopio HP 54645D.
 8. Un tacómetro Extech 461995.

En la Figura 3.27. se presenta la electrónica de potencia, carga e instrumentación enumerada en el párrafo anterior.



Figura 3.21. Sistema ZYBO-opto acoplamiento-inversor trifásico STEVAL-IHM023V3

En la Figura 3.22. y Figura 3.23. se muestra al sistema ZYBO-opto acoplamiento-inversor trifásico dentro de una estructura metálica que cumple con la función de escudo electromagnético.



Figura 3.22. Estructura metálica que contiene al sistema ZYBO-opto acoplamiento-inversor trifásico



Figura 3.23. Borneras de conexiones periféricas de la estructura metálica



Figura 3.24. Motor de inducción trifásico Baldor

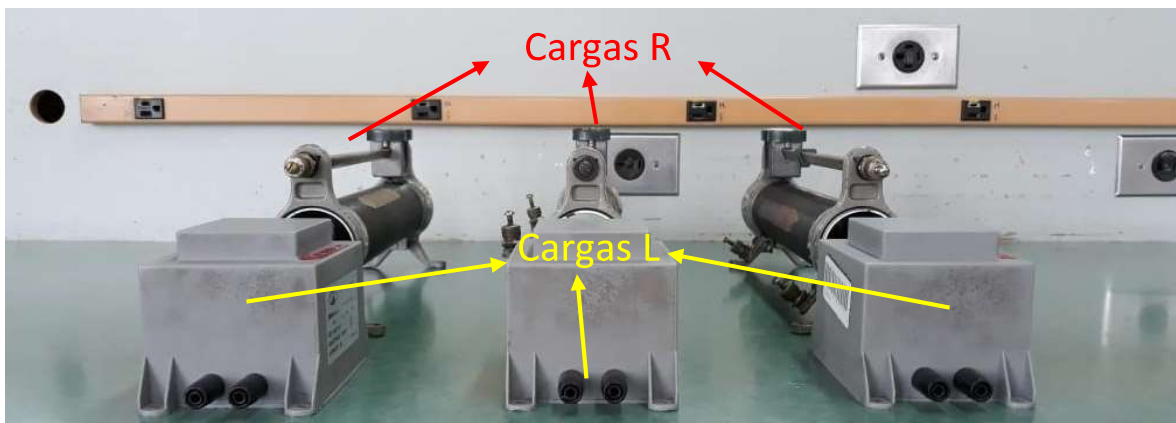


Figura 3.25. Cargas RL



Figura 3.26. Cargas R



Figura 3.27. Electrónica de potencia, carga e instrumentación empleada en la validación de la modulación SPWM configurada en ZYBO

3.3.1. PRUEBAS DIRECTAMENTE EN LAS SALIDAS DE ZYBO

La primera validación se la realiza directamente en el SOC ZYBO. Se utiliza un osciloscopio HP 54645D para mostrar todas las señales digitales de salida del puerto Pmod estándar en una misma pantalla. En la Tabla 3.3. se presentan las equivalencias entre las señales del puerto JE y el número de la señal que se observa en el osciloscopio, según consta en la Figura 3.28. y Figura 3.29.

Tabla 3.3. Orden de las señales observadas en el osciloscopio HP 54645D

Nombre de señal del puerto Pmod estándar JE	Pin de puerto Pmod estándar JE	Señal de osciloscopio HP 54645D
SPWM S1	Pin 1	0
SPWM S4	Pin 2	1
SPWM S3	Pin 3	2
SPWM S6	Pin 4	3
SPWM S5	Pin 7	4
SPWM S2	Pin 8	5
PWM R	Pin 9	6
PWM Portadora	Pin 10	7

Tanto la Figura 3.28. como la Figura 3.29. presentan tres fotografías de la pantalla del osciloscopio HP 54645D: a, b y c. En la subdivisión 'a' se permite visualizar el periodo de una onda modulante entre cada transición de la señal 6 del osciloscopio (PWM R); en 'b' se reduce la escala de tiempo para poder observar por lo menos 5 ciclos de las ondas SPWM; y, en 'c' se reduce aún más la escala de tiempo para medir la frecuencia de la señal 7 que representa a la onda portadora del bloque de propiedad intelectual en la FPGA. En las subdivisiones 'a' y 'c' se utilizan los cursores para medir las frecuencias que son indicadas en la parte inferior derecha de la pantalla del osciloscopio, cuyo valor es presentado como el inverso de la diferencia de tiempo entre los cursores ($1/\Delta t$).

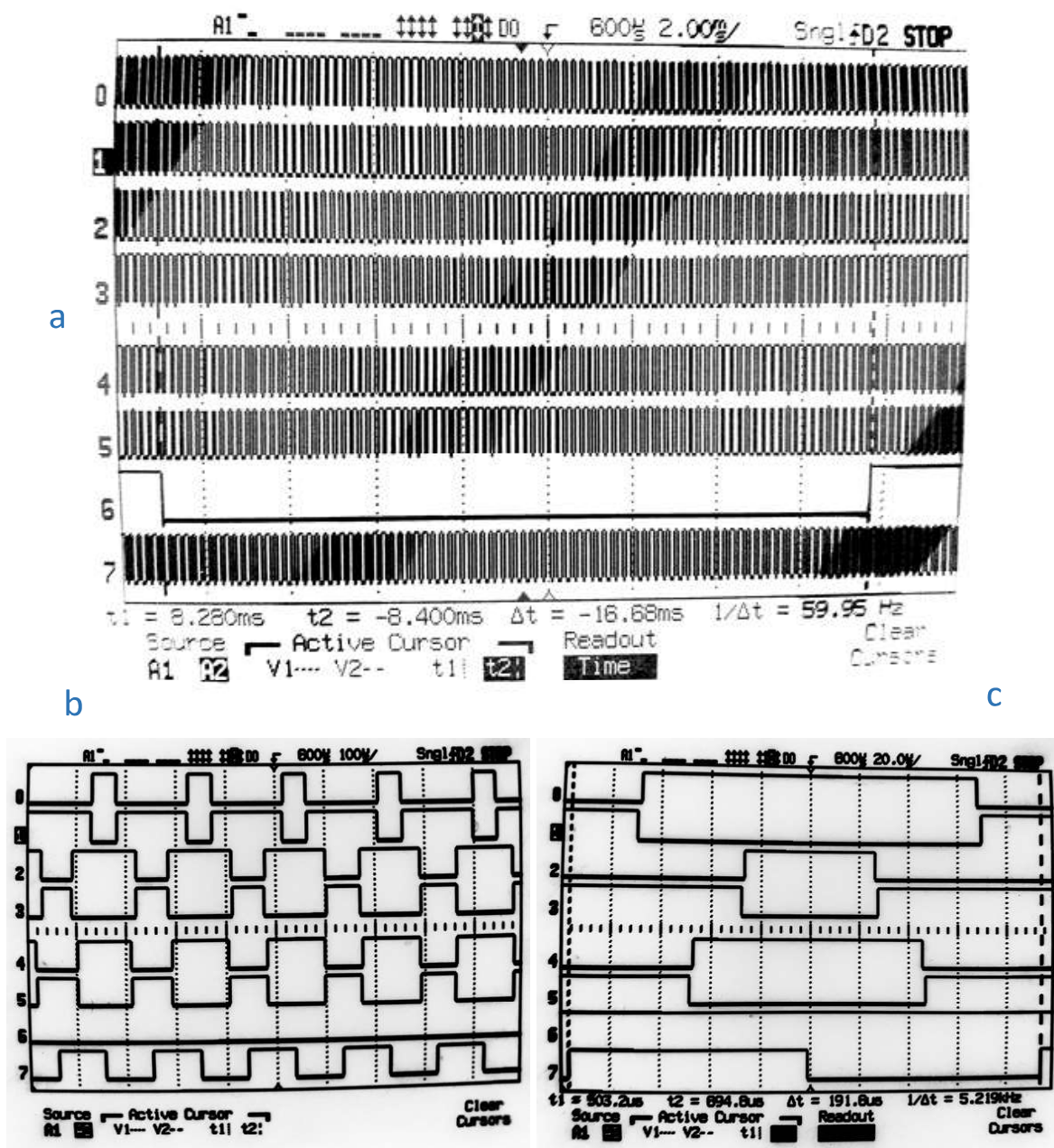


Figura 3.28. Señales de salida del puerto Pmod JE de ZYBO para $f=60\text{ Hz}$ e $I_m=50\%$

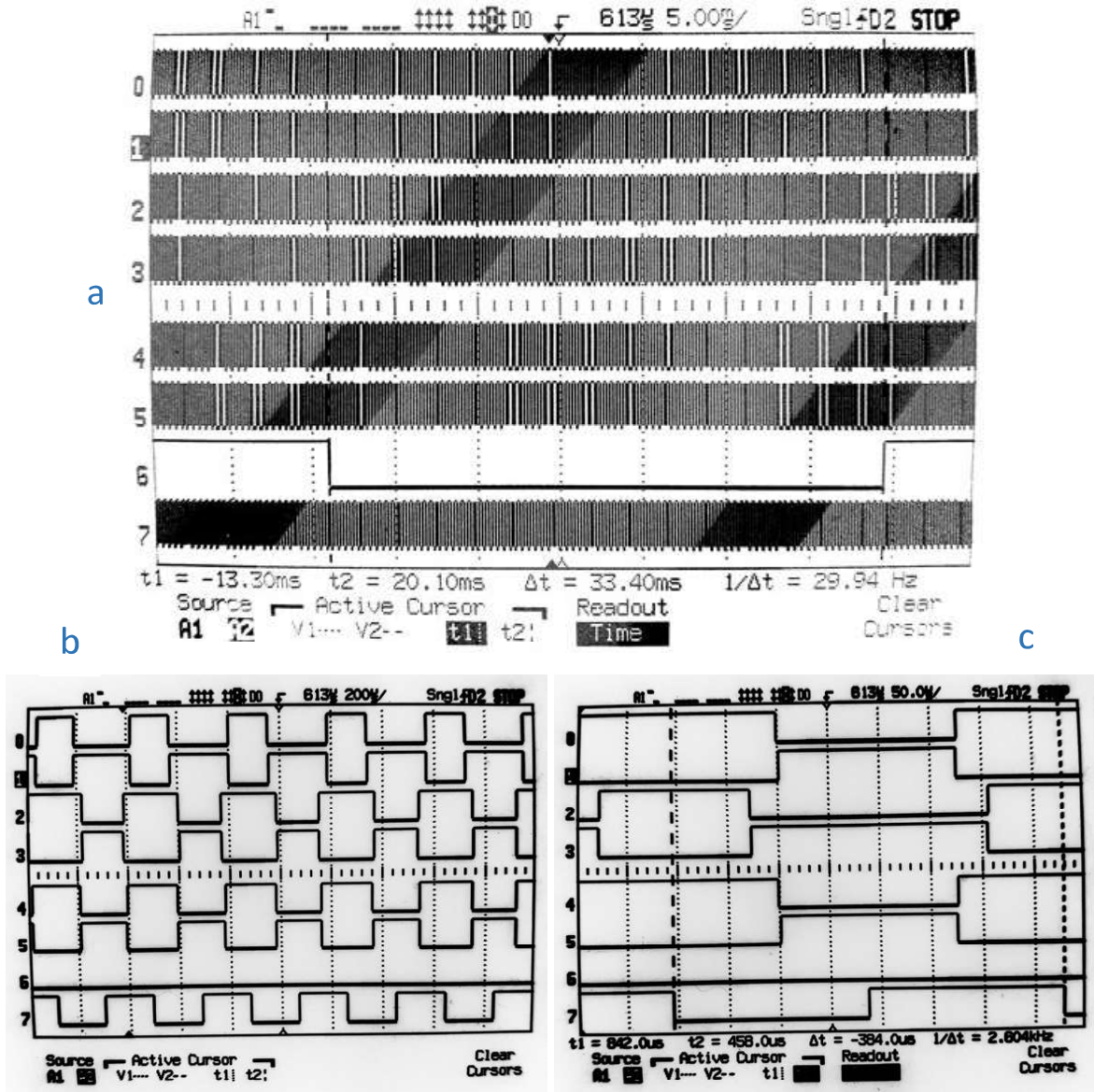


Figura 3.29. Señales de salida del puerto Pmod JE de ZYBO para $f=30$ Hz e $I_m=20\%$

En la Figura 3.28.b. se ha configurado un I_m de 50% y en la Figura 3.29.b. se lo ha reducido a 20%, se observa que los periodos de las SPWMs, en este cambio, se aproximan a una relación de trabajo del 50%, de esta manera se confirma experimentalmente que el I_m afecta el comportamiento de las SPWMs. En las mediciones de frecuencia de las subdivisiones 'a' y 'c' de la Figura 3.28. y Figura 3.29. se aprecia que las frecuencias son prácticamente 60 y 30 Hz para la subdivisión 'a'; y 5220 y 2610 para la subdivisión 'b', respectivamente, verificando que la relación de modulación de frecuencia es 87.

3.3.2. PRUEBAS CON MOTOR DE INDUCCIÓN

Para realizar las pruebas de la modulación SPWM con cargas se analiza la salida del inversor, es decir, el voltaje entre líneas, de este voltaje se puede extraer toda la

información necesaria para validar el funcionamiento de la aplicación programada en ZYBO, esto es, frecuencia de la fundamental de voltaje, formas de onda, voltaje RMS entre líneas, análisis de la transformada rápida de Fourier (FFT-*Fast Fourier transform*).

En la Figura 3.30. se muestra la captura del osciloscopio Tektronix TDS 2022C, donde se observa el comportamiento del voltaje entre las líneas a y b con una frecuencia fundamental de la carga de 60 Hz e índice de modulación del 96%, se destaca que se obtienen aproximadamente los tres niveles esperados en las formas de onda de voltaje: V_{bus} , 0, $-V_{bus}$. La fundamental del voltaje de la figura es 60 Hz, medido con el analizador de armónicos Fluke 41B y el voltaje es aproximadamente 220 Vrms, medido con el multímetro Fluke 87 V. El voltaje medido en el bus dc es 306.5 Vdc y la frecuencia de la portadora es 5.22 KHz.

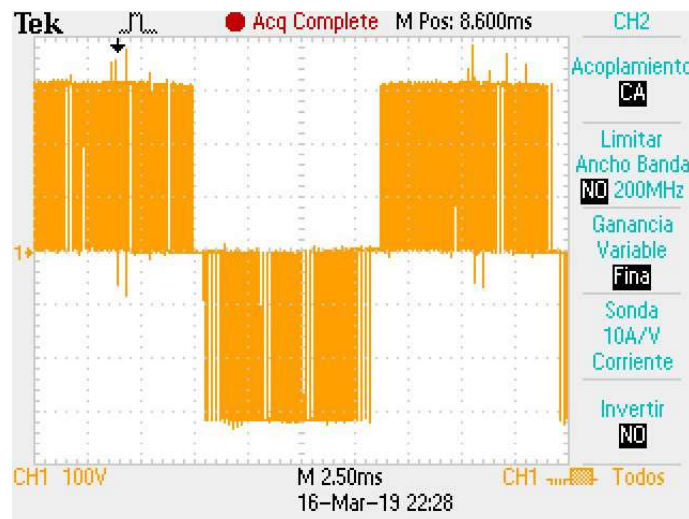


Figura 3.30. Voltaje V_{ab} con $f=60\text{Hz}$ e $I_m=96\%$ para motor de inducción

En la Figura 3.31. se presenta un acercamiento del voltaje entre líneas reproducido por el osciloscopio en la Figura 3.30., este permite divisar de mejor manera las formas de ondas cuadradas, frecuencia de conmutación, la variación de la relación de trabajo en cada ciclo que es una característica de una modulación SPWM.

En la Figura 3.32. se muestra otro punto de operación del motor de inducción, frecuencia fundamental de 20 Hz e índice de modulación de 50%. Se observa al voltaje entre líneas (color naranja) y a la corriente de la fase c (color verde claro). La corriente adquiere una forma sinusoidal debido a las inductancias de los bobinados del estator del motor de inducción; y, con relación al voltaje entre líneas, se mantienen los mismos niveles y similares formas de onda comparándolo con la Figura 3.30.

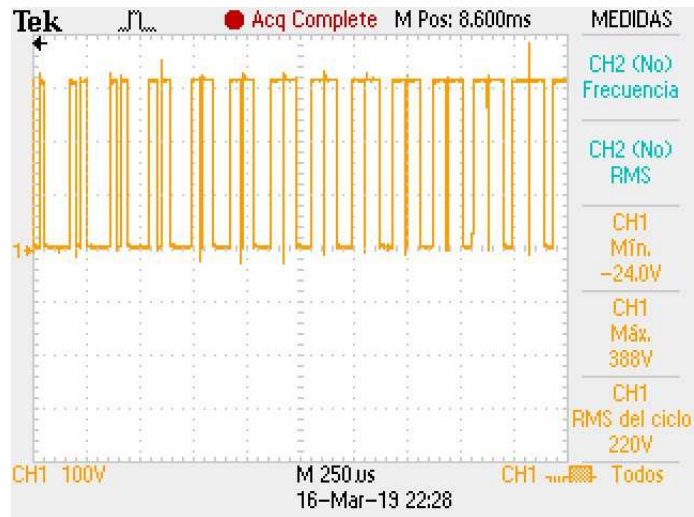


Figura 3.31. Acercamiento de Voltaje V_{ab} con $f=60\text{Hz}$ e $I_m=96\%$ para motor de inducción

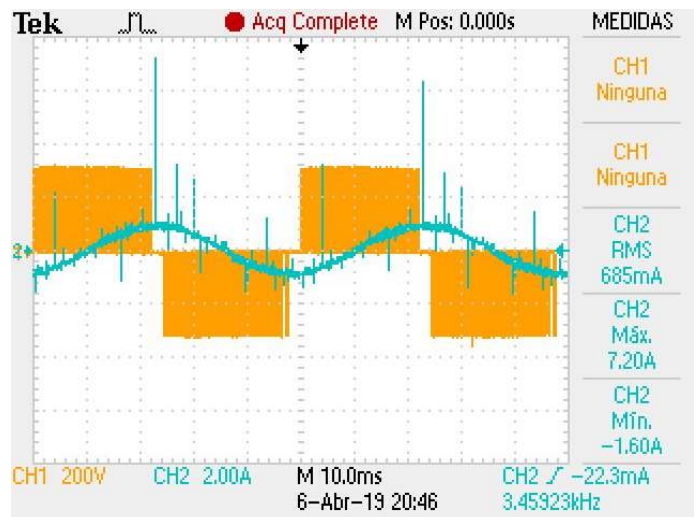


Figura 3.32. Voltaje V_{ca} con $f=20\text{Hz}$, $I_m=50\%$ y corriente de fase c para motor de inducción

En las gráficas de voltaje entre líneas de las capturas del osciloscopio (Figura 3.30., Figura 3.31.), se espera obtener un voltaje máximo y mínimo de V_{bus} y $-V_{bus}$, respectivamente; sin embargo, se debe tomar en consideración la caída de voltaje en los IGBTs, lo que ocasiona que estos niveles de voltaje sean menores. El inversor trifásico utiliza los IGBTs STGP10H60DF, en su hoja de datos [29] se observan 3 valores típicos del voltaje de saturación colector-emisor (V_{sat}), 1.5, 1.65 y 1.7 V, que están en función de las temperaturas de juntura, 25, 125, 175 °C, se decide utilizar el promedio de estos 3 datos de voltaje para abarcar el rango de temperatura, obteniéndose un valor de 1.6 V.

En la Figura 3.33. se indica la activación de los IGBTs de un solo estado del inversor para modulación SPWM, es importante notar que se mide el voltaje RMS entre las líneas a y b,

por tanto, el análisis solo se realiza en esos dos ramales. El voltaje máximo que se obtiene entre a y b está definido en la Ecuación 3.1. como V_{bus}' .

$$V_{bus}' = V_{bus} - 2V_{sat} \quad (3.1)$$

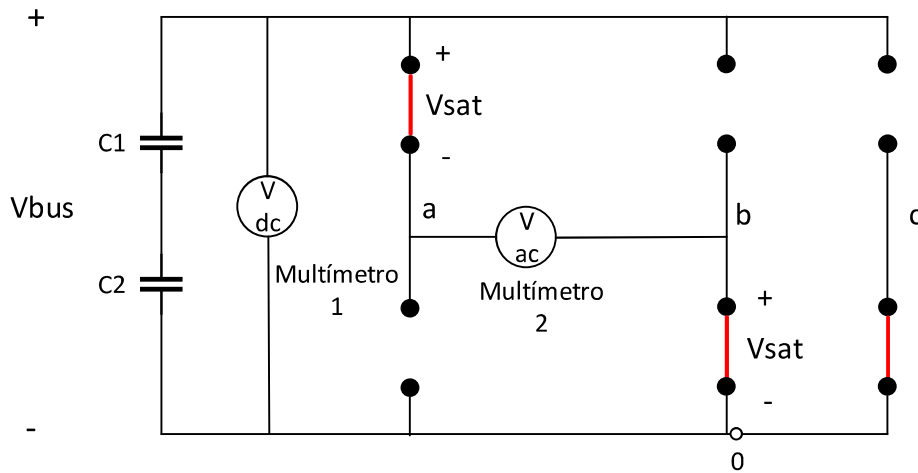


Figura 3.33. Caídas de voltaje en los interruptores de potencia del inversor trifásico

En la Figura 3.34. y Figura 3.35. se observa el resultado de la transformada rápida de Fourier (FFT-Fast Fourier transform) del voltaje entre líneas del inversor trifásico (V_{ca}) para el orden de armónicos y frecuencia, respectivamente, calculado en Matlab por medio de los puntos de V_{ca} obtenidos del osciloscopio. El primer pico, desde la izquierda, representa a la frecuencia fundamental de 60 Hz y desplazándose hacia la derecha se observan a los armónicos más representativos. En las citadas figuras se indican los valores de los picos más relevantes por medio de etiquetas en donde Y representa el eje de la magnitud en porcentaje con respecto a la fundamental y X el orden del armónico o frecuencia. Se observa, además, que el orden de los armónicos del análisis FFT concuerda teóricamente con la Ecuación 1.5. Al calcular los armónicos alrededor de la relación de modulación de frecuencia, utilizando la Ecuación 1.5., se obtienen los resultados de los armónicos más representativos presentados en la Ecuación 3.2. hasta la Ecuación 3.5., que son los mismos resultados que se muestran en la Figura 3.34. Para convertir de orden de armónicos a frecuencia, se debe multiplicar el orden del armónico por la frecuencia fundamental.

$$h_{a1} = j(mf) \pm k = 1(87) - 2 = 85 \quad (3.2)$$

$$h_{a2} = j(mf) \pm k = 1(87) + 2 = 89 \quad (3.3)$$

$$h_{b1} = j(mf) \pm k = 2(87) - 1 = 173 \quad (3.4)$$

$$h_{b2} = j(mf) \pm k = 2(87) + 1 = 175 \quad (3.5)$$

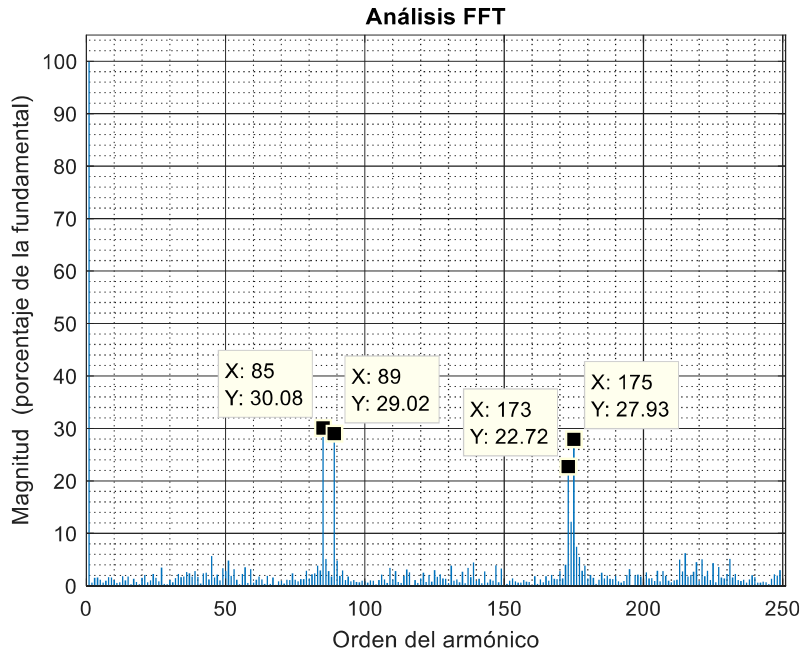


Figura 3.34. Análisis FFT de orden de armónicos de Vca para $f=60$ e $I_m=92\%$ para motor de inducción

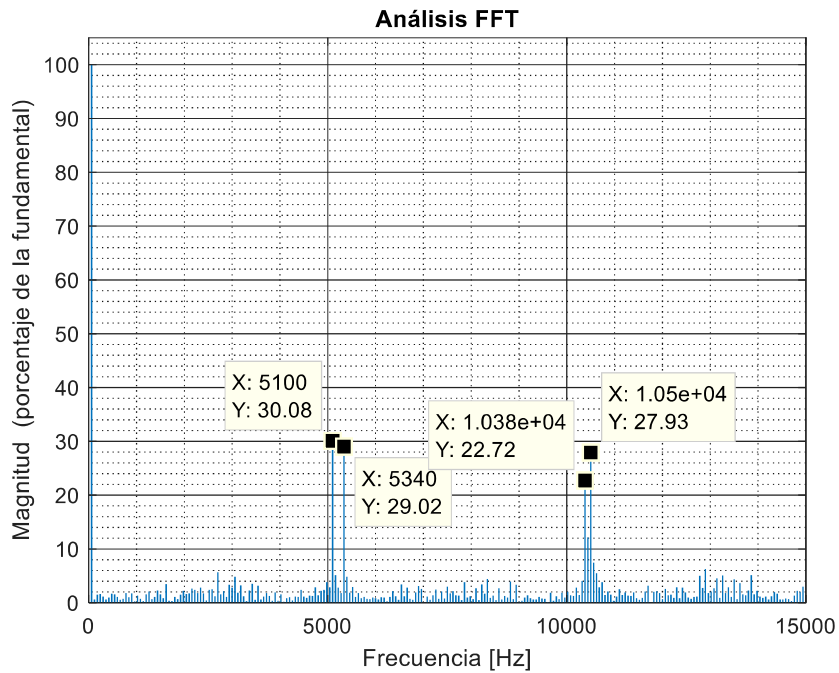


Figura 3.35. Análisis FFT de frecuencias de Vca para $f=60$ e $I_m=92\%$ para motor de inducción

Para validar el voltaje de salida RMS entre líneas del inversor, se conectan distintos tipos de carga y se utiliza el software de simulación PSIM, debido a que esta herramienta está especialmente diseñada para simular aplicaciones de electrónica de potencia, drivers para motores y proporciona resultados acertados. Para simular se usa V_{bus} para alimentar al inversor en vacío y con el tiempo muerto de 1.288 us configurado en ZYBO, así se logra replicar las condiciones del verdadero voltaje máximo sin considerar la influencia que pueda generar ciertos tipos de carga en las salidas de voltaje del inversor, de esta manera se obtiene un acertado valor teórico del voltaje entre líneas.

En la Tabla 3.4. se presentan los datos experimentales para validar la aplicación en ZYBO, utilizando el motor de inducción como carga del inversor, en donde:

- f = frecuencia fundamental (Entrada de ZYBO ingresada por el usuario).
- $I_m\%$ = Índice de modulación en porcentaje (Entrada de ZYBO ingresada por el usuario).
- V_{bus} = Voltaje del bus dc, medido con un multímetro Fluke 87 V.
- V_{ca} RMS = Voltaje RMS entre las líneas c y a del motor de inducción, medido con un multímetro Fluke 87 V.
- V_{ca} RMS simulación = Voltaje RMS entre las líneas del inversor en simulación, utilizando el software PSIM.
- $Er\% V_{ca}$ = Error relativo porcentual, considerando V_{ca} RMS como el valor medido y V_{ca} RMS simulación como el valor teórico. Su cálculo se muestra en la Ecuación 3.6.
- n = Velocidad del motor de inducción, medida con el tacómetro Extech 461995.
- V_1 RMS = Voltaje de la fundamental entre las líneas 'ca', medido con el analizador de armónicos Fluke 41B.
- THD V_{ca} = Distorsión armónica total de V_{ca} calculado en Matlab con los puntos de voltaje V_{ca} , obtenidos del osciloscopio Tektronix TDS 2022C.

$$Er\% = \frac{Valor\ medido - Valor\ teórico}{Valor\ teórico} 100 \quad (3.6)$$

Tabla 3.4. Datos de las pruebas con motor de inducción

f [Hz]	Im%	Vbus [v]	Vca rms [v]	Vca rms simulación [v]	Er% Vca	n [rpm]	V1 rms [v]	THD Vca
22	10	315.2	71.9	71.63	0.38	1080	17.8	3.91
24	12	315.2	79.6	78.68	1.17	1242	21.8	3.55
26	15	314.1	88.5	87.84	0.75	1409	27.3	2.95
28	17	314.7	94.2	93.78	0.45	1516	31	2.84
30	19	313.3	100.1	98.76	1.36	1639	35.1	2.61
32	22	314.4	108	106.79	1.13	1783	41.2	2.47
34	25	314	114.4	113.8	0.53	1902	46.1	2.25
36	28	314.1	121.7	120.56	0.95	2029	51.9	2.08
38	31	314	127.4	126.89	0.40	2154	57.4	1.97
40	34	313.9	133.6	132.92	0.51	2266	63.1	1.84
42	38	314	141.9	140.67	0.87	2400	70.9	1.72
44	42	313	148.8	147.5	0.88	2500	78.3	1.6
46	45	313.2	154	152.81	0.78	2602	83.9	1.51
48	49	313.5	160.9	159.68	0.76	2723	91.5	1.44
50	54	312.9	169	167.4	0.96	2831	101.2	1.36
52	58	313.5	174.8	173.88	0.53	3010	108.5	1.19
54	63	312.9	182	180.94	0.59	3124	117.8	1.15
56	67	312.2	187.9	186.22	0.90	3216	124.9	1.12
58	72	312.1	194.3	193.04	0.65	3324	134.9	1.03
60	77	311.9	201.5	199.55	0.98	3439	144	0.96
60	92	312.8	220.4	218.65	0.80	3453	173	0.78

Para validar el sistema ZYBO-inversor-motor de inducción, se verifica la frecuencia fundamental de la salida de voltaje entre líneas del inversor y su voltaje RMS. Para los datos analizados en la Tabla 3.4., Tabla 3.5. y Tabla 3.6., la frecuencia fundamental ingresada por el usuario al sistema en chip ZYBO es la misma que se observa en las mediciones de frecuencia fundamental del analizador de armónicos Fluke 41B, por lo tanto, no hay necesidad de calcular errores para este parámetro. El voltaje RMS entre líneas presenta pequeñas desviaciones del valor teórico, por este motivo, se presentan los errores relativos calculados para cada punto de operación. Para la Tabla 3.4. se obtiene un promedio de error relativo de 0.78%.

Los errores del voltaje RMS en promedio se encuentran debajo del 1%, lo que puede considerarse insignificante, sin embargo, a continuación se explica la existencia de estos errores:

- Errores en las mediciones: Las mediciones del voltaje del bus dc y el voltaje entre líneas se las realizan con los multímetros digitales Fluke 87 V. Existen pequeñas fluctuaciones en las pantallas de los multímetros, especialmente del que mide el

voltaje del bus dc, por esta razón no se puede divisar un número concreto y se decide tomar el valor más frecuente mostrado.

- Rango de exactitud del elemento de medida: Las especificaciones del fabricante de los multímetros Fluke 87 V presentan un rango de exactitud para mediciones de voltajes AC de $\pm [0.7\%(\text{valor medido}) + 0.2]$ [30], [31]. Se entiende entonces que la medida verdadera puede estar presente dentro de este rango.

Para los valores mostrados en la Tabla 3.4. se determina que 11 de los 21 datos del valor teórico del voltaje en las líneas RMS se encuentran dentro del rango de exactitud del multímetro que mide Vca RMS, lo que indica que existe una posibilidad de que 11 de los valores verdaderos coincidan con los teóricos; el resto de los datos que no se encuentran dentro del rango, están alejados del mismo por pocas décimas de voltaje. Este análisis ayuda a comprender la existencia de los errores relativos de la Tabla 3.4. y pone en evidencia que los elementos de medición, aunque tienen un rango de exactitud aceptable y suficiente, no entregan una medición perfecta, pudiendo ocasionar pequeños errores en las mediciones.

La salida del motor, velocidad, se incrementa a medida que se aumenta la frecuencia de la fundamental, comprobando así que la velocidad varía en función de la frecuencia. En las dos últimas filas de la Tabla 3.4. se presentan puntos de operación en donde la frecuencia se mantiene constante y aun así se observa un incremento en la velocidad del motor, debido a que el voltaje RMS de alimentación aumenta con el incremento del índice de modulación.

Se observa que el voltaje de la fundamental entre líneas aumenta proporcionalmente con el índice de modulación, al igual que el voltaje RMS total; y, además, el THD decremента de manera inversamente proporcional con I_m , esto se debe a que a medida que se incrementa I_m , el valor de la fundamental de voltaje adquiere un valor más significativo. En la Ecuación 3.7. se puede observar la proporcionalidad mencionada, donde:

- V_{rms} = Voltaje rms total de la onda periódica analizada.
- V_{dc} = Componente DC de voltaje de la onda periódica analizada.
- V_1 = Fundamental de voltaje de la onda periódica analizada.

$$THD = \frac{\sqrt{V_{rms}^2 - V_{dc}^2 - V_1^2}}{V_1} \quad (3.7)$$

El mismo formato de la Tabla 3.4. se utiliza para la validación con cargas R y RL, con la diferencia que se miden los voltajes en las líneas ab y las cargas no tienen salida de velocidad.

3.3.3. PRUEBAS CON CARGA TRIFÁSICA BALANCEADA RESISTIVA-INDUCTIVA EN CONEXIÓN Y

Para realizar las pruebas con cargas RL (~330 ohmios, 250 mH), se procede a conectar las cargas trifásicas en conexión 'Y' y alimentarlas con distintos puntos de operación. En la Tabla 3.5. se muestran los datos obtenidos en laboratorio y mediante análisis de los valores de error relativo ($Er\% V_{ab}$), se determina que el promedio del error es 0.68%, representando valores despreciables; y, además, 15 de los 21 datos del voltaje RMS entre líneas teórico se encuentran dentro del rango de exactitud del multímetro que mide V_{ab} RMS y los 6 datos teóricos restantes se hallan alejados por pocas décimas de voltios. De este análisis se concluye que el 71% de las mediciones de voltaje entre líneas tienen la probabilidad de ubicarse en el mismo valor teórico.

En la Figura 3.36. y Figura 3.37. se muestran las formas de onda de voltaje (V_{ab}) y corriente (i_c), respectivamente, es evidente que las características de las señales son idénticas a los resultados obtenidos con el motor de inducción, esto se debe a que las salidas de voltaje del inversor no deberían verse comprometidas con la carga conectada y las corrientes son sinusoidales por la presencia de inductores.

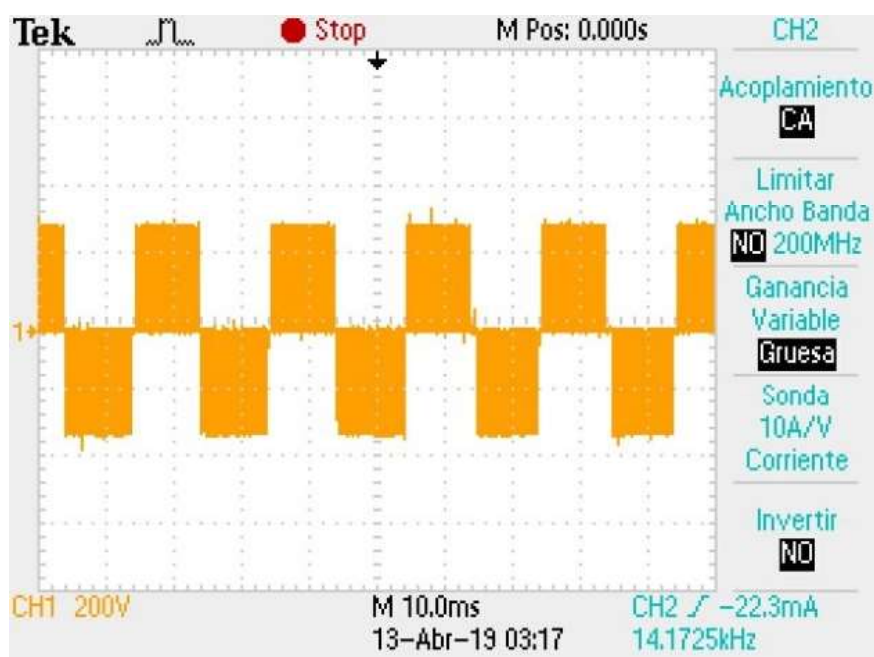


Figura 3.36. Voltaje V_{ab} con $f=50$ Hz e $I_m=83\%$ para carga RL trifásica

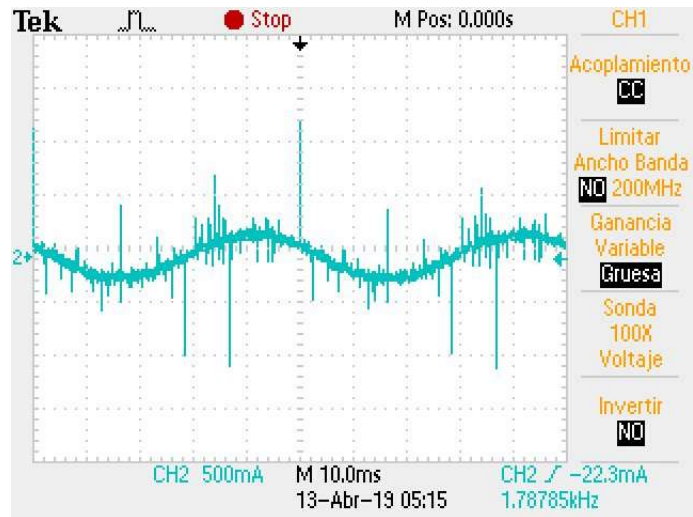


Figura 3.37. Corriente de fase c con $f=20$ e $I_m=50\%$ para carga RL trifásica

El THD en estas pruebas no supera 1 debido a que se utilizan índices de modulación a partir de 82%, suficientemente grandes para generar un voltaje fundamental significativo. Finalmente, la frecuencia de la fundamental ingresada en ZYBO, que se presenta en la primera columna de la Tabla 3.4., Tabla 3.5. y Tabla 3.6., es la misma obtenida en las mediciones, por tanto, no es necesario calcular errores de frecuencia.

Tabla 3.5. Datos de las pruebas con carga RL trifásica

f [Hz]	$I_m\%$	V_{bus} [v]	V_{ab} rms [v]	V_{ab} rms simulación [v]	$Er\%$ V_{ab}	V_1 rms [v]	THD V_{ab}
50	82	302.1	201.8	199.73	1.04	149	0.93
50	83	302	202.7	200.9	0.90	151	0.88
50	84	301.5	203.3	201.78	0.75	152	0.87
50	85	302.4	204.5	203.6	0.44	154	0.85
50	86	302	206	204.54	0.71	156	0.84
50	87	301.9	207.4	205.67	0.84	158	0.84
50	88	302.1	208.4	207	0.68	160	0.84
50	89	302	209.3	208.12	0.57	161	0.84
50	90	301.6	211	209.02	0.95	163	0.82
50	91	302.1	212	210.54	0.69	165	0.78
60	82	302.5	200.4	199.76	0.32	149	0.92
60	83	302.6	202.7	201.05	0.82	150	0.88
60	84	302.4	202.9	202.14	0.38	151	0.86
60	85	301.9	204	203.02	0.48	153	0.89
60	86	301.4	205.5	203.89	0.79	155	0.84
60	87	302.1	206.4	205.57	0.40	158	0.88
60	88	302.1	208.2	206.76	0.70	160	0.82
60	89	301.6	208.5	207.6	0.43	161	0.83
60	90	301.6	210.6	208.78	0.87	163	0.82
60	91	301.6	211.6	209.95	0.79	165	0.79
60	92	301.7	212.4	210.8	0.76	167	0.78

3.3.4. PRUEBAS CON CARGA TRIFÁSICA BALANCEADA RESISTIVA EN CONEXIÓN Y

Por último, se analizan los datos obtenidos a partir de cargas resistivas (~ 330 ohmios) en conexión 'Y', que se presentan en la Tabla 3.6. De la misma manera que en las pruebas con motor de inducción y cargas RL, la frecuencia de la fundamental no tiene errores y se analizan únicamente las desviaciones del voltaje RMS entre líneas. El promedio de error relativo porcentual es 0.32% y el multímetro que mide V_{ab} RMS sitúa a todos los valores teóricos dentro de su rango de exactitud. De nuevo, los errores encontrados son despreciables.

En la Figura 3.38. se observa el voltaje entre líneas en las cargas resistivas para un punto de operación, se analiza que el patrón en las formas de onda es constante para las salidas del inversor, independientemente de las cargas que se conecten.

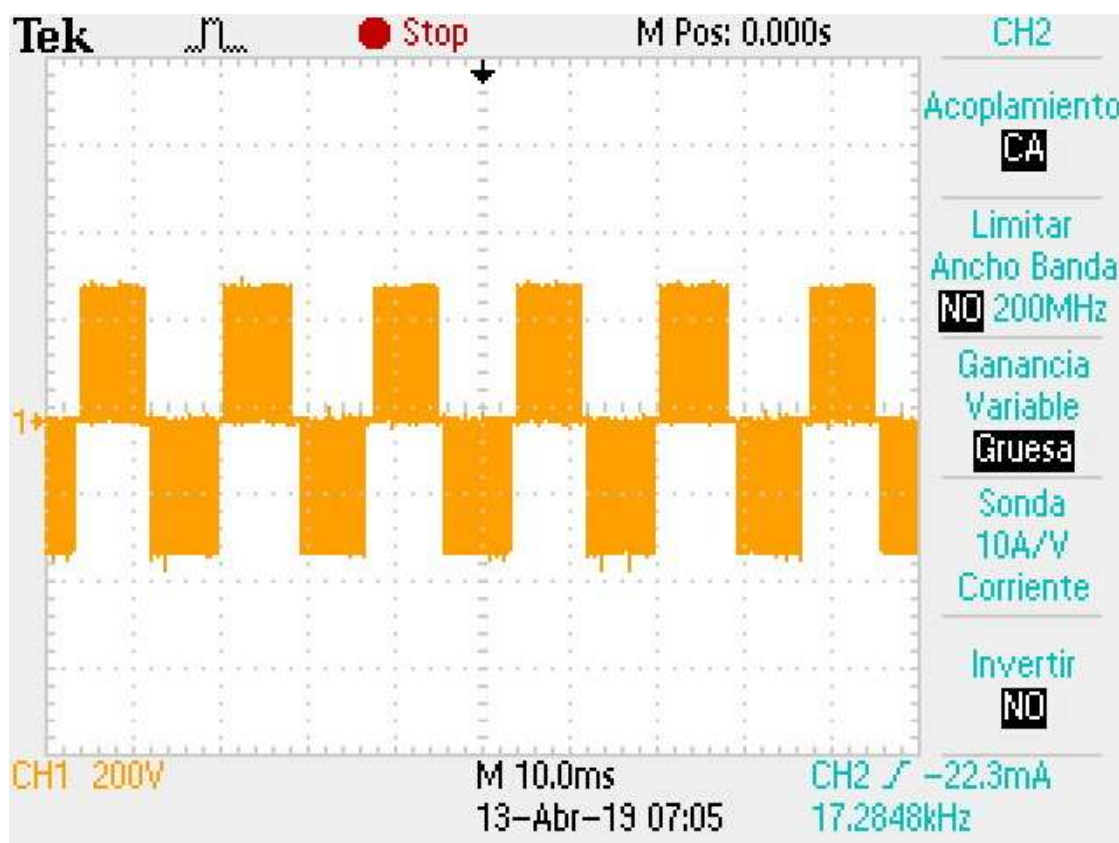


Figura 3.38. Voltaje V_{ab} con $f=60$ Hz e $I_m=97\%$ para carga R trifásica

Tabla 3.6. Datos de las pruebas con carga R trifásica

f [Hz]	Im%	Vbus [v]	Vab rms [v]	Vab rms simulación [v]	Er% Vab	V1 rms [v]	THD Vab
50	50	299.9	154.7	154.2	0.32	88.8	1.42
50	55	299.9	162.2	161.88	0.20	97.8	1.31
50	60	299.9	169.7	169.21	0.29	107	1.23
50	65	299	176.5	175.7	0.46	116.1	1.13
50	70	298.1	183.1	181.88	0.67	125.2	1.05
50	75	298.7	189.2	188.74	0.24	133.7	1
50	80	298.1	195.1	194.61	0.25	143	0.93
50	85	297.5	201.1	200.27	0.41	152	0.86
50	90	297.3	207.3	206	0.63	161	0.81
50	95	297.2	213.2	211.64	0.74	170	0.75
60	52	300.5	157.2	157.33	-0.08	92.3	1.37
60	57	300	165	164.61	0.24	101.2	1.3
60	62	299	171.7	171.25	0.26	110	1.17
60	67	299.3	178.6	178.33	0.15	119.5	1.1
60	72	298.6	184.5	184.55	-0.03	127.6	1.04
60	77	298.6	192.1	190.95	0.60	137.4	0.97
60	82	298.4	197.5	197.02	0.24	146	0.91
60	87	298.1	204.1	202.82	0.63	155	0.87
60	92	298.1	208.9	208.64	0.12	164	0.8
60	97	298.2	214.4	214.38	0.01	173	0.74

4. CONCLUSIONES Y RECOMENDACIONES

4.1. CONCLUSIONES

- Se creó un bloque de propiedad intelectual que se integra con una interfaz de usuario en el sistema en chip ZYBO para utilizarlo en aplicaciones SPWM de índice de modulación y frecuencia variable, de manera exitosa.
- La revisión bibliográfica permitió comprender el procedimiento de excitación de las SPWMs a un inversor trifásico para variar la frecuencia y el voltaje RMS en las líneas de un motor de inducción; y, la utilización de un ejemplo básico, que fueron el punto de partida para desarrollar todos los algoritmos descritos en este proyecto técnico con el propósito de comunicar, mediante protocolos AXI, el software del PS con el bloque IP de la PL.
- La metodología seguida en el trabajo de titulación permitió desarrollar la programación del sistema de procesamiento y de la lógica programable del sistema en chip ZYBO, en el lenguaje de programación C y VHDL, respectivamente; generar señales SPWM de índice de modulación y frecuencia variable que excitan un inversor trifásico, y validarlas en cargas R, RL y motor de inducción, cuyos resultados finales avalan el objetivo general de este trabajo de titulación.
- El bloque de propiedad intelectual SPWM desarrollado en VHDL, facilita reducir el tiempo de programación de aplicaciones más complejas, debido a que puede ser utilizado como uno o varios elementos de programación visual concurrentemente, cuyo elevado nivel de abstracción acelera el proceso de desarrollo, manteniendo el poder de procesamiento de la lógica programable.
- La aplicación desarrollada en ZYBO impide que los *flip-flops* ingresen a regiones de meta-estabilidad, debido a que el sistema digital configurado respeta los tiempos de *hold time* y *setup time*, lo cual es confirmado por los reportes de implementación de tiempo de Vivado Design Suite, eliminando la posibilidad de obtener lecturas erróneas de datos que desencadenen condiciones de peligro para los equipos y operadores del sistema.
- Se configuró un bloque de propiedad intelectual SPWM que facilitará al Laboratorio de Electrónica de Potencia, Control de Máquinas y Redes Eléctricas Inteligentes de la Escuela Politécnica Nacional crear una biblioteca de bloques de programación

para FPGAs, cuyo amplio potencial en el campo de electrónica de potencia y control de máquinas contribuirá al desarrollo de futuras investigaciones.

- En las pruebas con motor de inducción no se utilizaron frecuencias fundamentales menores a 20 hz, debido a que, al reducir la velocidad del motor con la frecuencia fundamental, se pierde paulatinamente el flujo de aire con el que el motor se refrigera por sí mismo. Por esta razón, las regiones de bajas frecuencias deben ser utilizadas de una manera temporal, caso contrario el motor se recalentará, pudiendo comprometer su vida útil.
- El desarrollo de la fase de simulación de la metodología permitió reproducir extensamente las características de las señales SPWMs en Vivado Design Suite y validarlas experimentalmente en el sistema en chip ZYBO, comprobando que las simulaciones del bloque de propiedad intelectual SPWM de la FPGA son fieles representaciones del sistema digital implementado físicamente.
- En el sistema ZYBO – Inversor trifásico se utilizó el voltaje entre líneas a la salida del inversor para realizar la validación global del proyecto técnico, debido a que esta onda de voltaje presenta características de: frecuencia fundamental, formas de onda pulsantes propias de la modulación SPWM, número de niveles, voltaje RMS y relación de modulación de frecuencia a partir del análisis FFT; permitiendo comprobar el funcionamiento correcto del sistema con una sola magnitud eléctrica de voltaje, en función de la variación del índice de modulación y frecuencia fundamental digitadas en la interfaz de usuario. Las cargas utilizadas en las pruebas de validación consisten en un motor trifásico de inducción y cargas trifásicas balanceadas R y RL, en las cuales se verificó exitosamente el funcionamiento del proyecto de titulación.

4.2. RECOMENDACIONES

- Durante la fase de validación se constató que el voltaje RMS trifásico de alimentación al laboratorio presenta variaciones, que se manifiestan en el lapso de horas, consecuentemente, el voltaje del bus DC no es fijo. La salida de voltaje RMS entre líneas del inversor está en función de la variable controlada, índice de modulación, y el voltaje del bus DC. Para automatizar la salida de voltaje RMS del inversor se recomienda incorporar, a futuro, una funcionalidad en ZYBO que permita sensar el voltaje del bus DC con el objeto de utilizarlo como un parámetro

de cálculo que compense sus fluctuaciones en las salidas de voltaje RMS del inversor.

- ZYBO puede ser programado en varios lenguajes y está orientado para ser utilizado por ingenieros electrónicos en la configuración de la FPGA y por ingenieros de software en la programación de las sentencias de código del sistema de procesamiento. El sistema de procesamiento (Dual-core ARM Cortex-A9) puede ser programado en C sin necesidad de conocer la arquitectura de un FPGA, por lo que, se sugiere crear un equipo mixto de investigación, que, enfocados en sus respectivas áreas de conocimiento, contribuyan en forma coordinada y simultánea a la aceleración del desarrollo de aplicaciones para la FPGA y los procesadores ARM Cortex-A9 de ZYBO.
- De acuerdo a los términos estipulados en la ley, reglamentos y normas vigentes, se extiende a la Escuela Politécnica Nacional la potestad de reproducir el bloque de propiedad intelectual SPWM, por lo que, se recomienda su difusión académica en beneficio de la investigación en áreas de electrónica de potencia y control de máquinas; y, la comercialización con los consecuentes beneficios de licencias y regalías.

5. REFERENCIAS BIBLIOGRÁFICAS

- [1] A. Cozma y E. Cigan, «FPGA-based Systems Increase Motor-control Performance,» *Analog Dialogue*, vol. 49, 2015.
- [2] H. Gürsoy y M. Önder, «Control system implementation on an FPGA platform,» 2016. [En línea]. Available: <https://www.sciencedirect.com/science/article/pii/S2405896316327252>.
- [3] M. Curkovic, K. Jezernik y R. Horvat, «FPGA-Based Predictive Sliding Mode Controller of a Three-Phase Inverter,» 2013. [En línea]. Available: <https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=6249747&tag=1>.
- [4] L. Crockett, R. Elliot, M. Enderwitz y R. Stewart, *The Zynq Book*, Glasgow: Strathclyde Academic Media, 2014.
- [5] Xilinx, «Zynq-7000 Soc Data Sheet: Overview,» 2018. [En línea]. Available: https://www.xilinx.com/support/documentation/data_sheets/ds190-Zynq-7000-Overview.pdf.
- [6] R. Kastner, J. Matai y S. Neuendorffer, *Parallel Programming for FPGAs*, 2018.
- [7] M. Rashid, *Power Electronics Handbook*, Canada: Academic Press, 2001.
- [8] N. Mohan, T. Undeland y W. Robbins, *Electrónica de potencia*, 3 ed., México: Mc Graw Hill, 2009.
- [9] S. Gamboa y P. Quelal, *Diseño y Construcción de un Variador de Velocidad con el Microcontrolador 80C196MC*, Quito: Escuela Politécnica Nacional, 2004.
- [10] S.-H. Kim, *Electric Motor Control DC, AC, and BLDC Motors*, Amsterdam: Elsevier, 2017.
- [11] A. Hughes, *Electric Motors and Drives, Fundamentals, Types and Applications*, 3 ed., Oxford: Elsevier, 2006.
- [12] ABB, «Technical Guide Book,» 2014. [En línea]. Available: <https://bit.ly/2I3Bo8R>.

- [13] Weg, «Induction Motors Fed by PWM Frequency Inverters,» 2019. [En línea]. Available: <https://static.weg.net/medias/downloadcenter/hcb/h20/WEG-induction-motors-fed-by-pwm-frequency-inverters-50029350-brochure-english-web.pdf>.
- [14] Digilent, «Zybo FPGA Board Reference Manual,» 2016. [En línea]. Available: https://reference.digilentinc.com/_media/zybo:zybo_rm.pdf.
- [15] Schneider Electric, «Altivar 71 Variable Speed Drives for Synchronous and Asynchronous Motors,» 2015. [En línea]. Available: <https://bit.ly/2GAf2cG>.
- [16] Schneider Electric, «Altivar 212 & S-Flex Enclosed Variable Speed Drives,» 2014. [En línea]. Available: <https://bit.ly/2thZRNE>.
- [17] ABB, «ABB General Purpose Drives,» 2018. [En línea]. Available: <https://bit.ly/2UOn5GF>.
- [18] Eaton, «The Next Generation of OEM Drives,» 2013. [En línea]. Available: <https://bit.ly/2SLXcd8>.
- [19] Weg, «Manual del Convertidor de Frecuencia CFW-09,» 2018. [En línea]. Available: <https://bit.ly/2TNwv5r>.
- [20] Fairchild Semiconductor, «Single-Channel: 6N137, HCPL2601, HCPL2611, Datasheet,» 2011. [En línea]. Available: <http://www.farnell.com/datasheets/1057512.pdf>.
- [21] ST, «AN2738 Application note, L6390 half-bridge gate driver,» 2009. [En línea]. Available: <https://bit.ly/2KY80jR>.
- [22] Component search engine, «L6390D - STMicroelectronics, PCB Footprints and Schematic Symbols,» 2017. [En línea]. Available: <https://componentsearchengine.com/L6390D/STMicroelectronics>.
- [23] P. Ashenden, Digital Design: An Embedded Systems Approach Using VHDL, Burlington: Elsevier, 2008.
- [24] V. Taraate, PLD Based Design with VHDL, Singapore: Springer, 2017.
- [25] Xilinx, Vivado Design Suite User Guide Implementation, 2018.

- [26] Analog Devices, «Pulse Width Modulator in AC Motor Control Applications,» 2016. [En línea]. Available: <https://www.analog.com/media/en/technical-documentation/application-notes/AN-1407.pdf>.
- [27] ST, «STEVAL-IHM023V3, 1 kW 3-phase motor control evaluation board, overview,» 2019. [En línea]. Available: <https://www.st.com/en/evaluation-tools/steval-ihm023v3.html#overview>.
- [28] S. Churiwala, Designing with Xilinx FPGAs Using Vivado, Hyderabad: Springer, 2017.
- [29] ST, «STGB10H60DF, STGF10H60DF, STGP10H60DF Trench gate field-stop IGBT, H series 600 V, 10 A high speed Datasheet,» 2013. [En línea]. Available: <https://www.st.com/resource/en/datasheet/stgp10h60df.pdf>.
- [30] Fluke, «Fluke 87V Industrial Multimeter, specifications,» 2019. [En línea]. Available: <https://www.fluke.com/en/product/electrical-testing/digital-multimeters/fluke-87v>.
- [31] Fluke, «Accuracy, resolution, range, counts, digits and precision, for Fluke Multimeters, specifications,» 2019. [En línea]. Available: <https://www.fluke.com/en/learn/best-practices/test-tools-basics/digital-multimeters/accuracy-resolution-range-counts-digits-precision>.
- [32] Digilent, «Installing Vivado and Digilent Board Files, instalation manual,» 2018. [En línea]. Available: <https://reference.digilentinc.com/vivado/installing-vivado/start>.
- [33] FTDI, «CDM Uninstaller 1.4 - Windows Device Driver Uninstaller,» 2010. [En línea]. Available: <http://www.ftdichip.com/Support/Utilities.htm#CDMUninstaller>.
- [34] FTDI, «CDM Driver,» 2017. [En línea]. Available: https://www.ftdichip.com/Drivers/CDM/CDM21228_Setup.zip.
- [35] Digilent, «Zybo development platform, overview,» 2018. [En línea]. Available: <https://reference.digilentinc.com/reference/programmable-logic/zybo/start>.
- [36] L. Crockett, R. Elliot, M. Enderwitz y D. Northcote, The Zynq Book Tutorials for Zybo and Zedboard, Glasgow: Strathclyde Academic Media, 2015.

ANEXOS

ANEXO A. Lista de abreviaturas

ANEXO B. Guía de usuario

ANEXO A

Lista de abreviaturas

ACP	Puerto de aceleración coherente (<i>Accelerator coherency port</i>)
APU	Unidad de procesamiento de aplicaciones (<i>Application processing unit</i>)
ASIC	Circuito integrado de aplicación específica (<i>Application specific integrated circuit</i>)
AXI	Interfaz extensible avanzada (<i>Advanced extensible interface</i>)
BSP	Paquetes de soporte de plataforma (<i>Board support package</i>)
CLB	Bloque de lógica configurable (<i>Configurable logic block</i>)
CPU	Unidad central de procesamiento (<i>Central processing unit</i>)
DSP	Procesamiento digital de señal (<i>Digital signal processing</i>)
Emf	Fuerza electromotriz (<i>Electromotive force</i>)
FF	<i>Flip-flop</i>
FFT	Transformada rápida de Fourier (<i>Fast Fourier transform</i>)
FPGA	Arreglo de compuertas lógicas programables en campo (<i>Field Programmable Gate Array</i>)
HDL	Lenguaje de descripción de hardware (<i>Hardware description language</i>)
Im	Índice de modulación
IGBT	Transistor bipolar de compuerta aislada (<i>Insulated-gate bipolar transistor</i>)
IOB	Bloque de entrada y salida (<i>Input/output block</i>)
JTAG	<i>Joint test action group</i>
IP	Propiedad intelectual (<i>Intellectual property</i>)
LUT	Tabla de búsqueda (<i>Lookup table</i>)
mf	Relación de modulación de frecuencia

MOSFET	Transistor de efecto de campo de metal-óxido-semiconductor (<i>Metal-oxide semiconductor field effect transistor</i>)
PL	Lógica programable (<i>Programmable logic</i>)
PMOD	Módulo periférico (<i>Peripheral module</i>)
PS	Sistema de procesamiento (<i>Processing system</i>)
RAM	Memoria de acceso aleatorio (<i>Random access memory</i>)
RMS	Medio cuadrático (<i>Root mean square</i>)
RTL	Registro a nivel de transferencia (<i>Register transfer level</i>)
SDK	<i>Software Development Kit</i>
SOC	Sistema en chip (<i>System on chip</i>)
SPWM	Modulación por ancho de pulso senoidal (<i>Sinusoidal pulse width modulation</i>)
THD	Distorsión armónica total (<i>Total harmonic distortion</i>)
THS	<i>Total hold slack</i>
TNS	<i>Total negative slack</i>
UART	Receptor/transmisor asíncrono universal (<i>Universal Asynchronous Receiver/Transmitter</i>)
VHDL	Lenguaje de descripción de hardware de circuito integrado de muy alta velocidad (<i>Very high-speed integrated circuit hardware description language</i>)
WHS	<i>Worst hold slack</i>
WNS	<i>Worst negative slack</i>
ZYBO	Tarjeta Zynq (<i>Zynq board</i>)

ANEXO B

Guía de usuario

B.1. Requisitos de software

- Instalar la versión gratis del software Vivado 2018.2 y SDK, webpack, de la página web de Xilinx. Mantener la dirección de instalación por defecto para que se instale en el disco duro C. El archivo pdf del CD, “*Installing Vivado and Digilent Board Files*”, tomado de [32] guiará al usuario en la instalación de Vivado 2018.2 y SDK, seguir únicamente los pasos de los numerales 1.
- Conectar el SOC ZYBO a la computadora con un cable USB con capacidad de envío de datos a través del periférico J11, esperar a que se instalen automáticamente los drivers, luego revisar que existe solo un puerto COM en el administrador de dispositivos (*Device manager*). Si se muestran dos puertos COM, entonces se debe reinstalar los drivers de FTDI con los siguientes pasos:
 1. Desconectar todos los periféricos de la computadora.
 2. Correr el programa CDM_UninstallerGUI [33] localizado en la carpeta “FTDI\CDMUninstaller_v1.4”.
 3. Escribir 0403 en Vendor ID en la ventana CDM Uninstaller y dejar en blanco a Product ID (Figura B.1.).

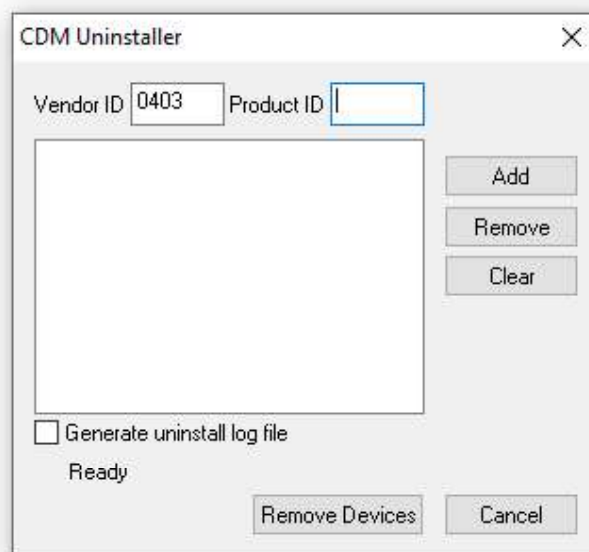


Figura B.1. Ventana del ejecutable CDM Uninstaller

4. Hacer click en *Add*.
 5. Hacer click en *Remove Devices*.
 6. Leer la notificación que explica que todos los drivers fueron removidos. Si la instalación del driver nunca se completó, se indicará que no hubo drivers para poder removerlos.
 7. Reiniciar la computadora.
 8. Ir a la carpeta “FTDI” y correr el instalador CDM21228_Setup [34] del driver de FTDI como administrador.
 9. Conectar ZYBO a la computadora con un cable USB con capacidad de envío de datos a través del periférico J11, esperar a que se instalen automáticamente los drivers y luego revisar que solo exista un puerto COM asignado en el administrador de dispositivos.
- Buscar en el CD la carpeta “vivado-boards-master\new\board_files”, dentro de esta dirección se encuentran las tarjetas de Digilent, copiar todo el contenido y pegarlo dentro de la carpeta “C:\Xilinx\Vivado\2018.2\data\boards\board_files”.
 - Instalar el ejecutable “Instalador_SPWM_GUI” que se encuentra en la carpeta “SPWM_GUI\SPWM_GUIfor_redistribution”. Se sugiere seleccionar la opción de crear un acceso directo en el escritorio para facilidad de uso.
 - Instalar el software PSIM con versión 9.0 o superior.

Nota Importante: Si se decide copiar las carpetas del CD a un computador, se debe garantizar que la carpeta de destino, donde se encuentra todo el proyecto, debe ubicarse en la computadora en una dirección que no tenga ningún espacio.

B.2. Pasos para programar y utilizar la aplicación de ZYBO

1. Abrir el proyecto de Vivado “proyecto_spwm”, que se encuentra en la dirección “SPWM\version1\proyecto_spwm” del CD.
2. Ir a File en la barra de menú, seleccionar Launch SDK, y dar click en ok.
3. Verificar que la tarjeta ZYBO tenga los siguientes requerimientos de conexión (Figura B.2.):
 - Jumper JP7 en selección USB.

- Jumper JP5 en selección QSPI.
- El cable USB (ZYBO-PC) se conecta desde el puerto J11 hacia un puerto USB de la computadora. Este cable USB debe presentar capacidades de envío de datos.

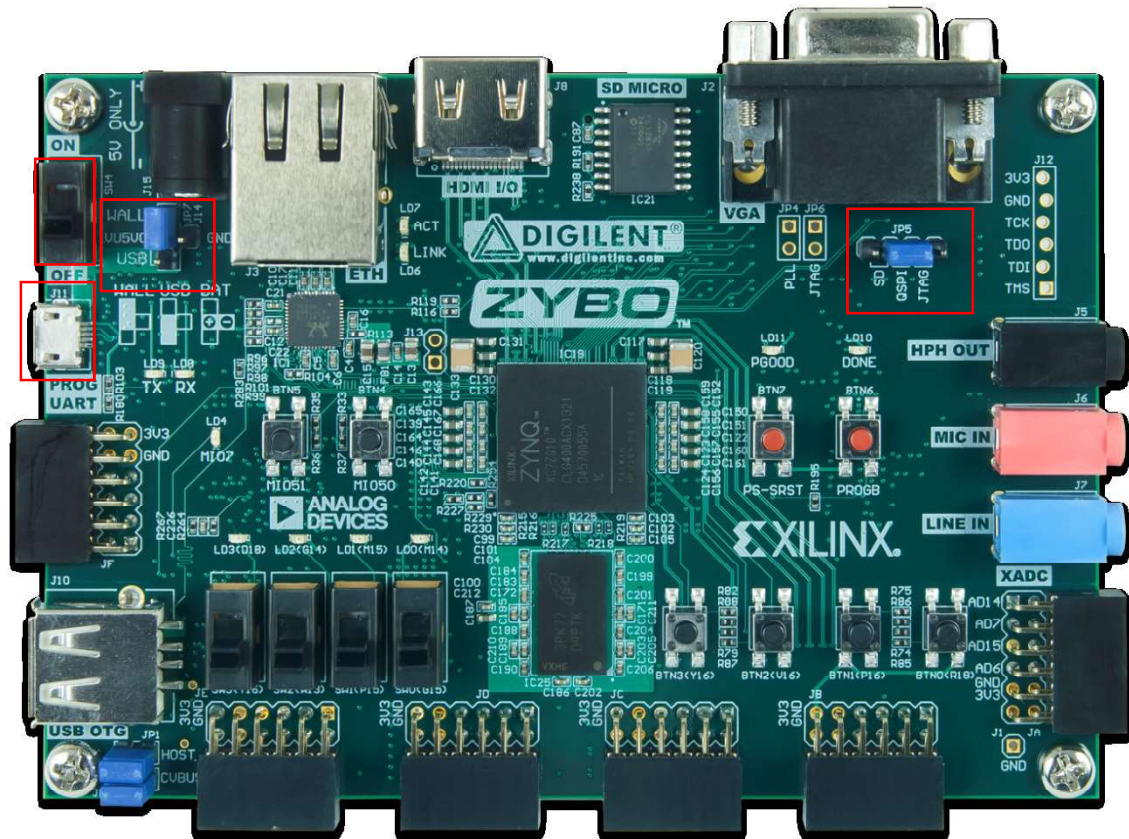


Figura B.2. Jumpers, interruptor y puerto USB a verificar para conexión [35]

4. Encender a ZYBO con el interruptor SW4 en posición ON (Figura B.2.).
5. Verificar que SDK haya cargado.
6. Ir a la barra de menú, seleccionar Xilinx > Program Flash and configurar lo siguiente (Figura B.3.):
 - Hardware Platform = diseño_PL_PS_SPWM_wrapper_hw_platform_0
 - Connection = Local
 - Device = Auto Detect
 - Image File = dirección del archivo BOOT.bin, "SPWM\version1\proyecto_spwm\proyecto_spwm.sdk\fsbl_0\bootimage\BOOT.bin"

- OFFSET = 0x00000000
- Flash Type = qspi_single
- FSBL File = dirección del archivo fsbl_0.elf, "SPWM\version1\proyecto_spwm\proyecto_spwm.sdk\fsbl_0\Debug\fsbl_0.elf"

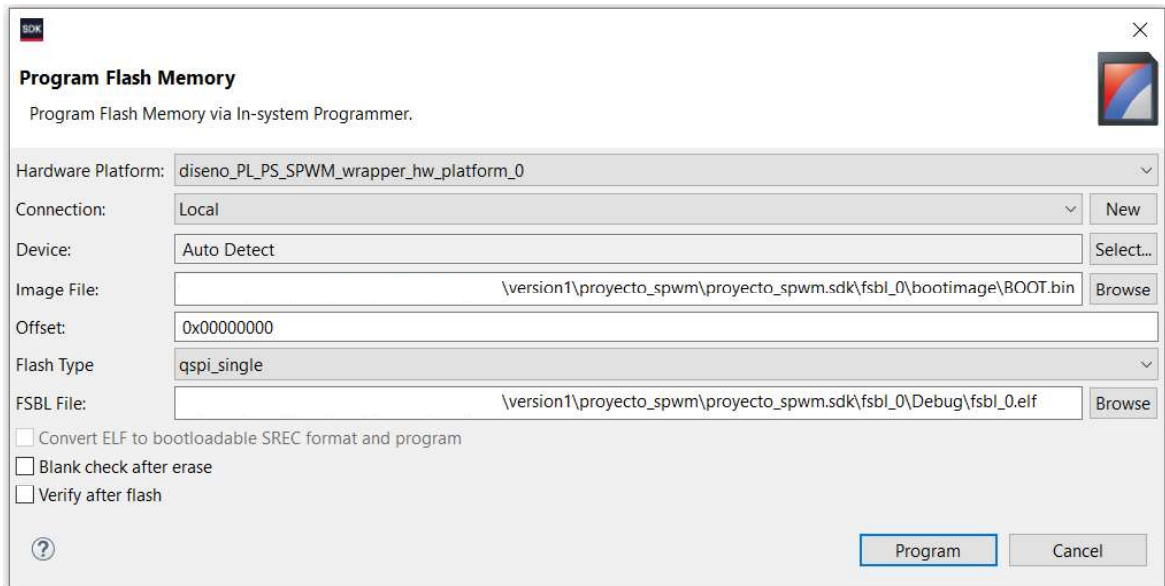


Figura B.3. Ventana de programación de la memoria flash de ZYBO

7. Dar click en Program.
8. Verificar que se haya programado la memoria flash correctamente en el mensaje final de la consola de SDK (Figura B.4.).

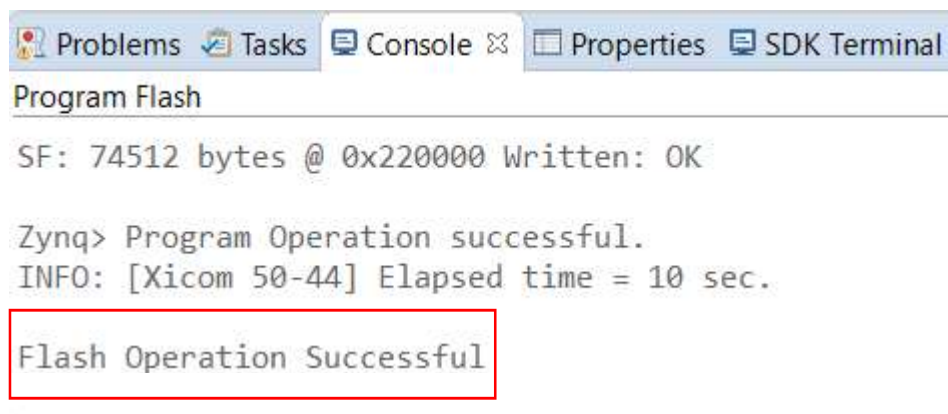


Figura B.4. Mensaje de confirmación de programación de memoria flash

9. Cerrar las aplicaciones SDK y Vivado.
10. Apagar y encender a ZYBO mediante el interruptor sw4.

11. Verificar el puerto de conexión serial USB (COM) de ZYBO en el administrador de dispositivos (Figura B.5.).

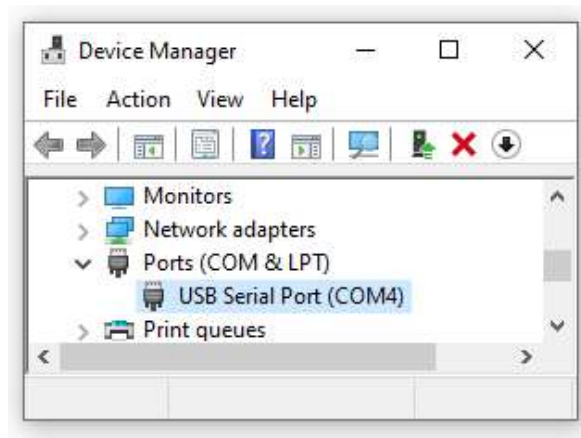


Figura B.5. Puerto de conexión serial

12. Correr el programa de la interfaz gráfica instalada previamente en la computadora.
13. Verificar el puerto COM correspondiente a ZYBO en la interfaz gráfica, elegirlo y dar click en el botón 'conectar' (Figura B.6.).

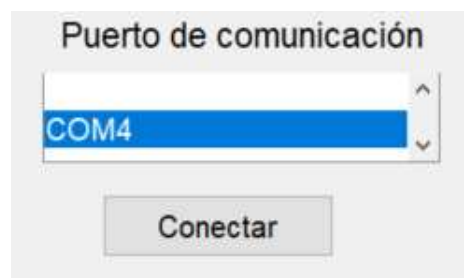


Figura B.6. Conexión de la interfaz gráfica con ZYBO mediante el puerto COM correspondiente

14. Ingresar los datos solicitados por la interfaz en uno de los recuadros blancos mediante el uso del teclado y un *enter* (Figura B.7.). Tomar en consideración que no existen rampas de aceleración ni desaceleración, por lo que se recomienda acelerar o desacelerar el motor paulatinamente.
 - 14.1. En el recuadro con el numeral 1, ingresar la frecuencia fundamental del sistema en hercios con un *enter*, luego observar la confirmación del dato ingresado en el recuadro "Variables del sistema" en el numeral 1, adicionalmente se mostrará la frecuencia de la onda portadora en la misma línea.

14.2. En el recuadro con el numeral 2 (se habilita después de realizar el paso anterior), ingresar el índice de modulación porcentual del sistema con un *enter*, luego observar la confirmación del dato ingresado en el recuadro “Variables del sistema” en el numeral 2.

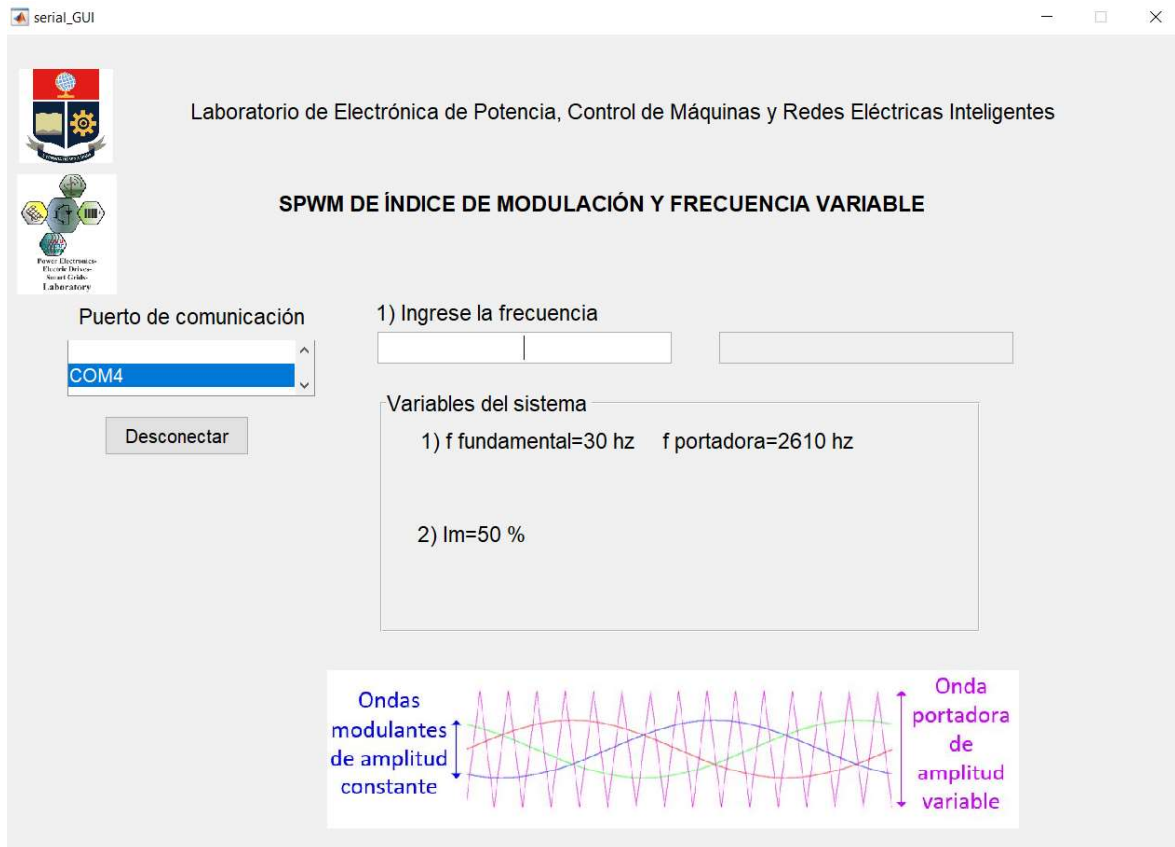


Figura B.7. Interfaz gráfica

15. Revisar las señales en el puerto Pmod estándar JE equivalentes a la Figura B.8. y organizadas como se muestra en la Tabla B.1.

JE

VCC	GND	SPWM S6	SPWM S3	SPWM S4	SPWM S1
VCC	GND	PWM Portadora	PWM R	SPWM S2	SPWM S5

Figura B.8. Pines de salida del puerto Pmod estándar (JE) con sus respectivas señales

Tabla B.1. Pines del puerto Pmod estándar JE con sus respectivas señales

Nombre de señal del puerto Pmod estándar JE	Pin de puerto Pmod estándar JE
SPWM S1	Pin 1
SPWM S4	Pin 2
SPWM S3	Pin 3
SPWM S6	Pin 4
SPWM S5	Pin 7
SPWM S2	Pin 8
PWM R	Pin 9
PWM Portadora	Pin 10

16. Finalizar el uso de la interfaz gráfica, con un click en el botón 'desconectar' y cerrar la ventana. Es importante notar que, aunque se desconecte la interfaz, ZYBO sigue ejecutando el PS y PL.

17. Desconectar la energía a ZYBO con la posición OFF del interruptor SW4.

Notas:

- Para utilizar la aplicación de nuevo, luego de haber programado la memoria flash de ZYBO con el paso 6 y 7, se debe revisar los requerimientos del paso 3 y seguir los pasos del manual a partir del numeral 10.
- Los códigos fuente de la aplicación de ZYBO e interfaz gráfica se encuentran en las siguientes direcciones:
 - "SPWM\version1\ip_repo\spwm_1.0\hdl", para el bloque de propiedad intelectual.
 - "SPWM\version1\proyecto_spwm\proyecto_spwm.srcs\constrs_1\new", archivo XDC para configuración de las restricciones de las salidas para el puerto Pmod estándar y el led 0.
 - "SPWM\version1\proyecto_spwm\proyecto_spwm.sdk\proyecto_software_spwm\src", para el software del PS programado en C.
 - "SPWM_GUI" para el archivo de Matlab de la interfaz gráfica (serial_GUI.m) y el archivo FIG (serial_GUI.fig).

- Para modificar o volver a crear toda la aplicación, a partir de los códigos fuente, se sugiere revisar el tutorial “*Exercise 4A Creating IP in HDL*” del libro *The Zynq Book Tutorials for Zybo and Zedboard* [36], disponible en la web gratuitamente en versión pdf, en donde se guía al usuario a través de la creación de los proyectos, configuración y empaquetamiento de un bloque IP, utilización del *Block Design* de Vivado para exportar el hardware a SDK, importación del código fuente del PS y conexión por comunicación serial UART utilizando el terminal propio de SDK.

B.3. Pasos para modificar el programa del sistema de procesamiento

El PS es el maestro que controla al bloque IP SPWM, por lo tanto, es de interés aprender como modificar su programa e incluir nuevos algoritmos o funciones en su software.

Se debe mencionar que las variables f , e I_m son las entradas principales de funcionamiento del software. En el programa del PS se puede identificar y desechar rápidamente el algoritmo de interacción con el usuario y manipular directamente las variables f e I_m como mejor se adapten al nuevo enfoque del uso del bloque IP.

A continuación, se muestran los pasos a seguir para modificar el programa del PS:

1. Abrir el proyecto de Vivado “proyecto_spwm”, que se encuentra en la dirección “SPWM\version1\proyecto_spwm” del CD.
2. Ir a File en la barra de menú, seleccionar Launch SDK, y dar click en ok.
3. Agregar el algoritmo deseado dentro de la función *main* o editar las constantes modificables mostradas en la Figura B.9. en el programa “sw_spwm.c” ubicado en SDK dentro de la ventana Project Explorer (Figura B.10.) en la dirección “proyecto_software_spwm/src”.

```
//CONSTANTES MODIFICABLES
#define mf 87 //para spwm sincrona de carga trifasica se recomienda un numero impar multiplo de 3
#define f_min 5 //[hz]
#define f_max 60 //[hz]
#define Im_min 10 //[%]
#define Im_max 500 //[%]
//CONSTANTES MODIFICABLES
```

Figura B.9. Constantes modificables del PS

4. Guardar y compilar con los botones que se muestran enmarcados con color rojo en la Figura B.10.

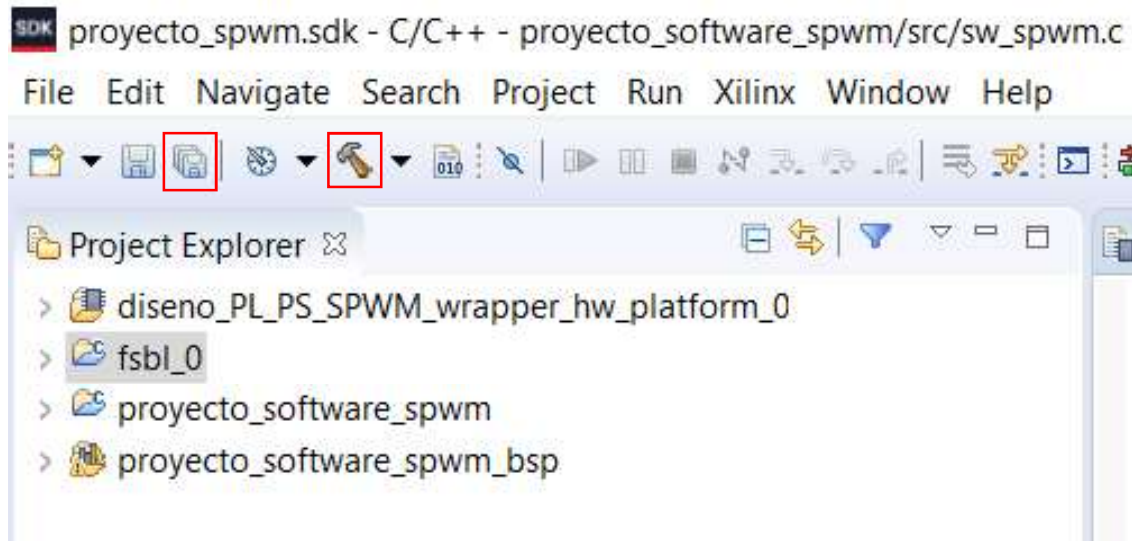


Figura B.10. Porción superior izquierda de SDK de Xilinx

5. Dar click derecho sobre la carpeta `fsbl_0` dentro de la ventana Project Explorer y escoger con un click la opción "Create Boot Image".
6. En la ventana emergente "Create Boot Image" seleccionar la dirección del archivo "proyecto_software_spwm.elf" en "Boot image partitions".
7. Dar click en *delete*.
8. Dar click en *Add*.
9. Dar click en "Browse...".
10. Ir a la dirección "SPWM \version1\proyecto_spwm\proyecto_spwm.sdk\proyecto_software_spwm\Debug" del CD, seleccionar el archivo "proyecto_software_spwm.elf" y dar click en *Open*.
11. Dar click en OK.
12. Dar click en "Create Image".
13. Dar click en OK.
14. Utilizar el nuevo programa siguiendo las instrucciones de **Pasos para programar y utilizar la aplicación de ZYBO**.

B.4. Pasos para simular el voltaje RMS entre líneas de un inversor trifásico excitado por la modulación SPWM de ZYBO

La simulación de PSIM de la modulación SPWM de ZYBO puede ser utilizada como herramienta de validación del voltaje RMS entre las líneas de un inversor trifásico. A continuación, se muestra el procedimiento para simular el voltaje RMS entre las líneas de un inversor trifásico.

1. Abrir PSIM.
2. Abrir el archivo "SPWM_INVERSOR_TRIFASICO" de la carpeta "Simulacion_PSIM" (Figura B.11).
3. Abrir la ventana "parametros_Im_f_Vbus.txt" mediante doble click en el bloque "File", ubicado en la esquina superior izquierda de los esquemáticos de simulación, y modificar los siguientes valores:
 - 3.1. Frecuencia fundamental (f) que ha sido ingresada a ZYBO.
 - 3.2. Índice de modulación porcentual (I_m) que ha sido ingresado a ZYBO.
 - 3.3. Voltaje del bus dc (V_{bus}) medido en el inversor.
 - 3.4. Relación de modulación de frecuencia (m_f , 87 es el dato programado en ZYBO el cual consta en el CD).
 - 3.5. Voltaje de saturación (V_{sat}) de los interruptores en los ramales del inversor trifásico.
4. Cerrar la ventana "parametros_Im_f_Vbus.txt" luego de haber realizado las modificaciones.
5. Dar doble click en el bloque "Simulation Control" (figura de reloj) y modificar el campo "Total Time" por lo menos con el doble de tiempo del periodo de la fundamental.
6. Cerrar la ventana "Simulation Control".
7. Dar click en Simulate > Run Simulation en la barra de menú.
8. Observar el dato del voltaje RMS entre las líneas a y b, que se mostrará en el voltímetro V_{ab_rms} (Figura B.11.).

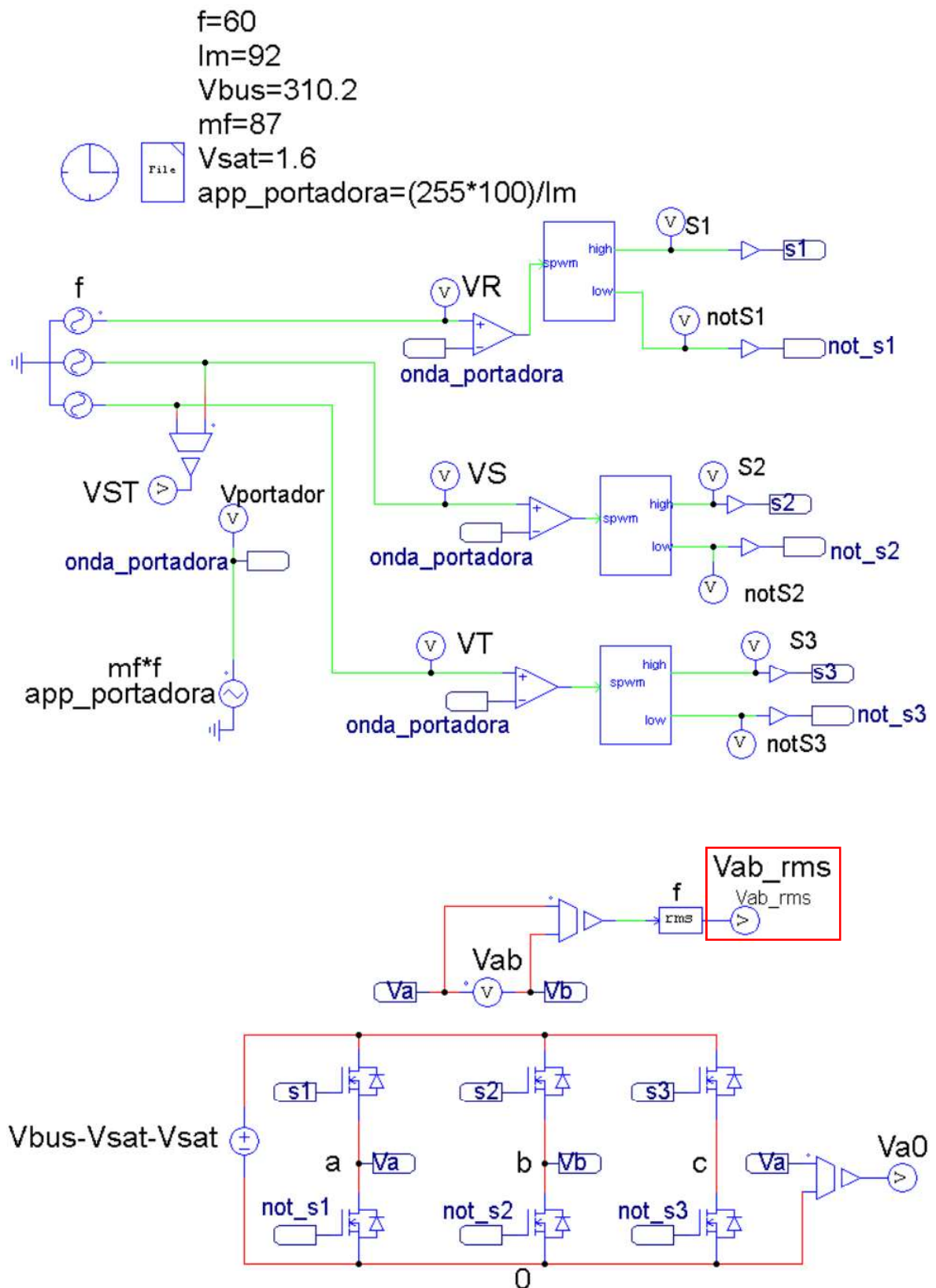


Figura B.11. Simulación en PSIM de la modulación SPWM de ZYBO en un inversor trifásico

ORDEN DE EMPASTADO