

 Institución Universitaria	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-27

HARDWARE IN THE LOOP PARA LA SIMULACIÓN DE CONVERTIDORES ELECTRÓNICOS DE POTENCIA

José Joaquín Guajo Trujillo

Ingeniería Electrónica

Sergio Serna

INSTITUTO TECNOLÓGICO METROPOLITANO

03/03/2017

	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

RESUMEN

El siguiente informe presenta la información relevante para el desarrollo del proyecto, donde el objeto principal del trabajo es la realización de un prototipo mediante hardware in the loop para la simulación de convertidores de potencia, con el fin de poder tener a la mano un patrón a pequeña escala que permita visualizar resultados mediante la simulación en tiempo real y posteriormente llevar la planta o el sistema a sus conexiones físicas reales.

El proyecto se realizó iniciando con la investigación y la posterior aplicación del lenguaje VHDL, importante ya que cuenta con las librerías necesarias que nos permitieron obtener la simulación HIL. Posteriormente se realiza un estudio de la tarjeta de desarrollo FPGA (Nexys 4), la cual contiene las características básicas como frecuencia de muestreo, puertos de entradas y salidas, entre otros. Se obtiene el modelo matemático de cada planta (convertidor DC – DC). A través del método de Euler, se resuelven las ecuaciones diferenciales obtenidas de cada sistema, esto, por medio del lenguaje de descripción de hardware VHDL.

Los temas descritos fueron muy importantes para el análisis, investigación y posterior desarrollo del prototipo, donde los resultados fueron los esperados ya que para cada convertidor se logró obtener una muestra de los datos a pequeña escala.

Cabe resaltar que el prototipo se implementó para los tres principales tipos de convertidores: Buck, Boost y Buck – Boost, y que su diseño e implementación se basó en obtener a su salida el voltaje y la corriente promediada a través del desarrollo de las ecuaciones diferenciales. Como anexo, se realiza el sistema de control para cada planta, obteniendo buenos resultados en la simulación, pero que, por cuestiones de muestreo, los resultados físicos en la simulación HIL no fueron los correctos.

Palabras claves: Hil, VHDL, convertidores de potencia, ecuaciones diferenciales, modelo matemático, voltaje, corriente, simulación, Fpga.

	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

RECONOCIMIENTOS

Quiero agradecerle en primera instancia a mi familia, amigos y allegados que me dieron apoyo moral y que siempre estuvieron atentos en cuanto a la evolución y resultados del producto.

Al docente y asesor Sergio Serna por su apoyo moral, económico y académico para lograr el objetivo trazado, permitiendo obtener los resultados esperados y donde este producto puede ser el camino para un futuro proyecto perteneciente al tema.

De igual manera a los profesores Daniel Montoya y Luis Castaño por sus asesorías frente a las inquietudes presentadas a lo largo del desarrollo del prototipo.

Al grupo de Investigación SIMER del parque i, liderado por Sergio Serna que me permitió contar con todos los equipos, instrumentos, software y materiales necesarios para la ejecución del proyecto.

 Institución Universitaria	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

ACRÓNIMOS

VHSIC Very High Speed Integrated Circuit

HDL (Hardware Description Language)

VHDL VHSIC + HDL

FPGA Field Programmable Gate Array

HIL Hardware in the Loop

PMOD Peripheral Modules

CONVERTER D/A Convertidor Digital a Analógico

FEM Fuerza Electromotriz

PWM Pulse Width Modulation

VC Voltaje en el condensador

Vo Voltaje de salida

IL Corriente del inductor

VG Voltaje en el generador (fuente)

Ts Tiempo de muestreo

PARQUE i Parque de investigaciones ITM

SIMER Semillero de investigación de microredes y energía renovable

 Institución Universitaria	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

TABLA DE CONTENIDO

1. INTRODUCCIÓN.....	6
2. MARCO TEÓRICO.....	7
2.1 ELECTRONICA DE POTENCIA.....	7
2.2 CONVERTIDOR BUCK.....	9
2.3 CONVERTIDOR BOOST.....	9
2.4 CONVERTIDOR BUCK - BOOST.....	11
2.5 MODELO MATEMATICO CONVERTIDORES DC - DC.....	12
2.6 VHDL.....	27
2.7 FPGA.....	28
2.8 HARDWARE IN THE LOOP (HIL).....	29
3. METODOLOGIA.....	32
3.1 ELEMENTOS DE VHDL.....	32
3.2 HARDWARE Y SOFTWARE.....	34
3.3 CALCULO COMPONENTES DE LOS CONVERTIDORES.....	35
3.4 CODIGO VHDL DEL PROTOTIPO.....	38
4. RESULTADOS Y DISCUSION.....	44
4.1 CONVERTIDOR BUCK.....	44
4.2 CONVERTIDOR BOOST.....	56
4.3 CONVERTIDOR BUCK - BOOST.....	68
5. CONCLUSIONES, RECOMENDACIONES Y TRABAJO FUTURO.....	79
REFERENCIAS.....	81
APENDICE.....	82

	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

1. INTRODUCCIÓN

La electrónica de potencia abarca un mercado amplio, su utilidad va desde el control a conmutación y conversión de energías. Posee un sin número de aplicaciones en la industria y en el área académico, donde, al ser un tema delicado por su gran importancia, es primordial que su proceso de modelado, diseño e implementación física sea desarrollado correctamente para posteriormente obtener el resultado esperado y efectuarlo en la aplicación requerida.

En muchas áreas, especialmente en la de Ingeniería es fundamental en muchas ocasiones crear prototipos de sistemas, plantas, controladores que nos permitan verificar su funcionamiento apropiado, esto con el fin de poder llevar a cabo la ejecución del sistema a gran escala, disminuyendo el riesgo de que no funcione debido a que anteriormente, gracias al prototipo se puede establecer que las variables del sistema y desarrollo, cumplen con las especificaciones del diseño.

Para poder obtener una simulación en tiempo real que nos describa el comportamiento a pequeña escala del sistema, existe una técnica llamada Hardware in the loop, que, mediante procesos de modelamiento, lenguaje VHDL y una tarjeta de desarrollo, podemos crear el prototipo de cualquier tipo de planta, en este caso, los convertidores DC a DC, específicamente el Buck, Boost y Buck – Boost.

El objetivo principal del producto realizado es dar a entender que en ingeniería es significativo tener a una escala menor de cualquier tipo de sistema, esto con el fin de darnos una idea de los posibles resultados que tendremos cuando ejecutemos la planta con sus conexiones reales.

Finalmente, es primordial saber que, para poder llevar a cabo cualquier idea o propósito, es fundamental investigar los temas que nos van a facilitar el desarrollo del proyecto, en este caso, investigar y aprender acerca del lenguaje VHDL, FPGA, convertidores DC a DC, HIL, entre otros, condujeron a la finalización correcta del producto.

	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

2. MARCO TEÓRICO

2.1 ELECTRONICA DE POTENCIA

La electrónica de potencia es una de las ramas de la ingeniería eléctrica donde se combinan la electrónica y el control. Su base es la conversión de energía eléctrica de una forma a otra más eficiente, compacta y robusta para una aplicación eficaz, todo esto, con la ayuda de dispositivos de conmutación, diodos, inductores, capacitores, sistemas de control, entre otros. Se destaca hoy día por su crecimiento y su aplicabilidad en la industria.

El control se encarga de las características dinámicas del sistema en lazo cerrado, por su parte la electrónica se ocupa de los dispositivos y circuitos de estado sólido requeridos en el procesamiento de señales para cumplir con los objetivos de control deseados. Su principal ventaja se da debido al modo de conmutación que maneja para la conversión de potencia, representando una alta eficiencia, que puede ser 96% a 99%.

Aborda cuatro tipos de conversiones posibles que son:

- **Conversión alterna - continua (Rectificadores)**
- **Conversión alterna - alterna (Cicloconvertidores)**
- **Conversión continua - continua (Choppers)**
- **Conversión continua - alterna (Inversores)**

En base a estos tipos de conversión, nombraremos algunas aplicaciones importantes que son base fundamental para el correcto desenvolvimiento de procesos industriales, equipos de cómputo, transporte, equipos de comunicación, energía renovable, entre otros:

	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

1. Rectificadores:

- Alimentación de todo tipo de sistemas electrónicos, donde se necesite energía eléctrica en forma de corriente continua.
- Control de motores de corriente continua utilizados en procesos industriales
- Cargadores de Baterías

2. Reguladores de alterna

- Calentamiento por inducción
- Control de iluminación
- Control de velocidad de motores de inducción

3. Inversores

- Accionadores de motores de corriente alterna en todo tipo de aplicaciones industriales
- Convertidores de corriente continua a alterna para fuentes no convencionales, tales como la fotovoltaica o eólica
- SAI o UPS

4. Troceadores

- Alimentación y control de motores de corriente continua
- Alimentación de equipos electrónicos a partir de baterías o fuentes autónomas de corriente continua

Cabe resaltar una aplicación importante que tiene la electrónica de potencia en la industria y es en el diseño de variadores de frecuencia (convertidores AC – AC), que se usan principalmente para el control de velocidad de motores AC.

El producto de investigación se basó principalmente en el hardware in the loop en FPGA de los convertidores DC – DC principales que son: Convertidor Buck, Convertidor Boost,

	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

Convertidor Buck – Boost. A continuación, mencionaremos características principales de cada uno y algunas aplicaciones.

2.2 CONVERTIDOR BUCK

Este tipo de conversor desempeña la función de reducir (a un valor menor o igual que el voltaje de entrada) la tensión de salida. Puede funcionar como convertidor DC a DC o DC a AC dependiendo si el valor de la señal de referencia es constante o sinusoidal. La figura 1 muestra su conexión básica.

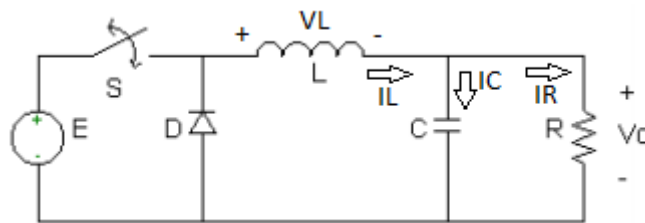


Figura 1. Convertidor Buck

Su conversión básicamente se da en dos operaciones, cuando el interruptor se encuentra cerrado o abierto. En el primer caso ($S = 1$), la corriente del inductor fluye a través de la carga y el condensador comienza a cargarse. Durante el intervalo en el que el interruptor está abierto ($S=0$), la corriente del inductor fluye a través del diodo, transfiriéndole energía a la carga.

El convertidor Buck es utilizado comúnmente en los ordenadores para reducir la tensión suministrada por la fuente de alimentación y así poder alimentar la CPU. Otras aplicaciones se dan en cargador de baterías y alimentación de periféricos como teclado y mouse en un dispositivo móvil.

2.3 CONVERTIDOR BOOST

Tiene como finalidad obtener a la salida un voltaje igual o mayor al de la entrada.

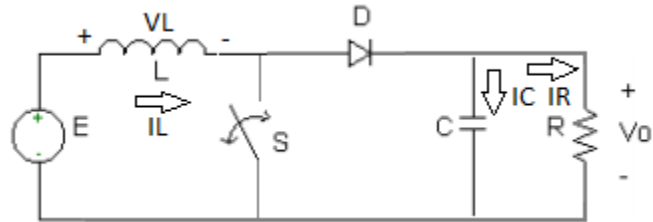


Figura 2. Convertidor Boost

Su principio básico de funcionamiento es el siguiente:

- Una fase de almacenamiento de energía, cuando el interruptor S está cerrado ($S=1$), lo que provoca un aumento de la corriente en el inductor, por lo tanto, el almacenamiento de una cantidad de energía en forma magnética. El diodo D está bloqueado y la carga es entonces desconectado de la fuente de alimentación.
- Cuando se abre el interruptor ($S=0$), la inductancia se encuentra entonces en serie con el generador y la FEM (fuerza electromotriz) se añade a la del generador. La corriente a través de la inductancia se hace pasar luego por el diodo D, el condensador C y la carga R. Esto da como resultado una transferencia de la energía acumulada en la inductancia a la carga.

Los convertidores Boost se utilizan en aplicaciones de baja potencia, tales como sistemas de iluminación portátiles. Un Led blanco requiere una tensión de 2.7 V a 3.6 V para operar, un convertidor aumenta la tensión suministrada por una batería de 1.5 V para poder suministrarle energía al Led.

En la industria Automotriz son utilizados para alimentar un motor eléctrico que requiere una tensión de 500 V. Sin un convertidor, el vehículo debe poseer 417 celdas de batería NiMH conectados en serie para alimentar el motor. Por medio del convertidor se puede lograr un voltaje de 500 V a la salida con una entrada de 202 V.

	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

2.4 CONVERTIDOR BUCK - BOOST

Este tipo de convertidor permite obtener a la salida un voltaje menor, igual o mayor al voltaje de entrada. A diferencia de los dos convertidores anteriores, a su salida se invierte la polaridad.

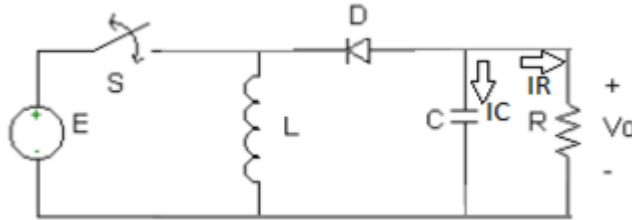


Figura 3. Convertidor Buck - Boost

Su funcionamiento básico en sencillo y es explicado a continuación:

- Durante el estado ON ($S=1$), la fuente de entrada de voltaje está directamente conectada al inductor. Por lo que se almacena la energía en L. En este paso, el condensador proporciona corriente a la carga de salida.
- Durante el estado OFF ($S=0$), el inductor está conectado a la carga de salida y el condensador, por lo que la energía es transferida del inductor al condensador y la carga

Es utilizado en aplicaciones como fuentes de alimentación de autorregulación, sistemas de energía de una batería, en control adaptativo, amplificador de potencia, entre otros.

Algo muy importante que interfiere en el valor de la salida para estos 3 tipos de convertidores DC a DC es el duty cycle (ciclo de trabajo), el cuales es la relación existente entre el tiempo en que una señal se encuentra en estado activo con el periodo de dicha señal, normalmente este término es aplicado para señales cuadradas, donde se intenta emular una señal analógica mediante la variabilidad del estado alto y bajo de una señal digital.

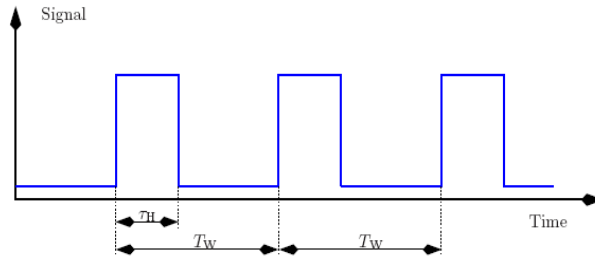


Figura 4. Señal PWM

En la imagen anterior, TH es el tiempo en estado activo y TW es el periodo de la señal, para calcular el duty cycle se aplica la siguiente formula. El 100 es simplemente para expresarlo en porcentajes.

$$D = 100\left(\frac{TH}{TW}\right)$$

La salida del PWM (onda cuadrada) es conectada a la entrada de excitación del dispositivo conmutador que se esté utilizando (generalmente mosfet), esta determina qué periodo de tiempo el interruptor debe estar abierto o cerrado, permitiendo que se genere a la salida el valor de voltaje deseado.

2.5 MODELO MATEMATICO CONVERTIDORES DC – DC

A la hora de empezar a modelar un convertidor se tienen en cuenta los dos estados que puede tener el dispositivo de switcheo, el cual puede ser abierto ($S=0$) o cerrado ($S=1$). Comenzaremos obteniendo las ecuaciones del circuito generado dependiendo de qué estado de conmutación se halle el interruptor. La finalidad del análisis de las dos etapas siempre será obtener las ecuaciones para el voltaje del inductor (V_L) y la corriente del capacitor (I_C). Para encontrar el modelo matemático final representando el voltaje de salida (voltaje en el condensador) y corriente de salida (corriente en el inductor), se realizarán unos pasos cortos y sencillos pero completos.

CONVERTIDOR BUCK

1. Obtener VL y IC cuando S=1 y S=0

- S=1 (CERRADO)

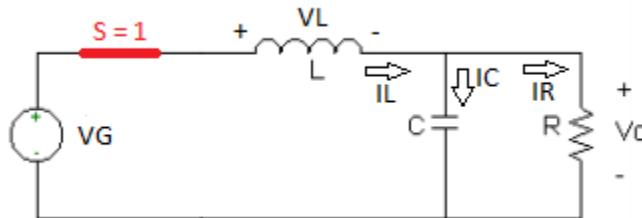


Figura 5. Conexión cuando S = 1

$$-VG + VL + VC = 0 \rightarrow VL = VG - VC$$

$$IL = IC + IR, \text{ pero } IR = \frac{V_o}{R} \rightarrow IC = IL - \frac{V_o}{R}$$

- S = 0 (ABIERTO)

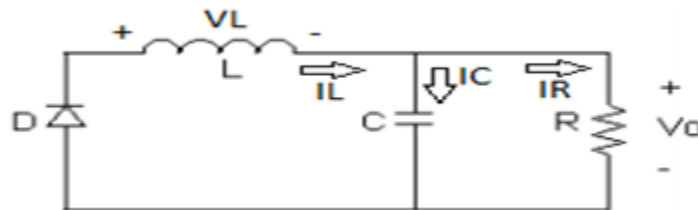


Figura 6. Conexión cuando S = 0

$$VL + VC = 0 \rightarrow VL = -VC$$

$$IL = IC + \frac{V_o}{R} \rightarrow IC = IL - \frac{V_o}{R}$$

	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

2. Realizar la gráfica VL vs Ts y IC vs Ts, donde Ts será el periodo de conmutación. Esto, con el fin de obtener el voltaje y la corriente de salida promedio.

- VL vs TS

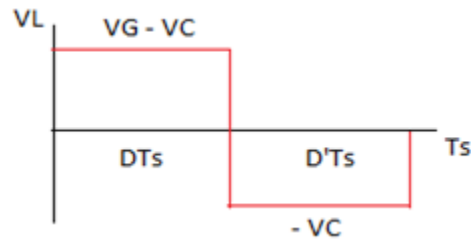


Figura 7. Grafico Voltaje promedio

La grafica se interpreta de la siguiente manera:

La ecuación de VL en el estado ON (S=1) es:

$$VL = VG - VC$$

Por esto, se representa en el momento DTs, donde D es el ciclo de trabajo.

Para el estado OFF (S=0), la ecuación de VL es: $VL = -VC$, representándose en el momento D'Ts, donde D' es la complementaria de D, entonces $D' = 1 - D$.

En el siguiente procedimiento matemático obtendremos el voltaje de salida promedio del convertidor, también conocido como M(D).

$$\langle VL \rangle = \frac{DTs * (VG - VC) + D'Ts(-VC)}{Ts}$$

$$\langle VL \rangle = D * VG - D * VC + (1 - D) * (-VC)$$

$$\langle VL \rangle = VG * D - D * VC + VC + D * VC$$

Un capacitor cargado no pide corriente y en un inductor cargado su voltaje es 0

$$\rightarrow 0 = D * VG - VC$$

	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

$$VC = VG * D$$

VC es el voltaje del condensador, por lo que será también el voltaje de salida y lo llamaremos Vo

$$\rightarrow Vo = VG * D$$

Finalmente, el M(D) del convertidor es:

$$M(D) = \frac{Vo}{VG} = D \text{ Ecuación (1)}$$

- Para obtener la corriente promedio IL, graficamos IC vs Ts.

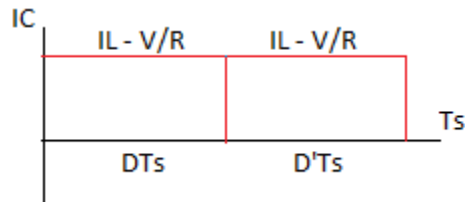


Figura 8. Grafico corriente promedio

$$\langle IC \rangle = \frac{\left(IL - \frac{Vo}{R} \right) * DTs + D'Ts * \left(IL - \frac{Vo}{R} \right)}{Ts}$$

$$\langle IC \rangle = \left(IL - \frac{Vo}{R} \right) * D + D' * \left(IL - \frac{Vo}{R} \right)$$

Un capacitor cargado no pide corriente y en un inductor cargado su voltaje es 0

$$0 = D * IL - \frac{Vo * D}{R} + D' * IL - \frac{D' * Vo}{R}$$

$$0 = D * IL - \frac{Vo * D}{R} + IL - D * IL - \left(\frac{Vo}{R} - \frac{Vo * D}{R} \right)$$

$$0 = IL - \frac{Vo}{R} \rightarrow IL = \frac{Vo}{R}, \text{ que es la corriente promedio de salida.}$$

	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

3. Llevar VL e IC en términos de derivadas, sabiendo que:

$$VL = L * \frac{dIL}{dt} \text{ y } IC = C * \frac{dVC}{dt}$$

- En el estado ON

$$\frac{dIL}{dt} = \frac{VG - Vo}{L}$$

$$\frac{dVC}{dt} = \frac{IL}{C} - \frac{Vo}{R * C}$$

- Para el estado OFF

$$\frac{dIL}{dt} = \frac{-Vo}{L}$$

$$\frac{dVC}{dt} = \frac{IL}{C} - \frac{Vo}{R * C}$$

4. Procedemos ahora a hallar el rizado del voltaje y la corriente de salida. Esto permitirá obtener los valores del inductor y el capacitor para poder obtener a la salida el voltaje deseado. Graficaremos ahora IC vs Ts y VL vs Ts teniendo en cuenta que la corriente en un inductor es triangular alrededor de un valor medido DC con un rizado.

- Para ΔIL (rizado corriente de salida)

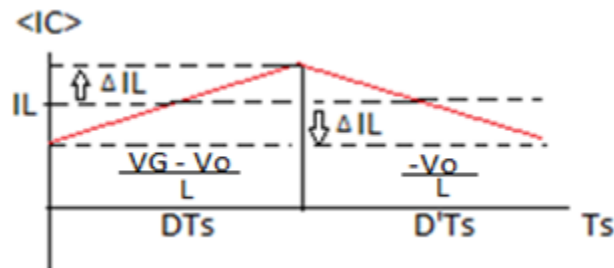


Figura 9. Grafico rizado corriente

Aplicamos la fórmula de la pendiente: $P = \frac{y2-y1}{x2-x1}$

	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

Podemos utilizar para el cálculo cualquiera de los dos estados, escogeremos el estado ON, en el cual, según la gráfica, la pendiente es: $P = \frac{VG-Vo}{L}$, el punto y_2 está dado por: $y_2 = IL + \Delta IL$ y el punto y_1 por: $y_1 = IL - \Delta IL$. En cuanto al eje X, el punto x_2 sería: $x_2 = DT_s$, mientras que x_1 es igual a 0.

Por lo tanto

$$\frac{VG - Vo}{L} = \frac{(IL + \Delta IL) - (IL - \Delta IL)}{DT_s}$$

$$\frac{VG - Vo}{L} = \frac{2 * \Delta IL}{DT_s}$$

$$\rightarrow \Delta IL = \left(\frac{VG-Vo}{2*L} \right) * DT_s \text{ Ecuación (2)}$$

- Para el cálculo del rizado en un convertidor Buck, se aplica la siguiente formula:

$$\Delta VC = \frac{\Delta IL * Ts}{8 * C} \text{ Ecuación (3)}$$

5. En este último paso encontraremos las ecuaciones del modelo final del convertidor. Para esto, se suman las ecuaciones obtenidas en el paso 3. La suma corresponde a la ecuación resultante en el estado ON más la obtenida en el estado OFF. Cada ecuación es multiplicada por D o D', dependiendo del estado en el que hayan resultado.

- $\frac{dIL}{dt}$

$$\frac{dIL}{dt} = \left(\frac{VG - Vo}{L} \right) * D - \left(\frac{Vo}{L} \right) * D'$$

$$\frac{dIL}{dt} = \frac{VG * D}{L} - \frac{Vo * D}{L} - \frac{Vo}{L} * (1 - D)$$

Finalmente:

$$\frac{dIL}{dt} = \frac{VG * D}{L} - \frac{Vo}{L} \text{ Ecuación (4)}$$

 Institución Universitaria	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

- $\frac{dV_C}{dt}$

$$\frac{dV_C}{dt} = \left(\frac{I_L}{C} - \frac{V_o}{R * C} \right) * D + \left(\frac{I_L}{C} - \frac{V_o}{R * C} \right) * D'$$

$$\frac{dV_C}{dt} = \left(\frac{I_L}{C} - \frac{V_o}{R * C} \right) * D + \left(\frac{I_L}{C} - \frac{V_o}{R * C} \right) * (1 - D)$$

Al operar obtenemos finalmente:

$$\frac{dV_C}{dt} = \frac{I_L}{C} - \frac{V_o}{R * C} \text{ Ecuación (5)}$$

CONVERTIDOR BOOST

Para este convertidor y el siguiente se simplificó el procedimiento. La manera de cómo se realiza es igual para cada paso, teniendo en cuenta que cada circuito es diferente, por lo que las ecuaciones iniciales cambian.

1. Obtener VL y IC cuando S=1 y S=0.

- S = 1

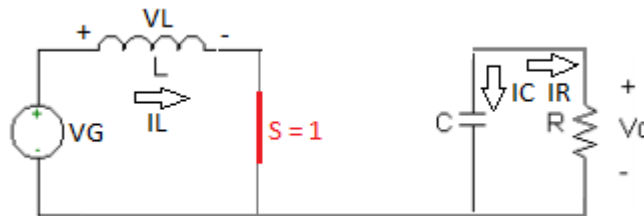


Figura 10. Conexión cuando S = 1

$$V_L = -V_G$$

$$I_C = -\frac{V_o}{R}$$

- S = 0

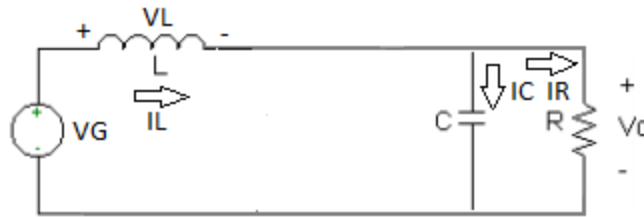


Figura 11. Conexión cuando S = 0

$$VL = VG - Vo$$

$$IC = IL - \frac{Vo}{R}$$

2. Realizar la gráfica VL vs Ts y IC vs Ts, donde Ts será el periodo de conmutación. Esto, con el fin de obtener el voltaje y la corriente de salida promedio.

- VL vs Ts

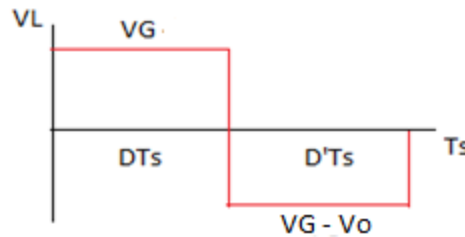


Figura 12. Grafico voltaje promedio

$$\langle VL \rangle = \frac{VG * D * Ts + (VG - Vo) * D' * Ts}{Ts}$$

$$\langle VL \rangle = VG * D * Ts + (VG - Vo) * D' * Ts$$

Un capacitor cargado no pide corriente y en un inductor cargado su voltaje es 0

$$0 = VG - Vo * D'$$

$$VG = Vo * D'$$

$$M(D) = \frac{Vo}{VG} = \frac{1}{1-D'} \rightarrow \text{Voltaje promedio, Ecuación (6)}$$

	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

- Para obtener la corriente promedio I_L , graficamos I_C vs T_s .

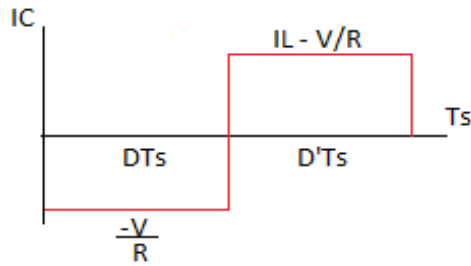


Figura 13. Grafico corriente promedio

$$\langle I_C \rangle = \frac{\left(\frac{-V_o}{R}\right) * DT_s + \left(I_L - \frac{V_o}{R}\right) * D'T_s}{T_s}$$

$$\langle I_C \rangle = \frac{-V_o * D}{R} + I_L * D' - \frac{V_o * D'}{R}$$

Un capacitor cargado no pide corriente y en un inductor cargado su voltaje es 0

$$0 = \frac{-V_o}{R} + I_L * D'$$

$$I_L = \frac{V_o}{R * D'} \rightarrow \text{Corriente promedio}$$

3. Llevar V_L e I_C en términos de derivadas, sabiendo que:

$$V_L = L * \frac{dI_L}{dt} \text{ y } I_C = C * \frac{dV_C}{dt}$$

- En el estado ON

$$\frac{dI_L}{dt} = \frac{V_G}{L}$$

$$\frac{dV_C}{dt} = -\frac{V_o}{R * C}$$

- Para el estado OFF

$$\frac{dI_L}{dt} = \frac{V_G - V_o}{L}$$

$$\frac{dVC}{dt} = \frac{IL}{C} - \frac{Vo}{R * C}$$

4. Procedemos ahora a hallar el rizado del voltaje y la corriente de salida. Esto permitirá obtener los valores del inductor y el capacitor para poder obtener a la salida el voltaje deseado. Graficaremos ahora IC vs Ts y VL vs Ts teniendo en cuenta que la corriente en un inductor es triangular alrededor de un valor medido DC con un rizado.

- Para ΔIL (rizado corriente de salida)

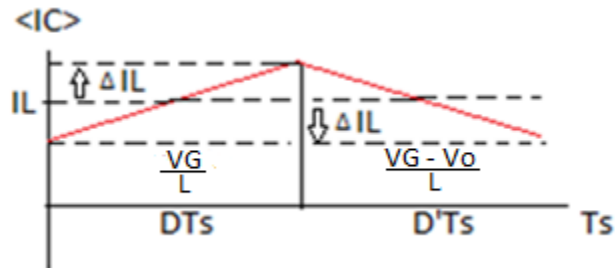


Figura 14. Grafico rizado corriente

$$\frac{VG}{L} = \frac{(IL + \Delta IL) - (IL - \Delta IL)}{DTs}$$

$$\frac{VG}{L} = \frac{2 * \Delta IL}{DTs}$$

$$\Delta IL = \frac{VG * D * TS}{2 * L} \text{ Ecuacion (7)}$$

- El mismo procedimiento se aplica para encontrar el rizado del voltaje de salida ΔVC

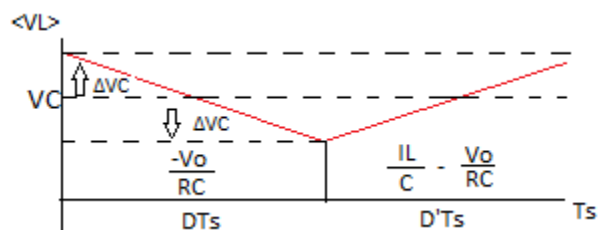


Figura 15. Grafico rizado de voltaje

 Institución Universitaria	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

$$\frac{-V_o}{RC} = \frac{(VC - \Delta VC) - (VC + \Delta VC)}{DT_s}$$

$$\frac{-V_o}{RC} = \frac{-2 * \Delta VC}{DT_s}$$

$$\Delta VC = \frac{V_o * D * T_s}{2 * R * C} \text{ Ecuación (8)}$$

5. En este último paso encontraremos las ecuaciones del modelo final del convertidor. Para esto, se suman las ecuaciones obtenidas en el paso 3. La suma corresponde a la ecuación resultante en el estado ON más la obtenida en el estado OFF. Cada ecuación es multiplicada por D o D', dependiendo del estado en el que hayan resultado

- $\frac{dIL}{dt}$

$$\frac{dIL}{dt} = \left(\frac{VG}{L}\right) * D + \left(\frac{VG - V_o}{L}\right) * D'$$

$$\frac{dIL}{dt} = \frac{VG}{L} - \frac{V_o * D'}{L} \text{ Ecuación (9)}$$

- $\frac{dVC}{dt}$

$$\frac{dVC}{dt} = \left(\frac{-V_o}{RC}\right) * D + \left(\frac{IL}{C} - \frac{V_o}{RC}\right) * D'$$

Al operar obtenemos finalmente:

$$\frac{dVC}{dt} = \frac{-V_o}{RC} + \frac{IL * D'}{C} \text{ Ecuación (10)}$$

CONVERTIDOR BUCK - BOOST

1. Obtener V_L y I_C cuando $S=1$ y $S=0$

- $S = 1$

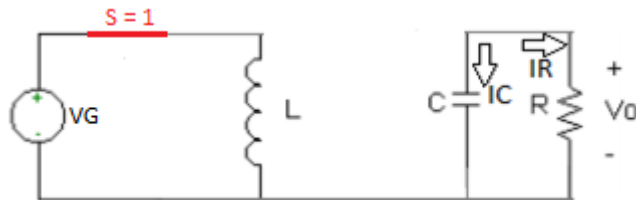


Figura 16. Conexión cuando $S = 1$

$$V_L = V_G$$

$$I_C = \frac{-V_o}{R}$$

- $S = 0$

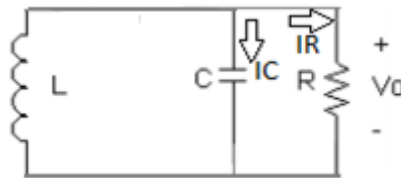


Figura 17. Conexión cuando $S = 0$

$$V_L = V_o$$

$$I_C = I_L - \frac{V_o}{R}$$

2. Realizar la gráfica V_L vs T_s y I_C vs T_s , donde T_s será el periodo de conmutación. Esto, con el fin de obtener el voltaje y la corriente de salida promedio

- VL vs TS

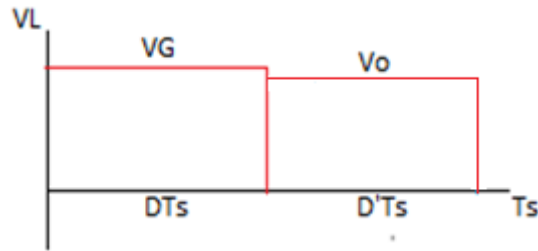


Figura 18. Grafico voltaje promedio

$$\langle VL \rangle = \frac{VG * D * Ts + Vo * D' * Ts}{Ts}$$

Un capacitor cargado no pide corriente y en un inductor cargado su voltaje es 0

$$0 = \frac{VG * D * Ts + Vo * D' * Ts}{Ts}$$

$$VG * D + Vo * D' = 0$$

Finalmente:

$$M(D) = \frac{Vo}{VG} = -\frac{D}{D'} \text{ Ecuación (11)}$$

- Para obtener la corriente promedio IL, graficamos IC vs Ts.

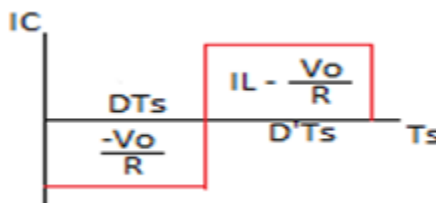


Figura 19. Grafico corriente promedio

$$\langle IC \rangle = \frac{\left(-\frac{Vo}{R}\right) * DTs + \left(IL - \frac{Vo}{R}\right) * D'Ts}{Ts}$$

Un capacitor cargado no pide corriente y en un inductor cargado su voltaje es 0

 Institución Universitaria	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

$$0 = \frac{\left(-\frac{V_o}{R}\right) * DT_s + \left(IL - \frac{V_o}{R}\right) * D'T_s}{T_s}$$

$$-\frac{V_o}{R} + IL * D' = 0$$

$$IL = \frac{V_o}{R * D'} \rightarrow \text{Corriente promedio}$$

3. Llevar VL e IC en términos de derivadas, sabiendo que:

$$VL = L * \frac{dIL}{dt} \text{ y } IC = C * \frac{dVC}{dt}$$

- En el estado ON

$$\frac{dIL}{dt} = \frac{VG}{L}$$

$$\frac{dVC}{dt} = -\frac{V_o}{R * C}$$

- Para el estado OFF

$$\frac{dIL}{dt} = \frac{V_o}{L}$$

$$\frac{dVC}{dt} = \frac{IL}{C} - \frac{V_o}{R * C}$$

4. Procedemos ahora a hallar el rizado del voltaje y la corriente de salida. Esto permitirá obtener los valores del inductor y el capacitor para poder obtener a la salida el voltaje deseado. Graficaremos ahora IC vs Ts y VL vs Ts teniendo en cuenta que la corriente en un inductor es triangular alrededor de un valor medido DC con un rizado.

- Para ΔIL (rizado corriente de salida)

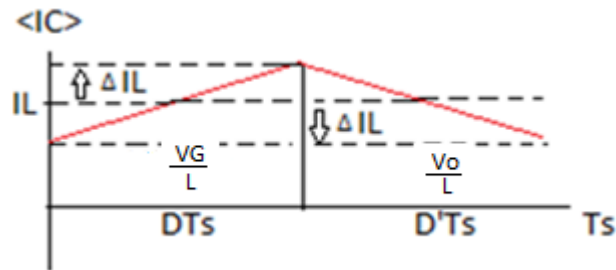


Figura 20. Grafico rizado corriente

$$\frac{VG}{L} = \frac{(IL + \Delta IL) - (IL - \Delta IL)}{DT_s}$$

$$\frac{VG}{L} = \frac{2 * \Delta IL}{DT_s}$$

$$\Delta IL = \frac{VG * D * T_s}{2 * L} \text{ Ecuacion (12)}$$

- El mismo procedimiento se aplica para encontrar el rizado del voltaje de salida ΔVC

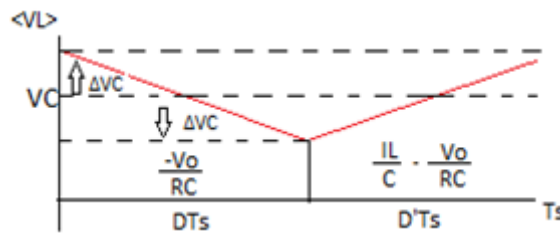


Figura 21. Grafico rizado voltaje

$$\frac{-Vo}{RC} = \frac{(VC - \Delta VC) + (VC + \Delta VC)}{DT_s}$$

$$\frac{-Vo}{RC} = \frac{-2 * \Delta VC}{DT_s}$$

$$\Delta VC = \frac{Vo * D * T_s}{2 * R * C} \text{ Ecuacion (13)}$$

5. En este último paso encontraremos las ecuaciones del modelo final del convertidor. Para esto, se suman las ecuaciones obtenidas en el paso 3. La suma corresponde a

 Institución Universitaria	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

la ecuación resultante en el estado ON más la obtenida en el estado OFF. Cada ecuación es multiplicada por D o D', dependiendo del estado en el que hayan resultado.

- $\frac{dIL}{dt}$

$$\frac{dIL}{dt} = \left(\frac{VG}{L}\right) * D + \left(\frac{Vo}{L}\right) * D'$$

$$\frac{dIL}{dt} = \frac{VG*D}{L} - \frac{Vo*D'}{L} \text{ Ecuacion (14)}$$

- $\frac{dVC}{dt}$

$$\frac{dVC}{dt} = \left(\frac{-Vo}{RC}\right) * D + \left(\frac{IL}{C} - \frac{Vo}{RC}\right) * D'$$

Al operar obtenemos finalmente:

$$\frac{dVC}{dt} = \frac{-Vo}{RC} + \frac{IL*D'}{C} \text{ Ecuacion (15)}$$

Para cada convertidor se obtuvieron las dos ecuaciones diferenciales que describen el comportamiento del voltaje y la corriente de salida. Este modelo matemático es el que por medio de VHDL y FPGA se realiza el Hardware in the Loop para cada planta.

2.6 VHDL

Es un lenguaje definido por el IEEE (*Institute of Electrical and Electronics Engineers*) (ANSI/IEEE 1076-1993) usado por ingenieros y científicos para la descripción de circuitos digitales que utiliza distintos niveles de abstracción. El significado de las siglas VHDL es VHSIC (Very High Speed Integrated Circuits) hardware Description Language. VHDL es un lenguaje de descripción de hardware, que permite describir circuitos síncronos y asíncronos.

El lenguaje de descripción de Hardware (HDL) es un lenguaje de programación especializado que sirve para definir la estructura, diseño y operación de circuitos electrónicos, y más

	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

comúnmente, de circuitos digitales. En muchas ocasiones no es denominado como un lenguaje de programación ya que posee unas notables diferencias como: Las directivas representan operaciones concurrentes y existen diferentes formas de escribir una función.

2.7 FPGA

Una FPGA (Field Programmable Gate Array) es un dispositivo programmable que contiene bloques de lógica cuya interconexión y funcionalidad puede ser configurada mediante un lenguaje de descripción especializado. Son capaces de configurarse para cualquier tipo de aplicación, especialmente en aplicaciones de circuitos digitales.

Tienen en su interior un conjunto muy grande de componentes digitales elementales combinatoriales y secuenciales, compuertas AND, OR, NOT, FLIP – FLOPS, entre otros. Como cualquier dispositivo posee limitantes en cuanto a frecuencia y cantidad de compuertas, donde estos dos factores son principalmente los que determinan el precio de la FPGA.

Algunos ejemplos de aplicaciones sencillas pueden ser: decodificadores 7 segmentos, temporizadores, sumadores, ALUs, microprocesadores, protocolos de comunicación UART, I2C, SPI, memorias RAM.

La FPGA utilizada en el producto de investigación fue la NEXYS 4 de Digilent. A continuación, se nombran algunas de sus características principales:

- Periféricos incorporados como: Acelerómetro, sensor de temperatura, micrófono digital, amplificador de altavoz. Conexión USB y Ethernet
- 16 switches (se utilizan como entradas)
- 16 leds (empleados como salidas)
- 4 displays 7 segmentos
- 4 Puertos PMOD, puerto donde se conecta el conversor D/A que permitirá visualizar la señal análoga en el osciloscopio.
- Frecuencia de trabajo de 100 MHz

	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22



Figura 22. FPGA Nexys 4

2.8 HARDWARE IN THE LOOP (HIL)

Es un tipo de simulacion en tiempo real. Se utiliza la simulacion HIL para poder probar el diseño del proyecto que se este ejecutando, en muchas ocasiones puede ser un controlador. La simulacion HIL muestra como el controlador responde, en tiempo real, a estímulos virtuales realistas. Se puede implementar tambien para determinar si el modelo fisico del sistema (planta) es correcto. El hardware de desarrollo contiene el modelo que es capaz en tiempo real de implementar el controlador y la planta. El hardware del controlador contiene el software del controlador que se genera a partir del modelo del sistema. El procesador en tiempo real (hardware de destino) contiene codigo para el sistema fisico que resulta por medio del modelo de la planta. La figura 23 muestra una configuracion tipica de la simulacion HIL.

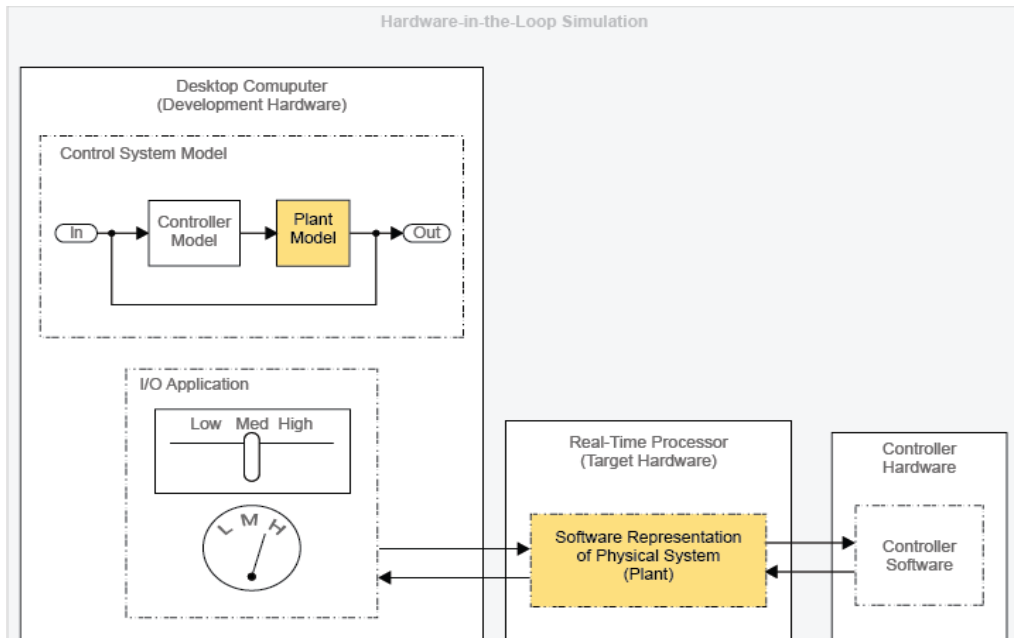


Figura 23. Configuración típica HIL

La simulación HIL se ha convertido en un tipo de desarrollo muy eficiente e importante porque permite visualizar la respuesta de una planta a través de un dispositivo de hardware, sin necesidad de realizar las conexiones físicas del sistema, en este caso, se implementa a través de código VHDL el modelo matemático de los tres convertidores ya descritos y se verifica su comportamiento mediante simulación HIL en la tarjeta de desarrollo Nexys 4 (FPGA).

El desarrollo mediante Hardware in the loop incluye dos factores importantes que manifiestan que su tipo de desarrollo y funcionalidad es eficiente y eficaz, estos son:

- **Costo:** Para su diseño, solo es necesario poseer un software y hardware de desarrollo. Cuando se está implementando algún sistema con conexiones físicas reales, pueden surgir inconvenientes como el daño de componentes. La simulación HIL permite el ahorro ante estos posibles percances.
- **Duración:** Todo sistema desarrollado se basa en un modelo matemático. Esto permite, que de una manera menos compleja su tiempo de diseño no sea extenso.

	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

Cualquier proyecto que se este ejecutando puede poseer cualquier tipo de inconvenientes, en especial fisicos. Gracias a la simulacion HIL, es mas facil evitar este tipo de problemas, permitiendo que el tiempo de ejecucion se prolongue menos.

En conclusión, la finalidad de la simulación HIL es poder realizar el diseño respectivo, probarlo y después poder llevar a cabo el proceso físico real de la planta o sistema a implementar.

	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

3 METODOLOGÍA

En el marco teórico, específicamente en el modelo de los convertidores DC a DC se obtuvieron las dos ecuaciones diferenciales que muestran el comportamiento del voltaje y corriente de salida. Estas dos ecuaciones fueron las que se introdujeron al código VHDL y que junto a otras variables y desarrollo se obtuvo el prototipo.

Algo importante que hay que destacar es la forma en la cual se desarrollaron las ecuaciones diferenciales, para esto se aplicó el método de Euler, el cual es un método numérico que permite ir incrementando paso a paso la variable independiente de la ecuación en diferencias a partir de un valor inicial dado, este valor se inicializa en cero y va aumentando a medida que se genere un flanco positivo del reloj.

3.1 ELEMENTOS DE VHDL

- **ESTRUCTURA**

La estructura que maneja VHDL consta principalmente de tres bloques: El primero son las librerías, las cuales almacenan los tipos de datos, operadores, componentes, objetos, funciones. Por otra parte, se halla la entidad (entity), que es el bloque que define y almacena las variables que se declaran como puertos de entradas y salidas. Finalmente se halla la arquitectura, que tiene como función describir el funcionamiento del circuito o diseño que se esté ejecutando. La figura 24 muestra la estructura.

 Institución Universitaria	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

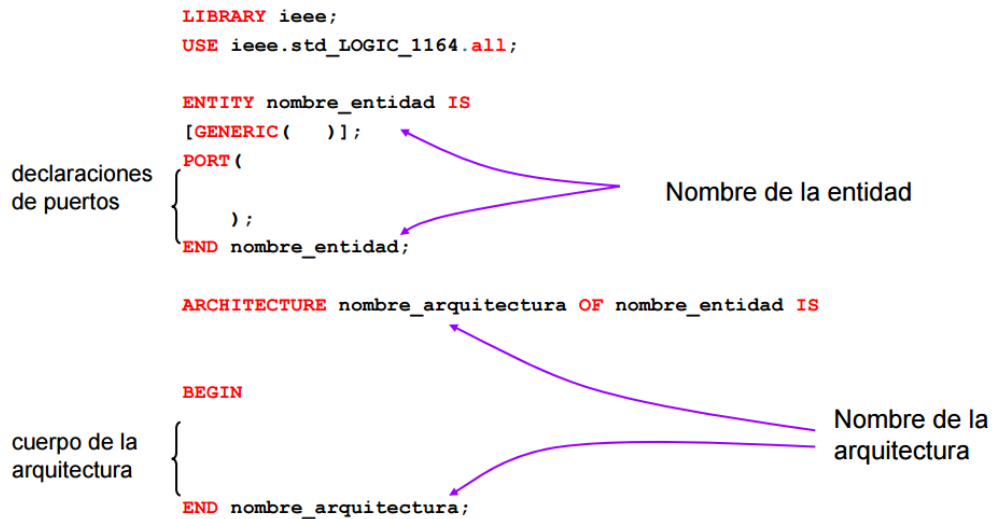


Figura 24. Estructura VHDL

- **LIBRERIAS UTILIZADAS**

Cada vez que creamos un proyecto nuevo por defecto se generan las siguientes librerías: *library IEEE, use IEEE.STD_LOGIC_1164.ALL*. Estas contienen definiciones de tipos, subtipos y funciones, podemos declarar variables de tipo *std_logic, std_logic_vector, rising_edge* y *falling_edge* para la detección de flancos de reloj.

La Librería *use IEEE.STD_LOGIC_ARITH.ALL* permite realizar operaciones matemáticas como la suma y la resta.

Las librerías *library IEEE_proposed, use IEEE_proposed.fixed_pkg.all*, son primordiales ya que permiten utilizar en el código variables de tipo punto flotante, donde, esta representación numérica permite almacenar un mayor rango de números que se puedan representar. Esto es primordial debido a que el cálculo de las ecuaciones diferenciales debe ser exacto. Su principal característica significativa se da por el hecho de que su síntesis es exitosa, permitiendo visualizar junto con el convertidor análogo – digital, la señal analógica en el osciloscopio.

- **SIGNAL Y PROCESS**

Las señales son variables que no son declaradas como puertos en la entidad y que se emplean dentro del algoritmo para llevar a cabo una función y donde interfieren en los

	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

variables tomadas como salidas. Se declaran como SIGNAL y se pueden declarar en las variables de tipo normal, register o bus. Es posible asignarles un valor inicial. Ejemplo:

SIGNAL sel: bit := '0';

SIGNAL datos: bit_vector (7 downto 0);

El process, es una sentencia concurrente en el sentido de que si en el algoritmo se declaran más de dos PROCESS estas sentencias se ejecutaran sin un orden establecido. No obstante, las sentencias que hay dentro del process se ejecutan de forma secuencial.

La estructura genérica es la siguiente:

```

PROCESS (lista de sensibilidad)
    (declaración de variables)

BEGIN
    (sentencias secuenciales)

END PROCESS;

```

En la lista de sensibilidad se agregan todas las variables que intervienen en el desarrollo del proceso e interfieren en los valores finales de salida.

3.2 HARDWARE Y SOFTWARE

Teniendo claro los conceptos de VHDL, junto con el modelo matemático, el siguiente proceso fue encontrar una tarjeta de desarrollo que tuviera las características principales que permitiera lograr el objetivo trazado. El hardware empleado fue la FPGA Nexys 4 de Digilent. Se describe a continuación cada propiedad con su función en el proyecto.

- Frecuencia de trabajo de 100 MHz: 100 millones de muestras por segundo es suficiente para el desarrollo del prototipo
- 16 switches y pulsadores: A través de ellos, se puede ingresar en forma binaria el valor perteneciente al ciclo de trabajo.
- Conector PMOD: Permite la conexión del conversor D/A. A su salida se obtiene el resultado de forma digital y este periférico se encarga de realizar su conversión a su valor respectivo análogo. Logrando también, visualizar la señal en el osciloscopio.

	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

Para dicha conversión, se utilizó el convertidor D/A PmodR2R. Sus características principales son:

- Conversor digital a analógico de 8 bits
- Frecuencia de conversión de datos de 25 MHz
- Fácil conexión con el osciloscopio para la visualización de la señal analógica
- Pequeño tamaño PCB flexible



Figura 25. Convertidor D/A PmodR2R

Para la programación en VHDL se empleó el software Vivado Design Suite, el cual es producido por Xilinx. Vivado nos permite sintetizar y realizar análisis de diseños HDL. Posee una interfaz sencilla pero completa, específicamente en la simulación donde podemos ver señales analógicas, a diferencia de software anteriores creados por su fabricante (Xilinx). La NEXYS 4 es compatible con este software.

3.3 CALCULO COMPONENTES DE LOS CONVERTIDORES

En la teoría, en el procedimiento del modelo matemático de cada convertidor, especialmente en el paso 4 se halló la fórmula para el cálculo del rizado en la salida de voltaje. Por medio de esta, se procedió a encontrar los componentes adecuados que garantizaran las siguientes especificaciones del diseño de cada convertidor, que son:

	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

- $V_{in} = 5 \text{ V}$
- $R \text{ (carga)} = 5 \Omega$

En el prototipo se darán como entrada 9 ciclos de trabajo. Para los cálculos se tomó el duty que dará como salida el voltaje mayor posible, que es de 0.9. La frecuencia a utilizar será de 100 MHz, donde $T_s = 10 \text{ ns}$.

1. Convertidor Buck

Con un duty de 0.9, en este convertidor se obtiene un voltaje máximo de 4.5 V.

- Calculo condensador:

Se asume un voltaje de rizado de 0.000000008. Mediante la ecuación (3), se despeja y se obtiene C

$$C = \frac{\Delta I L * T_s}{8 * \Delta V C}$$

Reemplazando:

$$C = \frac{0.000068 * 10 * 10^{-9}}{8 * 0.000000008}$$

$$C = 10 \text{ uF}$$

- Calculo bobina:

Corriente de rizado de 0.000068. A través de la ecuación (2), se despeja L:

$$L = \left(\frac{V_G - V_O}{2 * \Delta V C} \right) * D * T_s$$

$$L = \left(\frac{5 - 4.5}{2 * 0.000068} \right) * 0.9 * 10 * 10^{-9}$$

$$L = 330 \text{ uH}$$

	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

2. Convertidor Boost

Con un duty de 0.9, en este convertidor se obtiene un voltaje máximo de 50 V.

- Calculo condensador:

Se asume un voltaje de rizado de 0.0045 y mediante la ecuación (8), se despeja C, entonces:

$$C = \frac{V_o * D * T_s}{2 * R * \Delta V_C}$$

Reemplazando:

$$C = \frac{50 * 0.9 * 10 \times 10^{-9}}{2 * 5 * 0.0045}$$

$$C = 10 \mu f$$

- Calculo bobina:

Corriente de rizado de 0.000068. A través de la ecuación (7), se despeja L:

$$L = \frac{V_G * D * T_s}{2 * \Delta I_L}$$

$$L = \frac{5 * 0.9 * 10 \times 10^{-9}}{2 * 0.000068}$$

$$L = 330 \mu H$$

3. Convertidor Buck – Boost

Con un duty de 0.9, en este convertidor se obtiene un voltaje máximo de 45 V.

- Calculo condensador:

Se asume un voltaje de rizado de 0.0040 y mediante la ecuación (13), se despeja C, entonces:

	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

$$C = \frac{V_o * D * T_s}{2 * R * \Delta VC}$$

Reemplazando:

$$C = \frac{45 * 0.9 * 10 \times 10^{-9}}{2 * 5 * 0.0040}$$

$$C = 10 \text{ } \mu\text{f}$$

- Calculo bobina:

Corriente de rizado de 0.000068. A través de la ecuación (12), se despeja L:

$$L = \frac{VG * D * T_s}{2 * \Delta IL}$$

$$L = \frac{5 * 0.9 * 10 \times 10^{-9}}{2 * 0.000068}$$

$$L = 330 \text{ } \mu\text{H}$$

3.4 CODIGO VHDL DEL PROTOTIPO

A continuación, se explica el código VHDL implementado en el prototipo, se explica su procedimiento a partir de la estructura de VHDL, empezando por la declaración de las librerías hasta la escalización de la señal.

Declaración de librerías

```

1 library IEEE;
2 use IEEE.STD_LOGIC_1164.ALL;
3 use IEEE.STD_LOGIC_UNSIGNED.ALL;
4 use IEEE.STD_LOGIC_ARITH.ALL;
5
6 library IEEE_proposed;
7 use IEEE_proposed.fixed_pkg.all;

```

Figura 26. Declaración librerías

	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

En la figura 26, se muestran las librerías necesarias para le ejecución del código del proyecto. En la sección elementos de VHDL se explica su funcionalidad.

- Entidad

```

9 entity BUCK is
10     Port ( CLK : in STD_LOGIC;
11           RST : in STD_LOGIC;
12           D: in sfixed(1 downto -15);
13           Vo : out STD_LOGIC_VECTOR (7 downto 0) := (others => '0'));
14 end BUCK;
15

```

Figura 27. Declaración de puertos

Se tiene como puertos de entradas el CLK, un RST que vuelve a cero el voltaje de salida. El valor del duty es expresado en punto fijo, donde su valor en decimal, es representado en una combinación de 17 bits.

Una salida Vo previamente escalizada de 8 bits que será convertida a una señal análoga mediante el convertidor digital – analógico.

- Declaración de señales

```

16 architecture Behavioral of BUCK is
17
18 type estados is (e0, e1);
19 signal estado: estados;
20
21 signal x1n_fxp : sfixed(31 downto -32) := to_sfixed(0,31,-32);
22 signal x1n1_fxp : sfixed(31 downto -32) := to_sfixed(0,31,-32);
23 signal x2n_fxp : sfixed(31 downto -32) := to_sfixed(0,31,-32);
24 signal x2n1_fxp : sfixed(31 downto -32) := to_sfixed(0,31,-32);
25 signal Vin_fxp : sfixed(16 downto -16);
26 signal R_fxp : sfixed(31 downto -8);
27 signal L_fxp : sfixed(16 downto -16);
28 signal C_fxp : sfixed(1 downto -32);
29 signal D_fxp : sfixed(1 downto -15);
30 signal h_fxp : sfixed(1 downto -32);
31 signal clk_base : sfixed(1 downto -32);
32 signal n_fxp : sfixed(31 downto 0);
33 signal k, n : std_logic_vector(31 downto 0) := (others => '0');
34 signal clk_sys : std_logic := '0';
35 signal k1 : sfixed(16 downto 0);
36 signal k2 : sfixed(16 downto 0);
37 signal k3 : sfixed(16 downto 0);
--

```

Figura 28. Declaración de señales y estados

	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

Se declaran dos variables de tipo estado eo y $e1$, las cuales harán la transición entre el desarrollo de las ecuaciones diferenciales y la actualización del valor del voltaje y corriente por cada flanco de reloj. Implementado este proceso gracias al método de Euler explicado anteriormente

También, son declaradas las variables que almacenaran el voltaje y corriente de salida. De igual forma, se anexan las variables que contienen el valor de los componentes electrónicos y parámetros de cada convertidor. Se incluyen variables que permitirán obtener un divisor de frecuencia (reloj lento). Finalmente, se agregan las variables que permitirán obtener la señal de voltaje escalizada a 8 bits.

- Desarrollo arquitectura

Inicialización de variables: Dentro de este cuerpo, se ejecuta la inicialización de las variables a participar en el desarrollo de las ecuaciones diferenciales, divisor de frecuencia y escalización.

```

39 begin
40
41 Vin_fxp <= to_sfixed(5,16,-16);
42 R_fxp   <= to_sfixed(5,31,-8);
43 L_fxp   <= to_sfixed(0.000330,16,-16);
44 C_fxp   <= to_sfixed(0.000010,1,-32);
45 D_fxp   <= D;
46 h_fxp   <= to_sfixed(0.00000008,1,-32);
47 clk_base <= to_sfixed(0.00000004,1,-32);--
48 n_fxp   <= resize(h_fxp/clk_base, n_fxp);
49 n       <= to_slv(n_fxp);
50 k1     <= to_sfixed(21.6,16,0);
51 k2     <= to_sfixed(255,16,0);

```

Figura 29. Inicialización de variables

Divisor de frecuencia (reloj lento): Durante el desarrollo del prototipo fue necesario crear un reloj lento que convirtiera la frecuencia de trabajo de la FPGA a una frecuencia más baja, esto con el fin de poder visualizar la respuesta transitoria de cada convertidor en el osciloscopio.

	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

```

53 process (clk, k, n)
54 begin
55     if (rising_edge(CLK)) then
56         if (k < n-1 ) then
57             k <= k + '1';
58         else
59             clk_sys <= not(clk_sys);
60             k <= (others => '0');
61         end if;
62     end if;
63 end process;

```

Figura 30. Process reloj lento

Solución ecuaciones diferenciales: Es desarrollado dentro de un process que dentro de su lista de sensibilidad contendrá los parámetros de entrada y señales que intervendrán en el resultado de la salida, voltaje y corriente.

Como toda planta o sistema se incluye un Reset (RST), que lleva el voltaje y la corriente a cero cuando RST = 1 (Línea 67).

En la línea 73, mediante el **rising_edge** se detecta un flanco de subida del reloj lento creado anteriormente. Para la transición entre el desarrollo de las ecuaciones y la actualización del voltaje y corriente por cada flanco de reloj se producen dos estados. En el estado E0 se resuelve el modelo matemático obtenido, donde el voltaje se guarda en la señal **x2n1_fxp** y la corriente en **x1n1_fxp**. En el estado E1, a través de las señales **x1n_fxp** y **x2n_fxp** se almacenan los valores obtenidos en el estado E0, se suman en el desarrollo de las ecuaciones diferenciales, esto con ir aumentando el valor del voltaje y la corriente por cada flanco de reloj positivo y así iterativamente hasta finalmente llegar al resultado final,

 Institución Universitaria	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

dependiendo del duty ingresado. La figura 31 muestra el procedimiento descrito.

```

65 process (Vin_fxp, R_fxp, C_fxp, h_fxp, x1n_fxp, x1n1_fxp, x2n_fxp, x2n1_fxp, clk_sys)
66 begin
67   if (RST = '1') then
68     x1n_fxp <= to_sfixed(0,31,-32);
69     x2n_fxp <= to_sfixed(0,31,-32);
70     x1n1_fxp <= to_sfixed(0,31,-32);
71     x2n1_fxp <= to_sfixed(0,31,-32);
72   else
73     if (rising_edge(clk_sys)) then
74       case estado is
75         when e0 =>
76
77           x1n1_fxp <= resize( (((vin_fxp*d_fxp)/l_fxp)*h_fxp) - ((x2n_fxp*h_fxp)/l_fxp) + x1n1_fxp, x1n1_fxp);--Corriente Bobina
78           x2n1_fxp <= resize( ((x1n_fxp/c_fxp)*h_fxp) - ((x2n1_fxp*h_fxp)/(r_fxp*c_fxp)) + x2n_fxp , x2n1_fxp);--Voltaje Condensador
79
80           estado <= e1;
81
82         when e1 =>
83
84           x1n_fxp <= x1n1_fxp;
85           x2n_fxp <= x2n1_fxp;
86           estado <= e0;
87
88         end case;
89       end if;
90     end if;
91 end process;

```

Figura 31. Process, desarrollo ecuaciones diferenciales

Escalizacion: Por medio de estas líneas fue posible realizar la escalizacion y conversión de la señal de voltaje en punto flotante a binario de 8 bits. En la línea 93, se toma el valor representado en 8 bits que es 255 y mediante una regla de tres se halla el valor entre 0 y 255 que puede contener el voltaje de salida, teniendo en cuenta que la tensión máxima estará dada por el voltaje promedio (calculado en el modelo matemático). En la línea 94, la señal escalizada en punto flotante es convertida al binario de 8 bits para que pueda ser leída por el conversor digital – análogo.

```

93 k3 <= resize((x2n1_fxp*k2)/k1,k3);
94 Vo <= to_slv(k3(7 downto 0));
95
96 end Behavioral;

```

Figura 32. Escalizacion

	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

En el código y procedimiento descrito se da como ejemplo el diseño del convertidor Buck, para los dos faltantes, solo se hace necesario modificar las ecuaciones diferenciales obtenidas en el modelo matemático para cada uno. Para el convertidor Boost se agregan las ecuaciones 9 y 10, mientras que para el Buck – Boost la 14 y 15. El código completo de cada convertidor es anexado en el apéndice del informe.

4 RESULTADOS Y DISCUSIÓN

Se presentan los resultados del prototipo mediante la simulación en vivo del valor de voltaje obtenido por cada ciclo de trabajo para cada convertidor. El voltaje promedio mostrado en el osciloscopio pertenece a los nueve ciclos de trabajo en cada convertidor. Simulación en Matlab del voltaje promedio. Se realiza una comparación mediante el porcentaje error entre la simulación de Matlab y salida del osciloscopio del voltaje promedio. Finalmente, comparación de la respuesta transitoria resultante entre la adquirida por el osciloscopio y Matlab, esto para un duty de 0.5 para cada planta.

En la tabla 1 se muestra el valor de cada duty con su valor decimal y punto flotante representado en un binario de 17 bits. Esta combinación de bits es la que se ingresa por medio de switches a la FPGA y asigna su respectivo ciclo de trabajo.

Duty	Equivalente Binario
0.1	0000 11 00 11 00 11 00 1
0.2	000 11 00 11 00 11 00 11
0.3	00 1 00 11 00 11 00 11 00
0.4	00 11 00 11 00 11 00 11 00
0.5	1
0.6	01 00 11 00 11 00 11 00 1
0.7	01 0 11 00 11 00 11 00 11
0.8	011 00 11 00 11 00 11 00
0.9	111 11 00 11 00 11 0

Tabla 1. Dutys con su equivalente binario

4.1 CONVERTIDOR BUCK

Simulación Vivado (Voltaje promedio): Se muestra en la gráfica el voltaje de salida encerrado en un cuadro rojo. Es presentado también, el RST, Voltaje de entrada y el ciclo de trabajo.

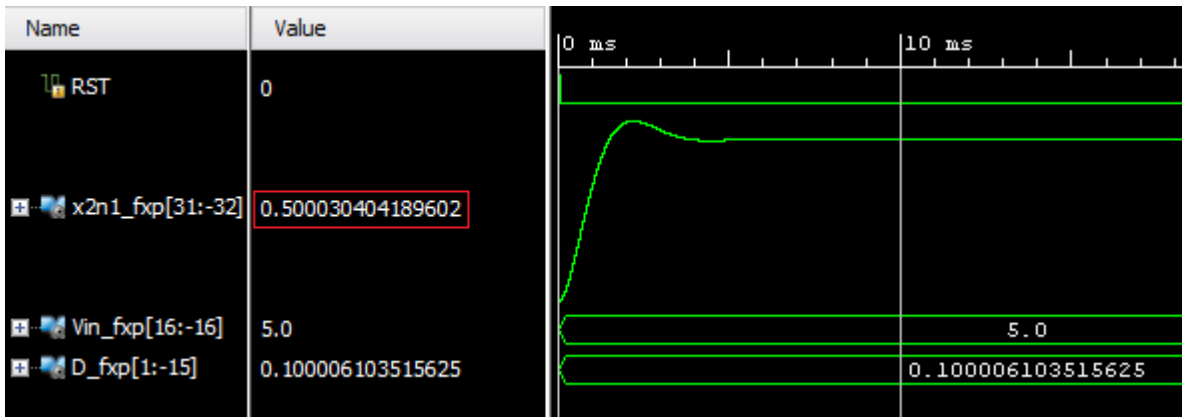


Figura 33. Duty = 0.1

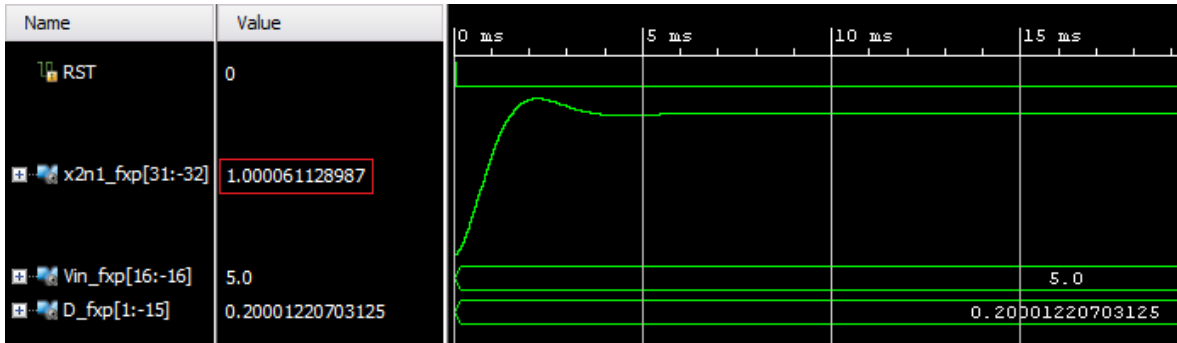


Figura 34. Duty = 0.2

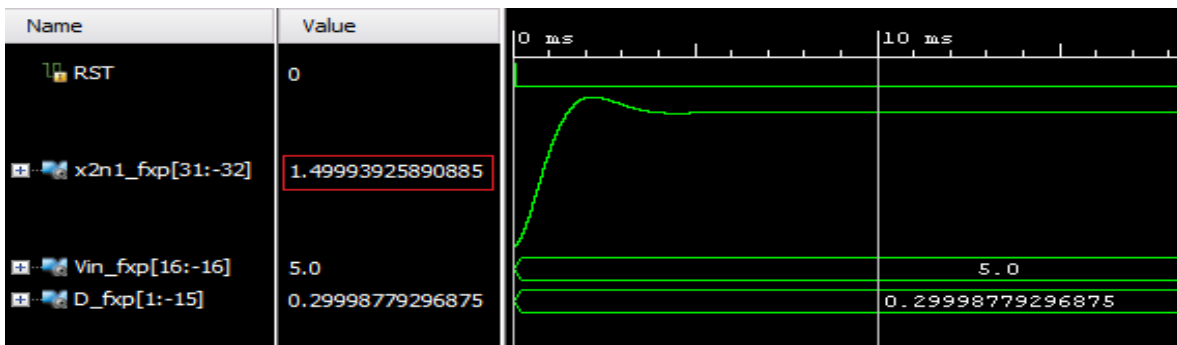


Figura 35. Duty = 0.3

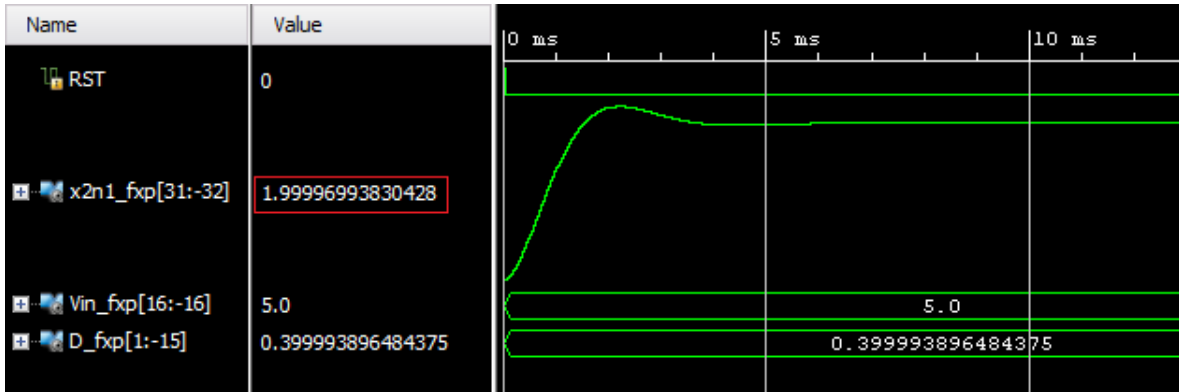


Figura 36. Duty = 0.4

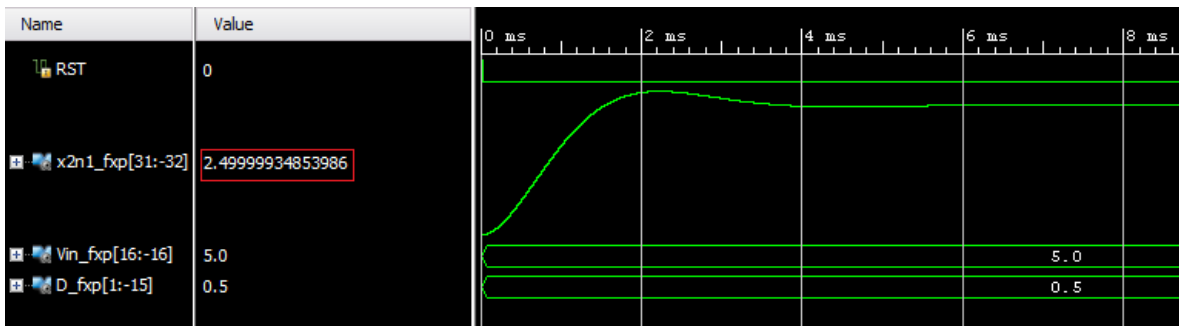


Figura 37. Duty = 0.5

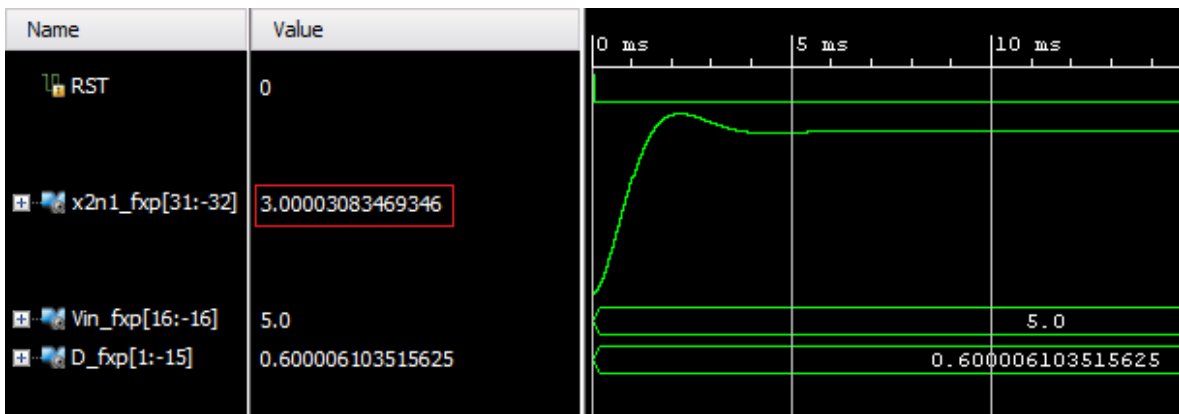


Figura 38. Duty = 0.6

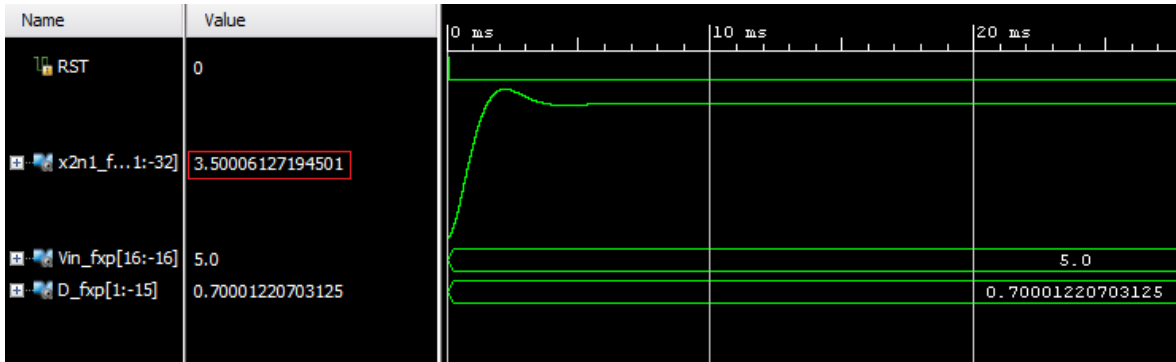


Figura 39. Duty = 0.7

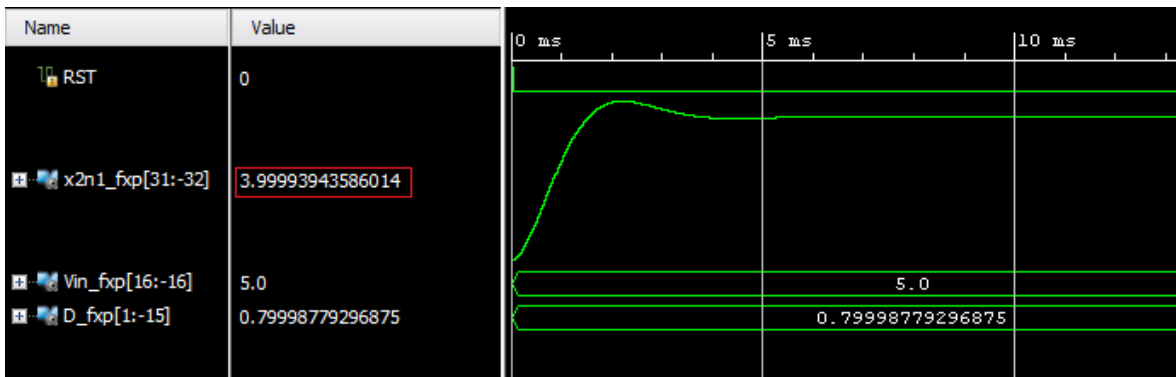


Figura 40. Duty = 0.8

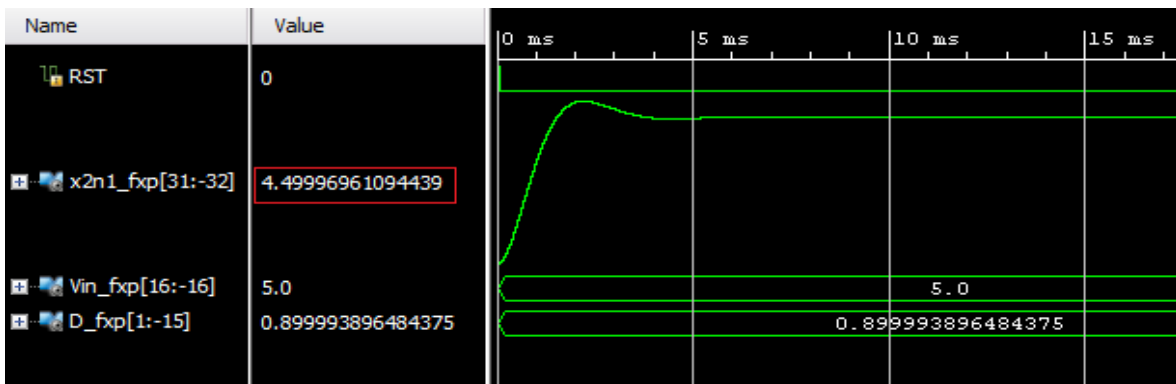


Figura 41. Duty = 0.9

Osciloscopio (voltaje promedio): Teniendo en cuenta que la salida máxima arrojada por la FPGA es 3.3 V y valor máximo del Buck es de 4.5 (duty de 0.9) es necesario realizar una escalización de la señal mediante la siguiente regla de tres:

$$V_f = \frac{V_o * 3.3}{4.5}$$

Donde V_f es el valor final escalizado y mostrado en el osciloscopio.

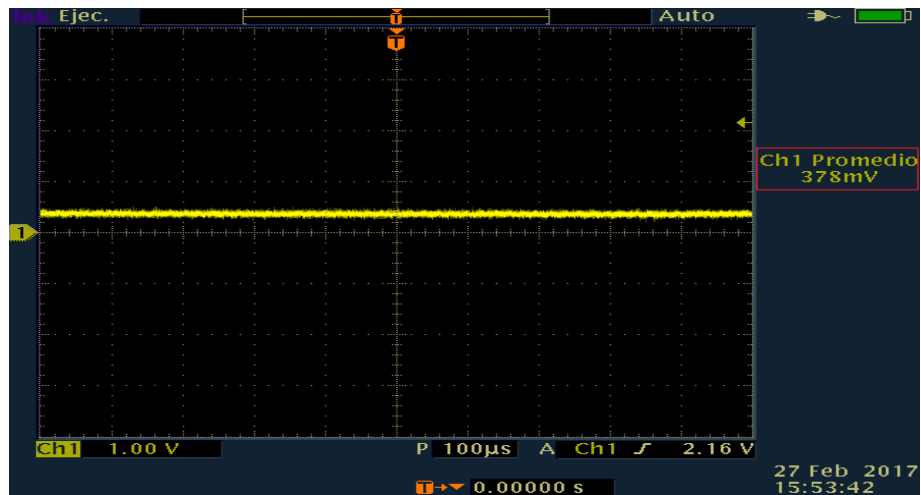


Figura 42. Duty = 0.1

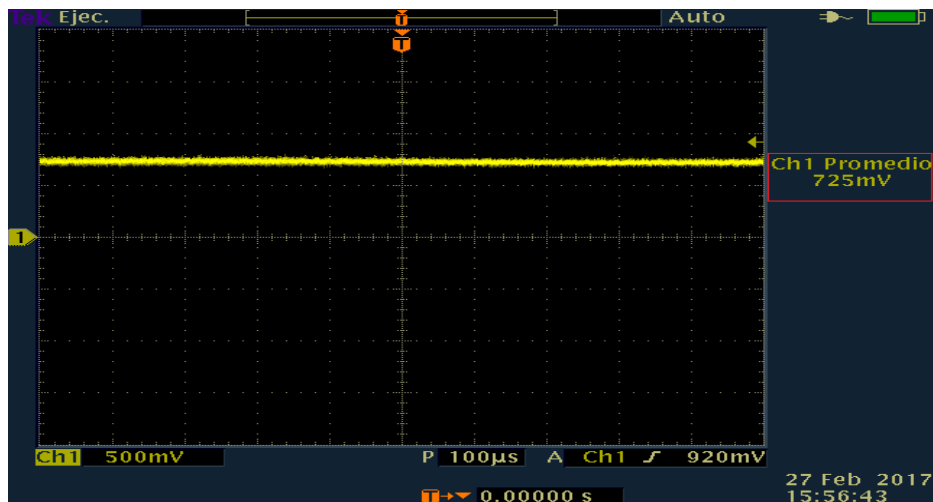


Figura 43. Duty = 0.2

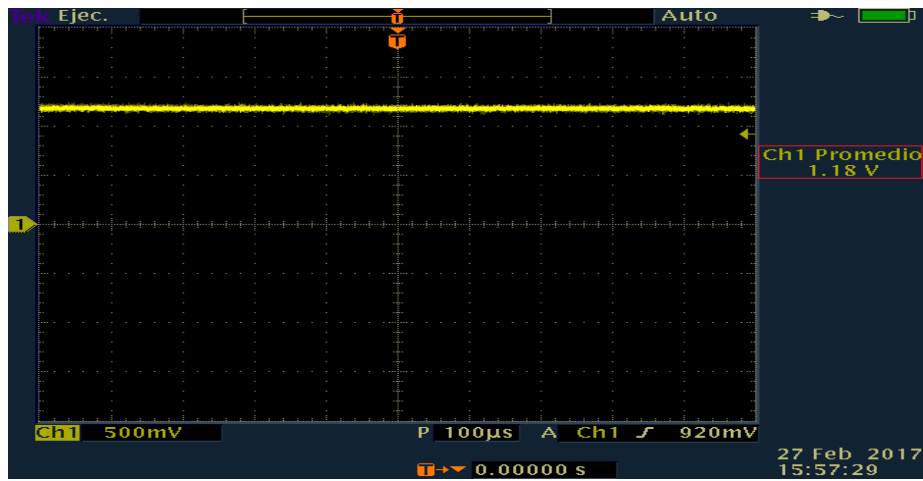


Figura 44. Duty = 0.3

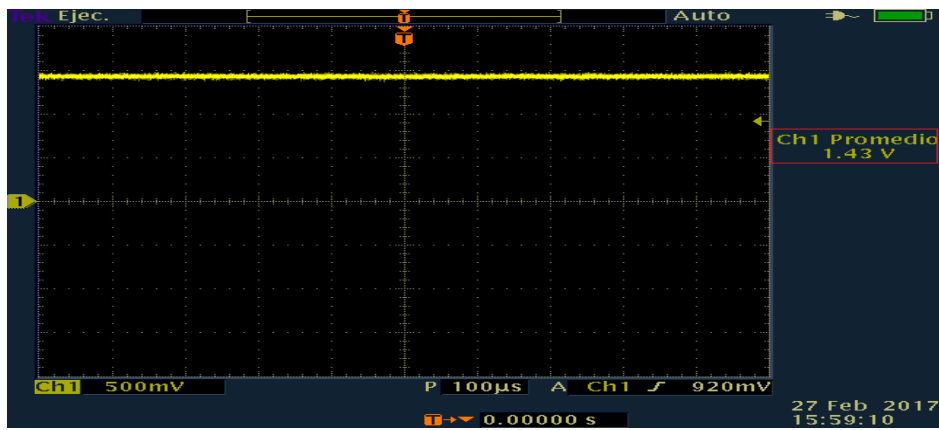


Figura 45. Duty = 0.4

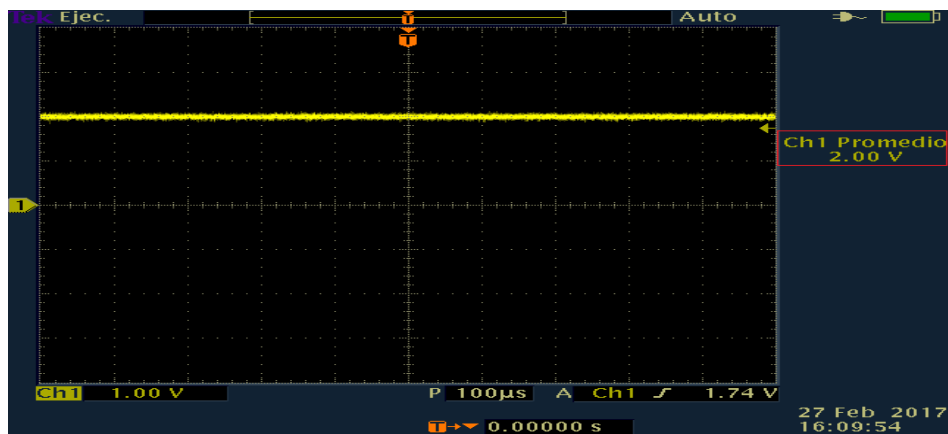


Figura 46. Duty = 0.5

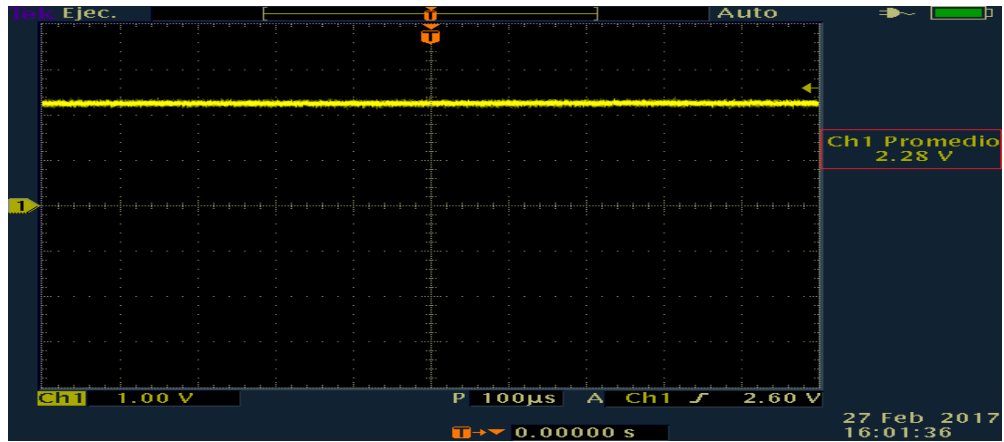


Figura 47. Duty = 0.6

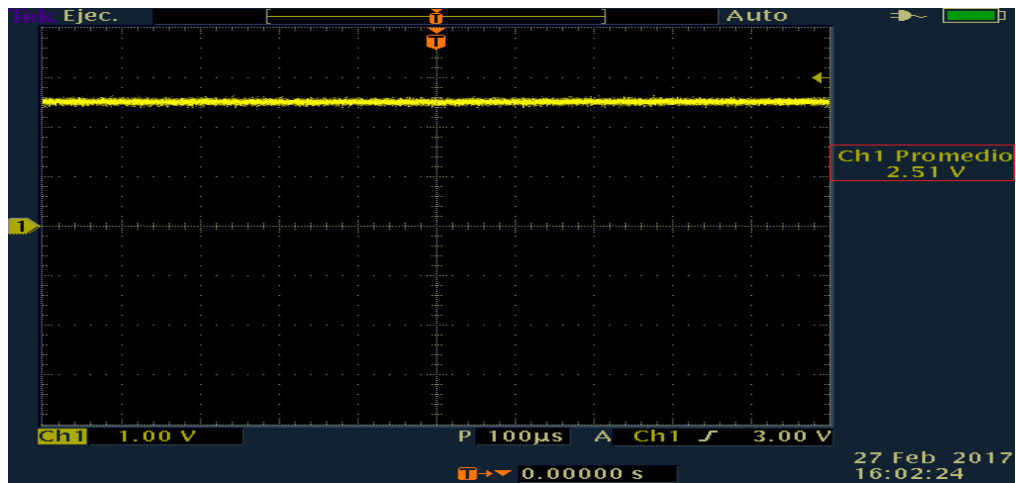


Figura 48. Duty = 0.7

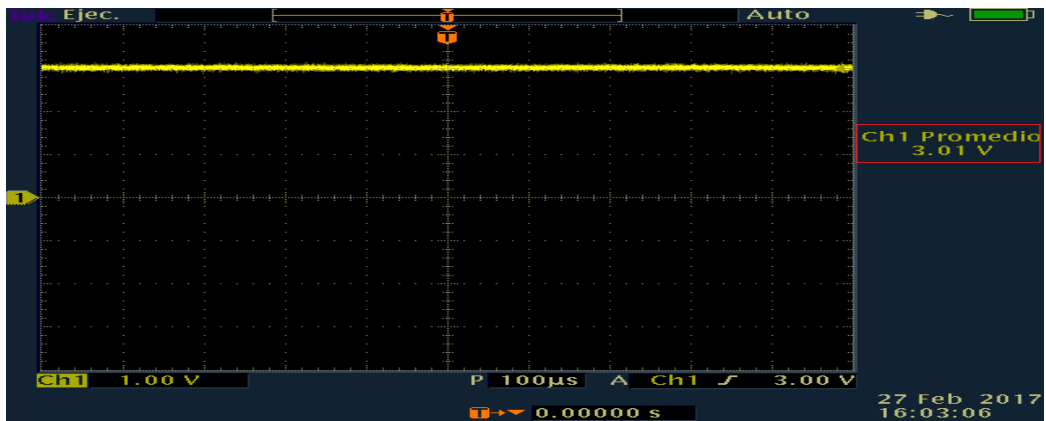


Figura 49. Duty = 0.8

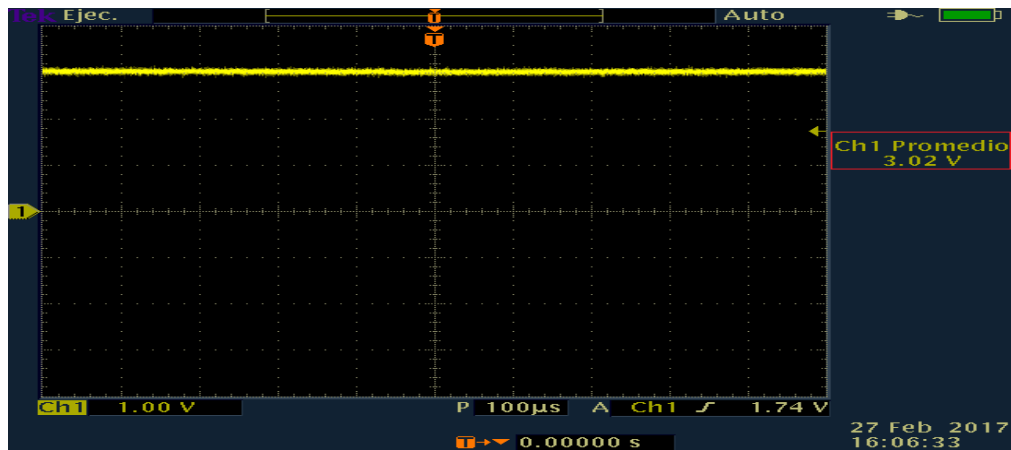


Figura 50. Duty = 0.9

Simulación Matlab (voltaje promedio):

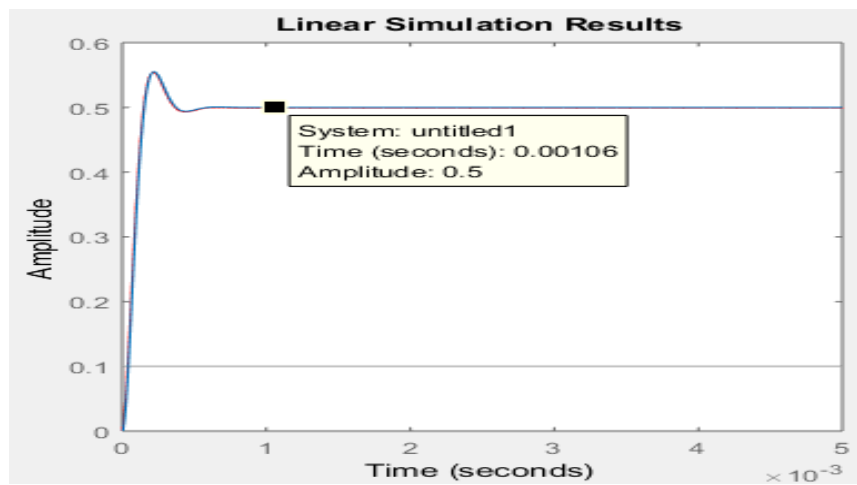


Figura 51. Duty = 0.1

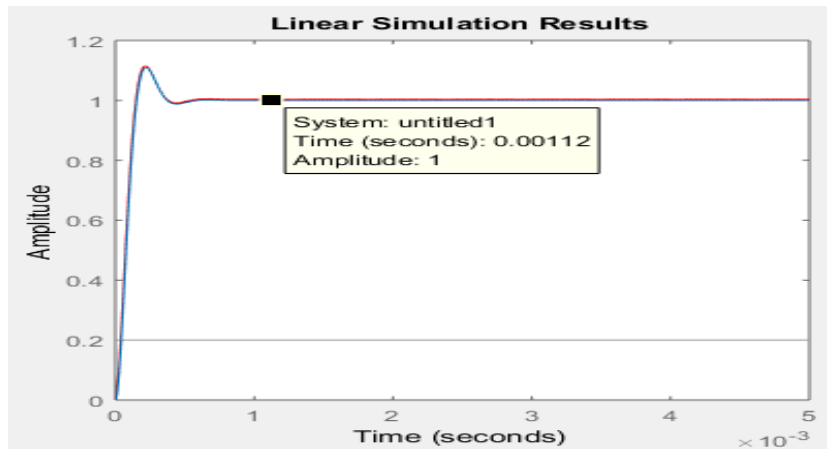


Figura 52. Duty = 0.2

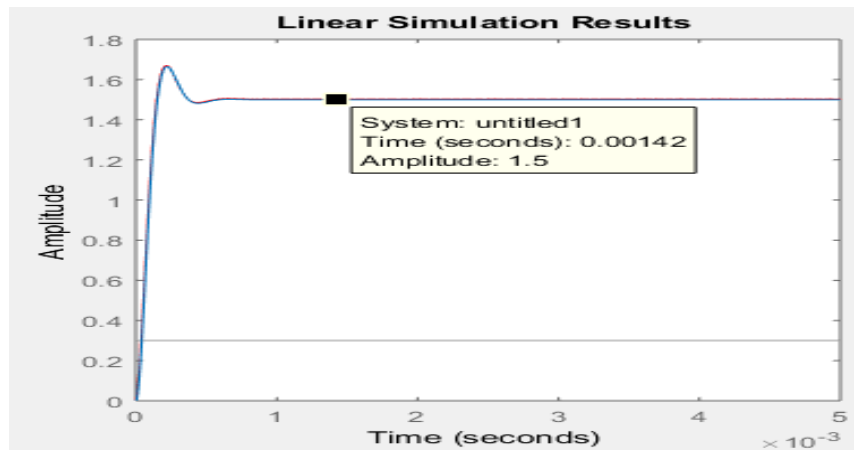


Figura 53. Duty = 0.3

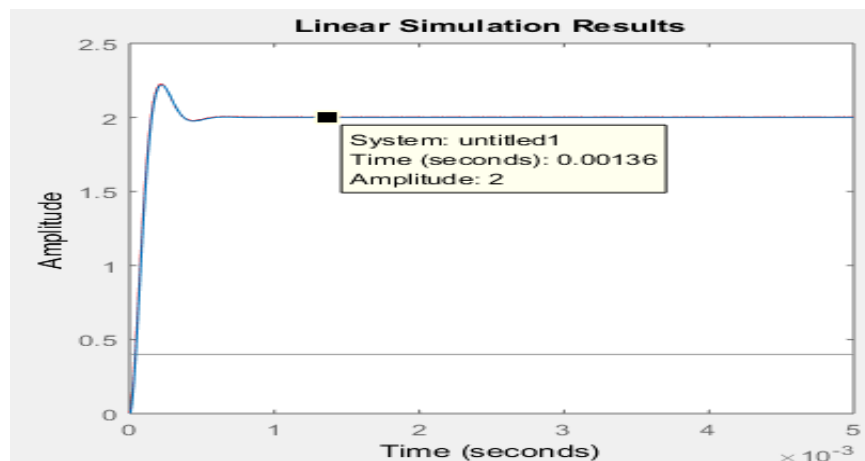


Figura 54. Duty = 0.4

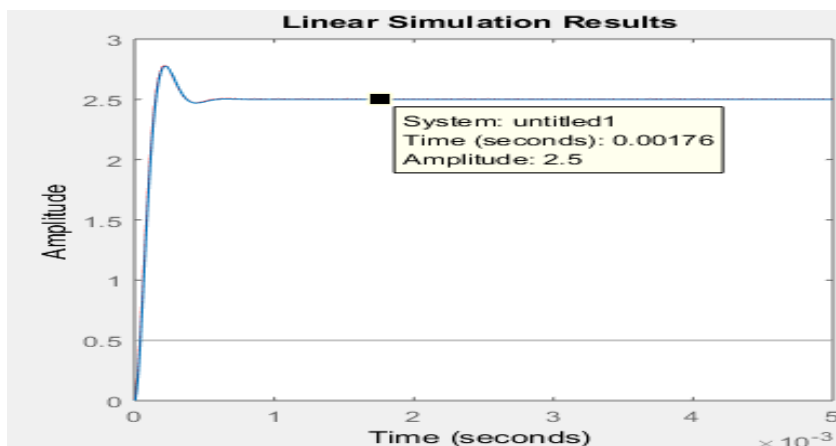


Figura 55. Duty = 0.5

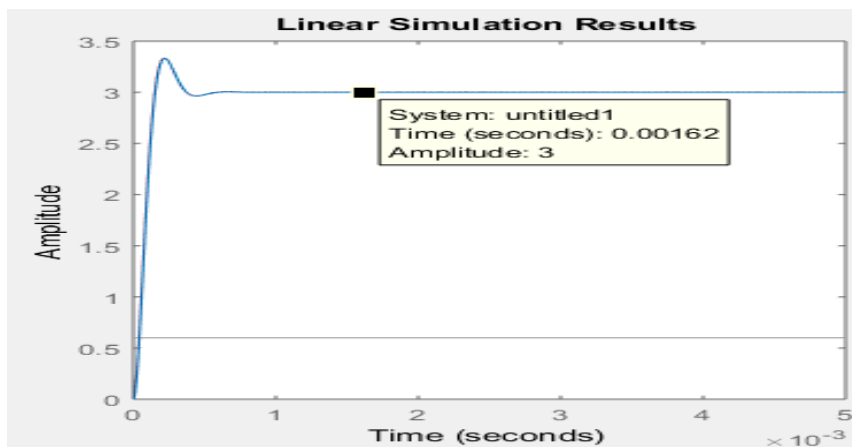


Figura 56. Duty = 0.6

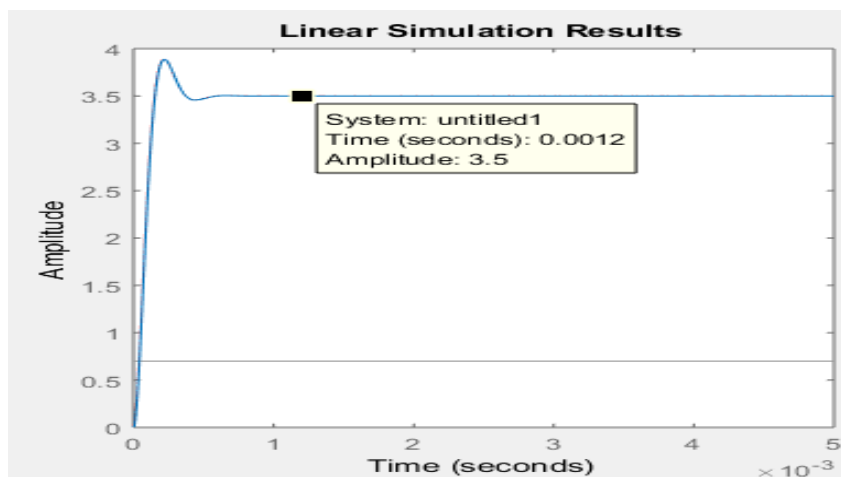


Figura 57. Duty = 0.7

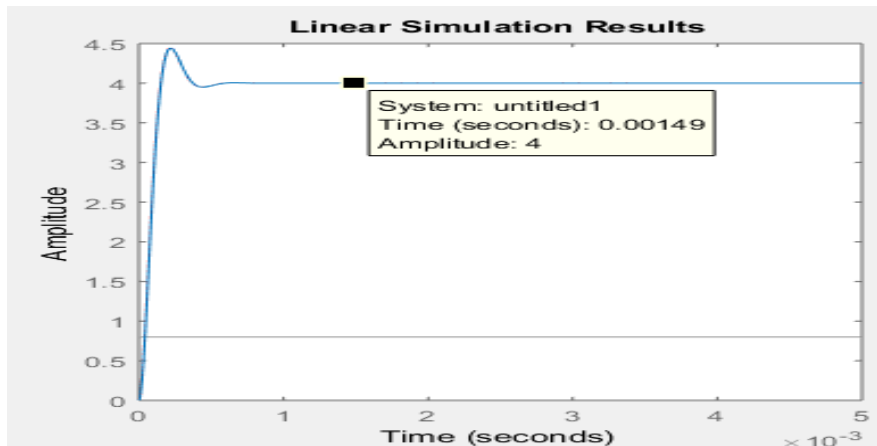


Figura 58. Duty = 0.8

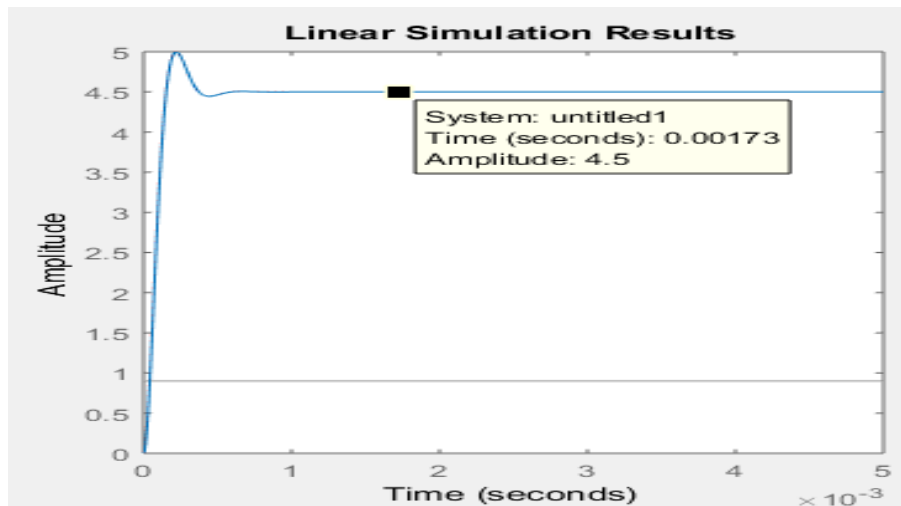


Figura 59. Duty = 0.9

Calculo porcentaje error: Se realiza la comparación de los resultados arrojados entre la Simulación en Matlab y el valor que arroja el osciloscopio del voltaje promedio de salida. El valor obtenido por el instrumento de medida es escalizado, por lo que, para poder hacer la comparación, es necesario que este valor se convierta al voltaje exacto que arroja el M (D) de cada convertidor, esto, mediante la siguiente formula.

$$V_o = \frac{V_f * 4.5}{3.3} \text{ Ecuación (16)}$$

Donde V_f es la señal escalizada

	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

El cálculo del porcentaje error está dado por la siguiente formula:

$$\% \text{ error} = \left| \frac{V_{\text{experimental}} - V_{\text{teorico}}}{V_{\text{teorico}}} \right| * 100$$

La tabla 2 muestra los valores promedio V_o de cada duty hallados, el % error y los valores obtenidos en las figuras 42 a 59.

Duty	Vf, señal escalizada (V)	V_o , Voltaje promedio, Ecuación 16 (V)	V_o , Simulacion Matlab (V)	% error (%)
0.1	0.378	0.51	0.5	2
0.2	0.725	0.98	1	2
0.3	1.18	1.60	1.5	6.66
0.4	1.43	1.95	2	2.5
0.5	2.00	2.72	2.5	8.8
0.6	2.28	3.10	3	3.33
0.7	2.51	3.42	3.5	2.28
0.8	3.01	4.10	4	2.5
0.9	3.02	4.11	4.5	8.66

Tabla 2. % error, Señal osciloscopio vs Simulacion Matlab

Respuesta transitoria: La figura 60, muestra la respuesta transitoria obtenida por Matlab y la hallada por el osciloscopio mediante las coordenadas en X y Y extraídas del medidor. La señal de color rojo es la experimental (osciloscopio) mientras que la azul la simulada por Matlab.

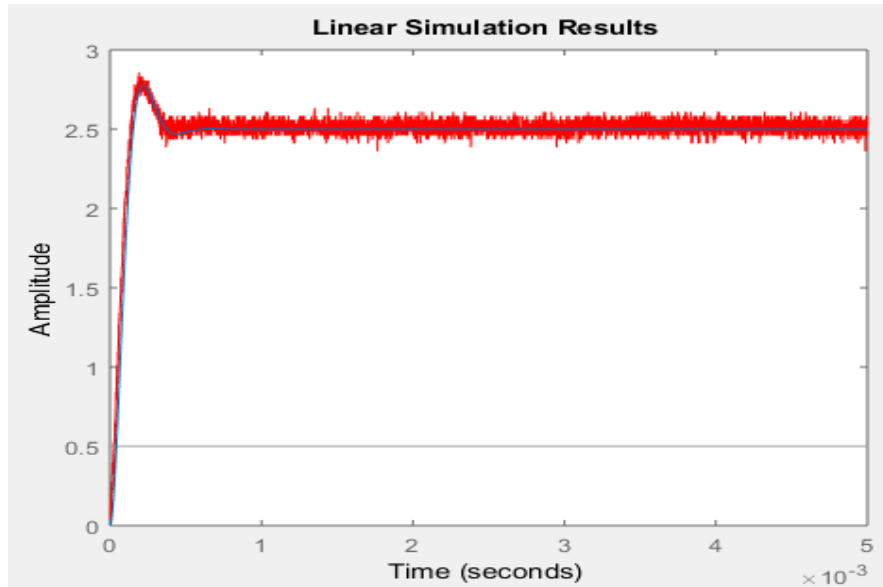


Figura 60. Respuesta transitoria, Matlab vs osciloscopio, Duty = 0.5

4.2 CONVERTIDOR BOOST

Simulación vivado:

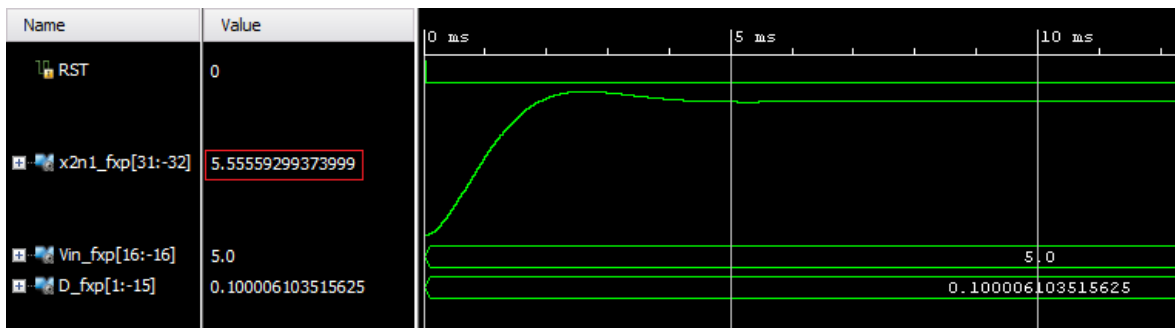


Figura 61. Duty = 0.1

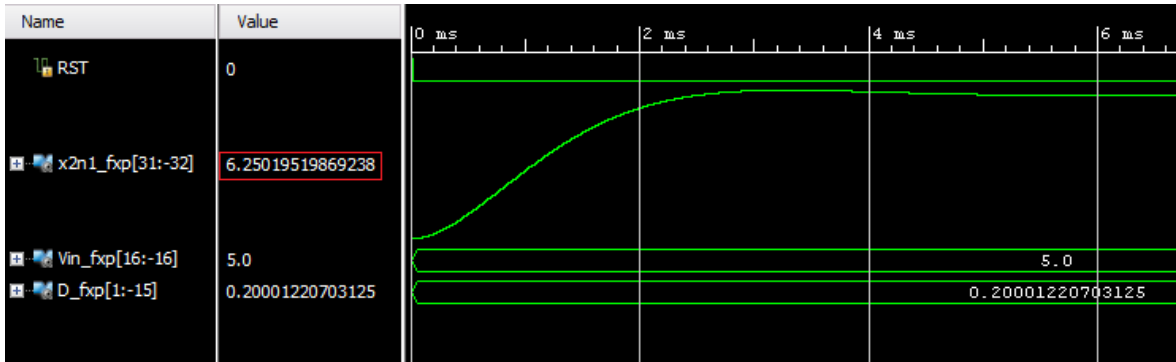


Figura 62. Duty = 0.2

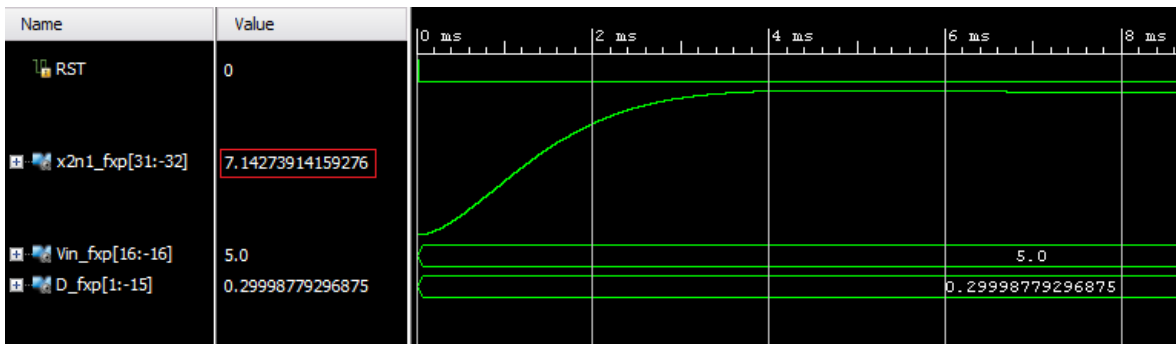


Figura 63. Duty = 0.3

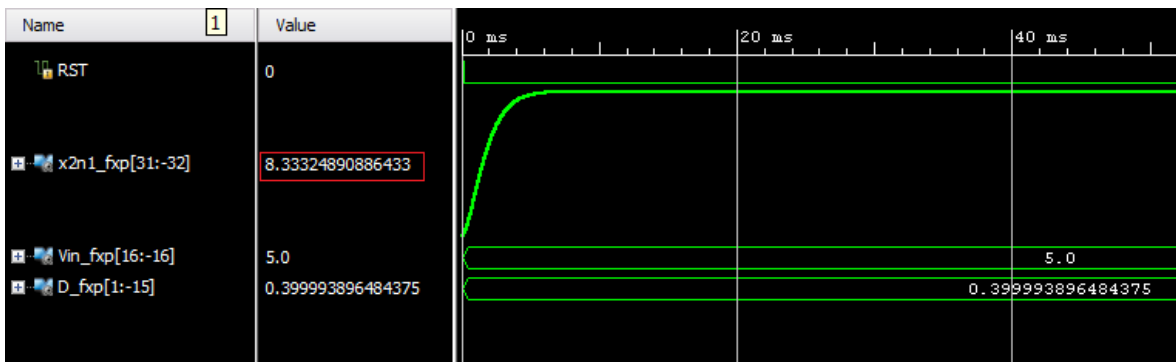


Figura 64. Duty = 0.4

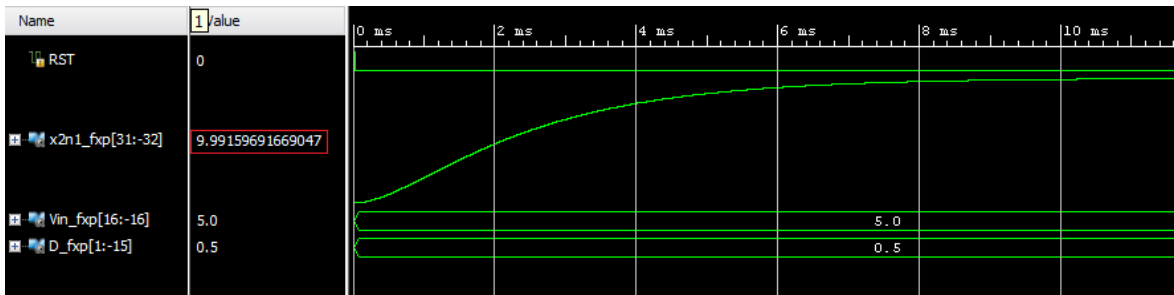


Figura 65. Duty = 0.5

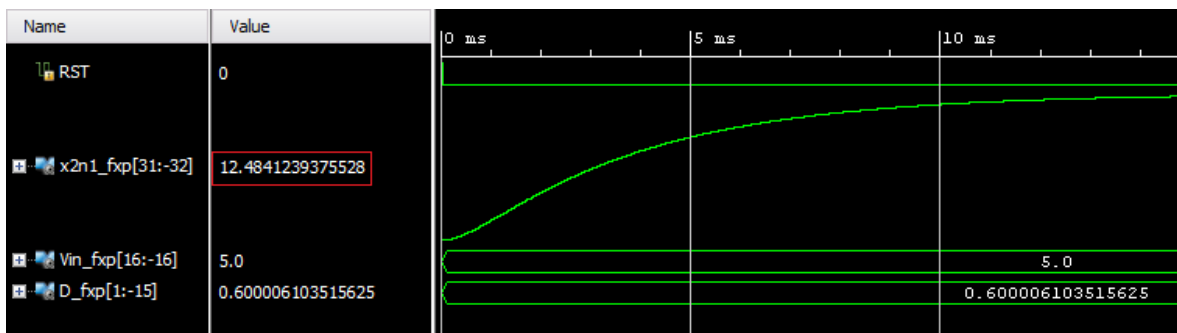


Figura 66. Duty = 0.6

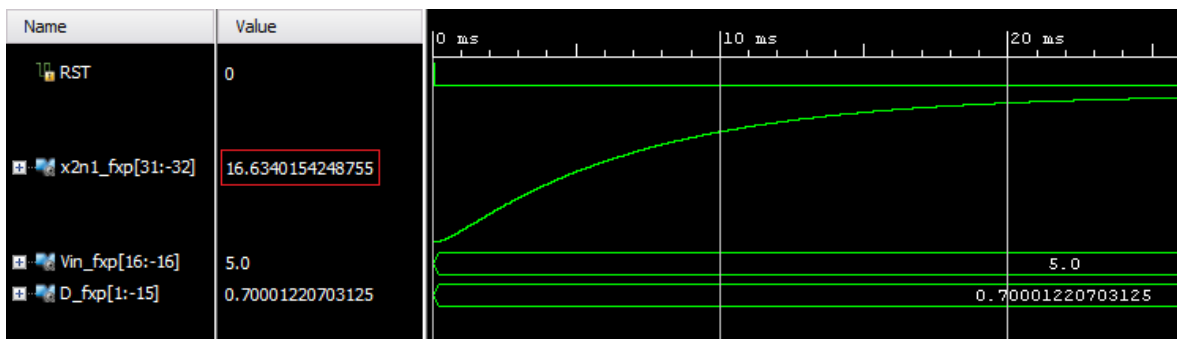


Figura 67. Duty = 0.7

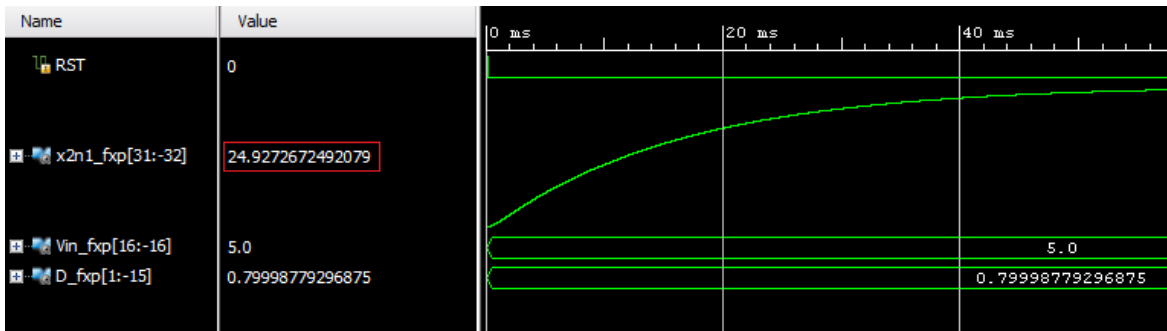


Figura 68. Duty = 0.8

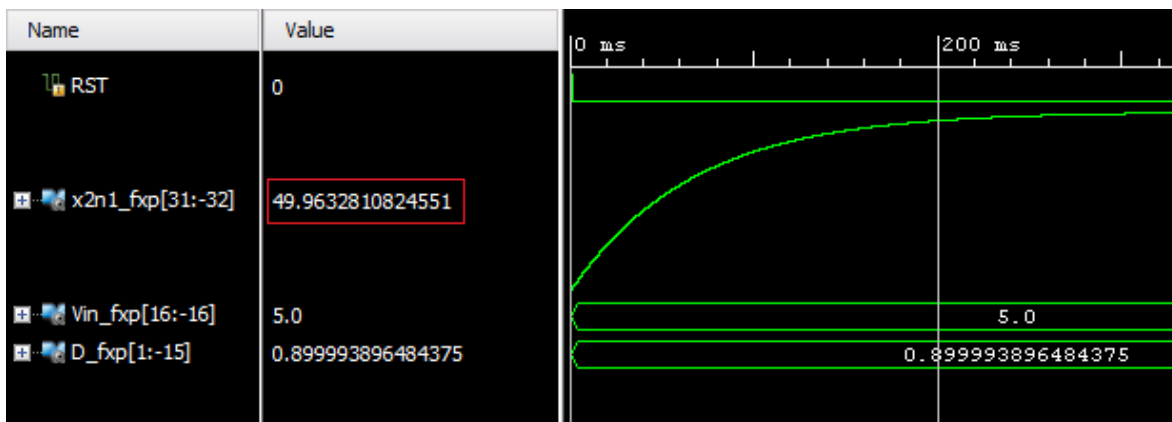


Figura 69. Duty = 0.9

Osciloscopio (voltaje promedio): Teniendo en cuenta que la salida máxima arrojada por la FPGA es 3.3 V y valor máximo del Boost es de 50 (duty de 0.9) es necesario realizar una escalización de la señal mediante la siguiente regla de tres:

$$V_f = \frac{V_o * 3.3}{50}$$

Donde V_f es el valor final escalizado y mostrado en el osciloscopio.

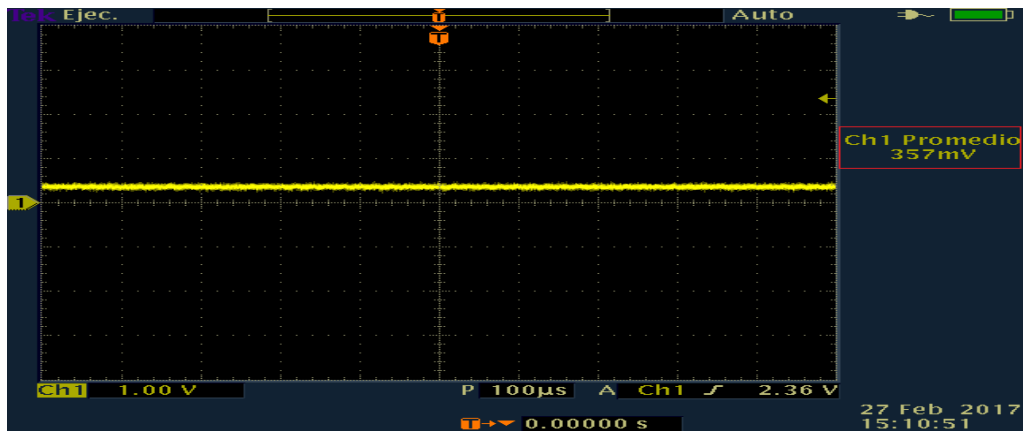


Figura 70. Duty = 0.1

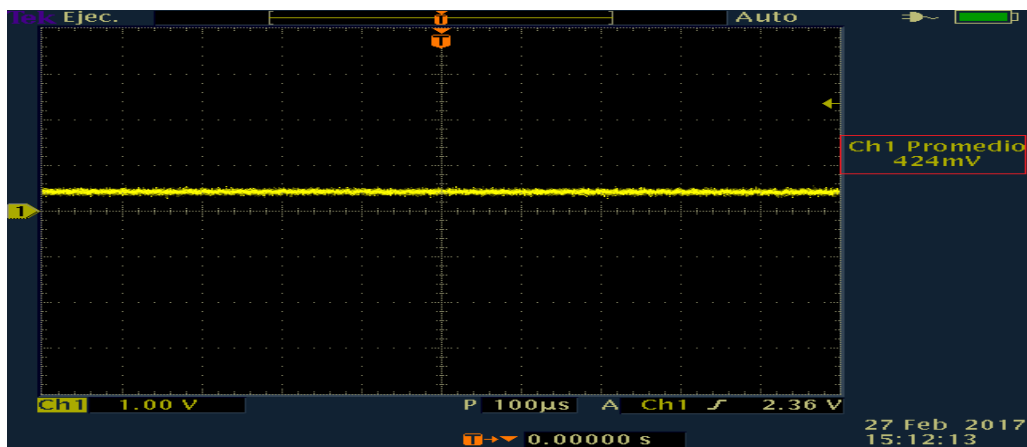


Figura 71. Duty = 0.2

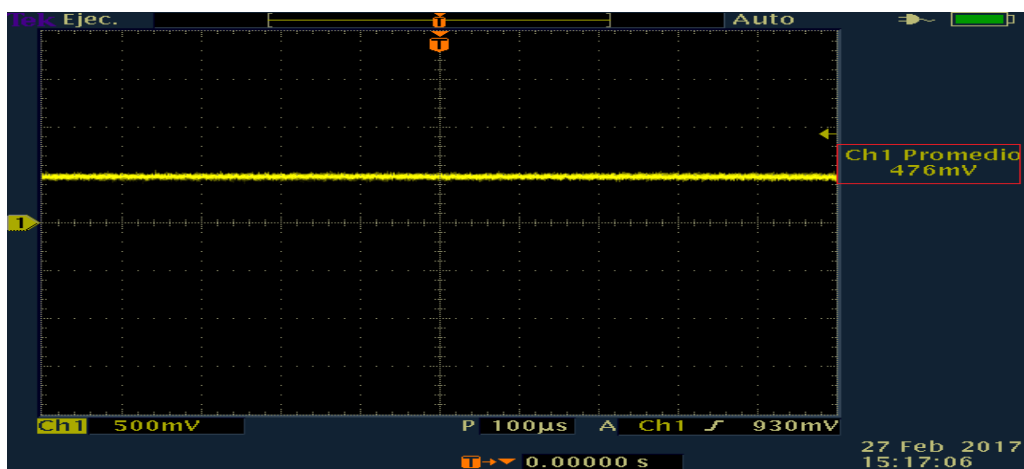


Figura 72. Duty = 0.3

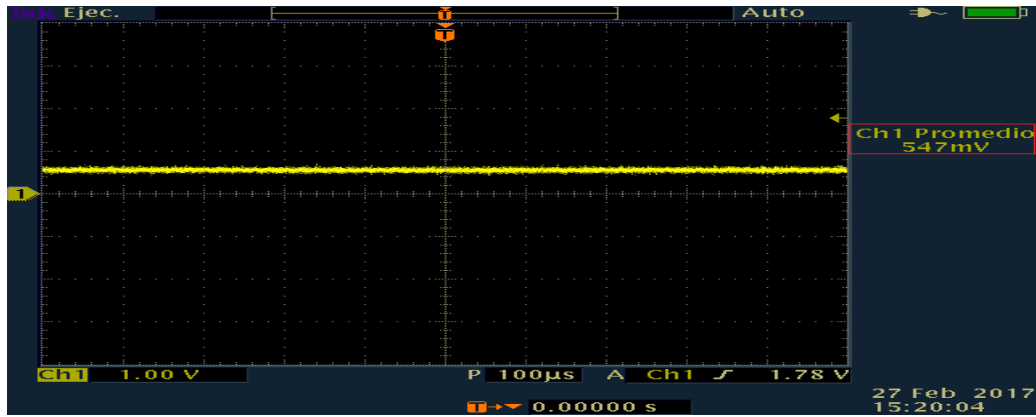


Figura 73. Duty = 0.4

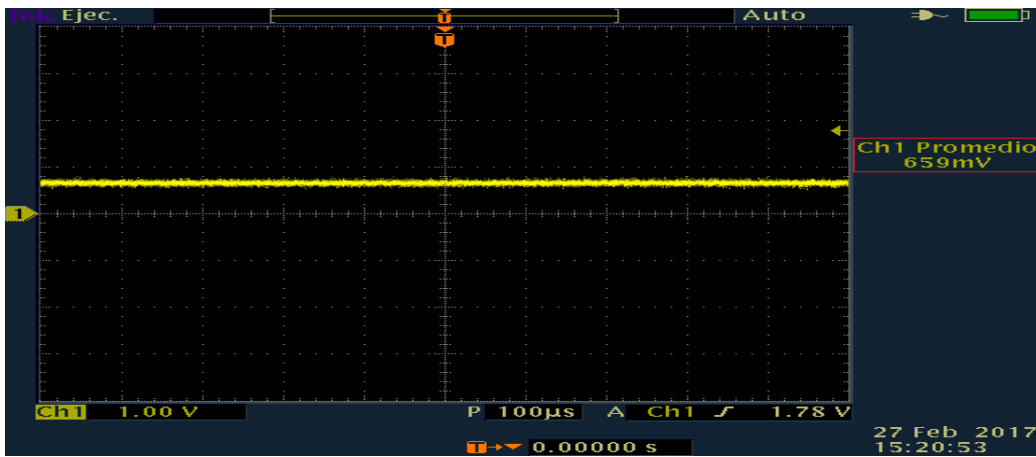


Figura 74. Duty = 0.5

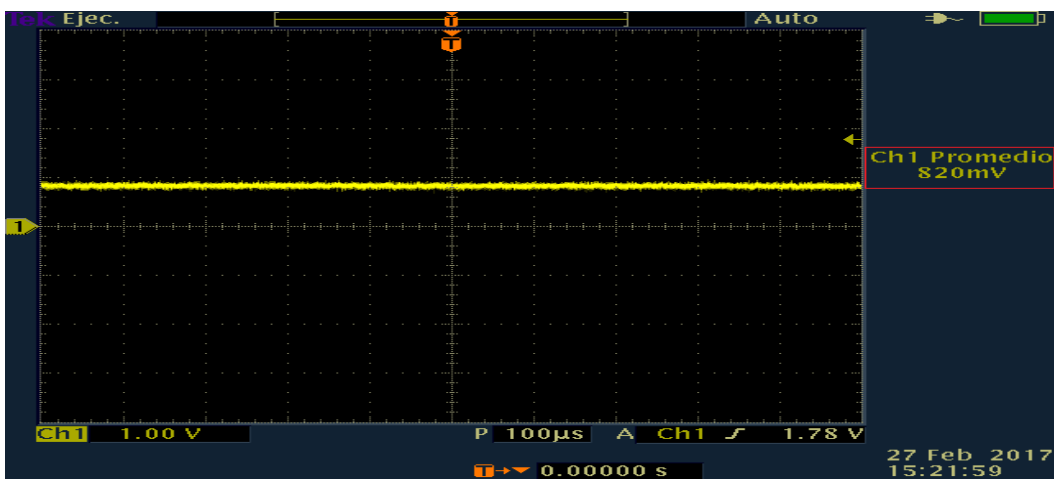


Figura 75. Duty = 0.6

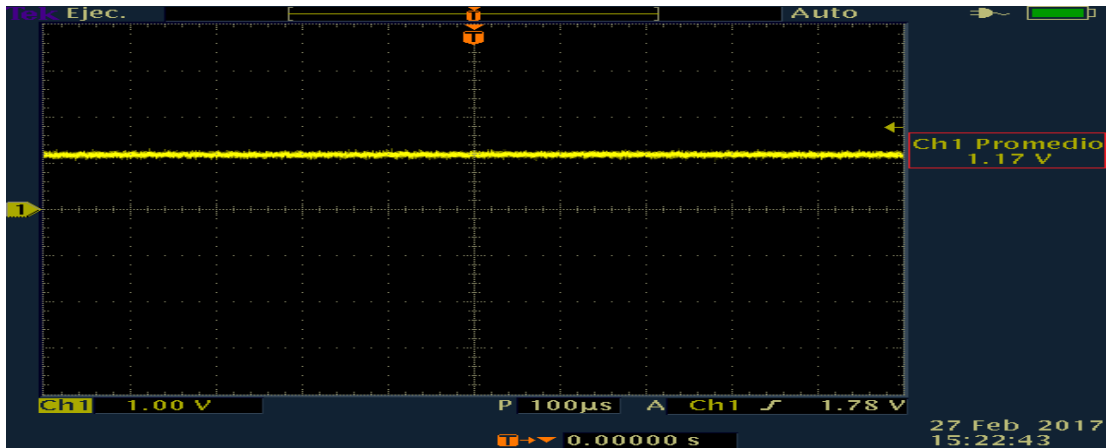


Figura 76. Duty = 0.7

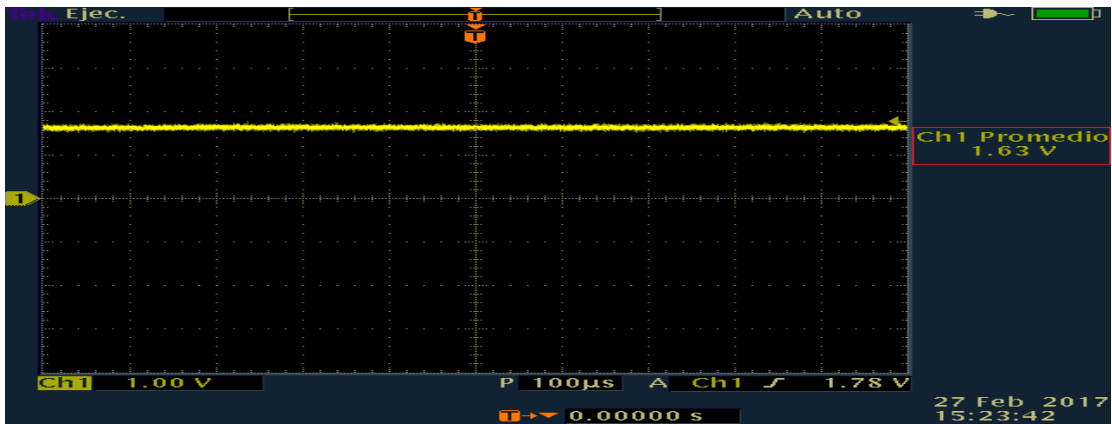


Figura 77. Duty = 0.8

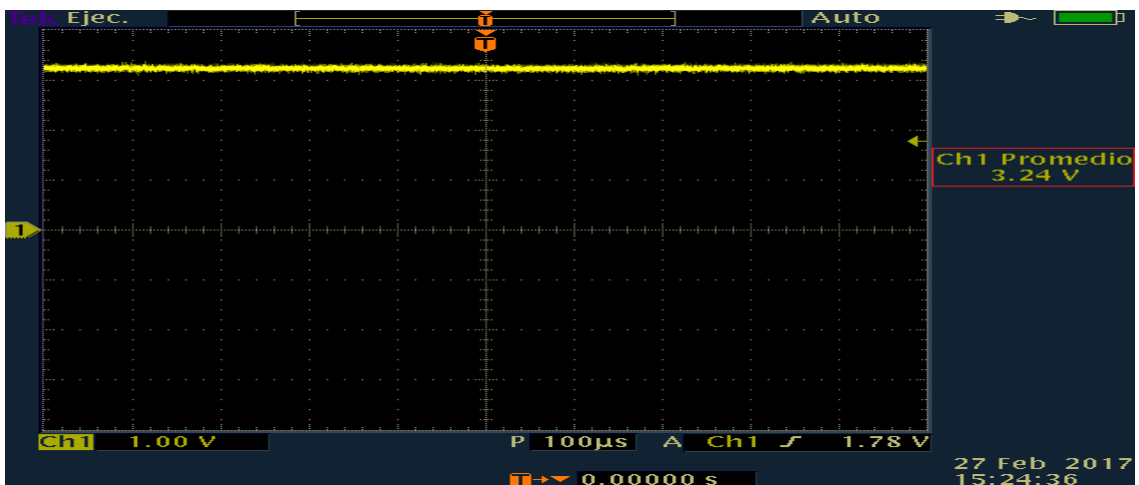


Figura 78. Duty = 0.9

Simulación Matlab (voltaje promedio):

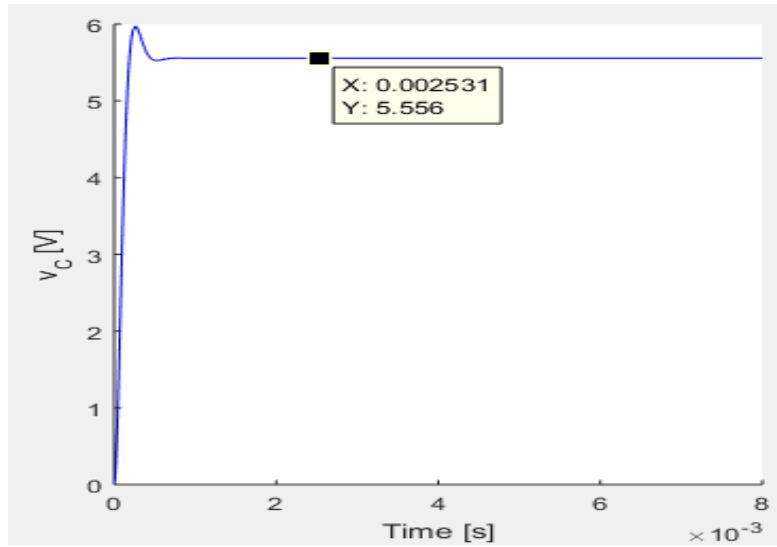


Figura 79. Duty = 0.1

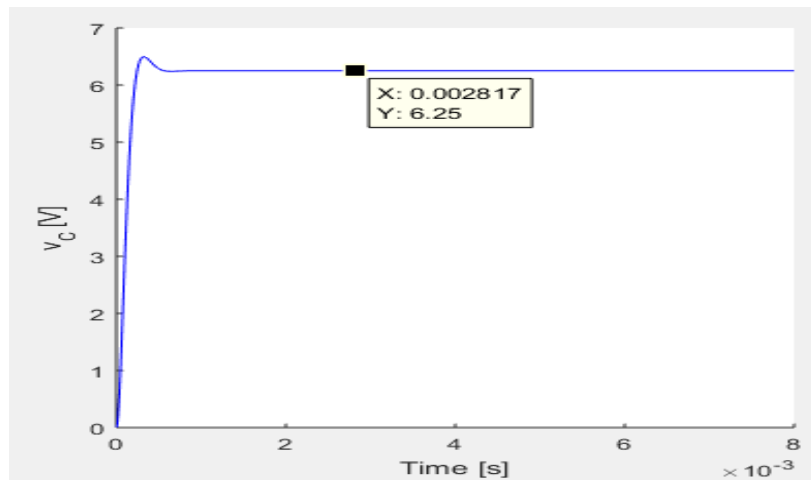


Figura 80. Duty = 0.2

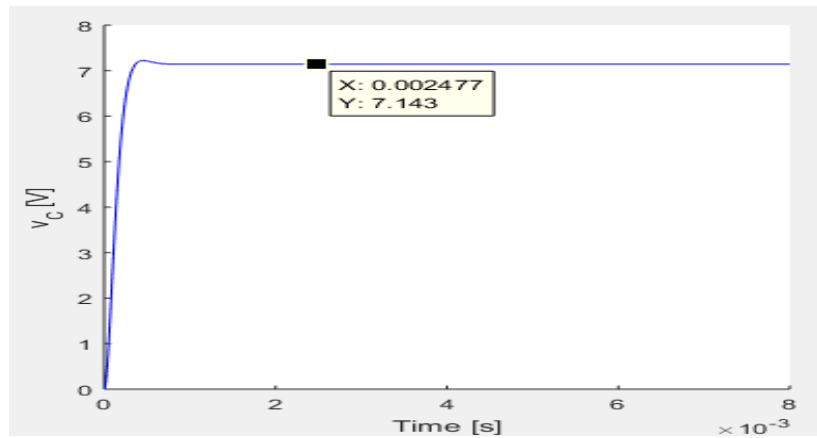


Figura 81. Duty = 0.3

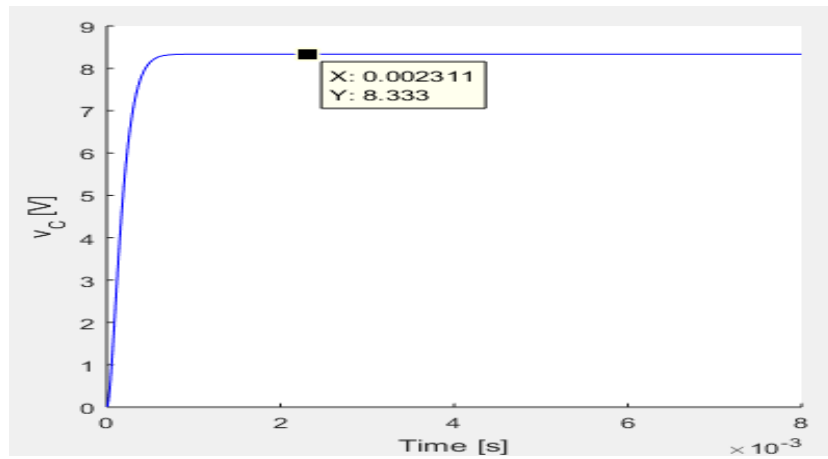


Figura 82. Duty = 0.4

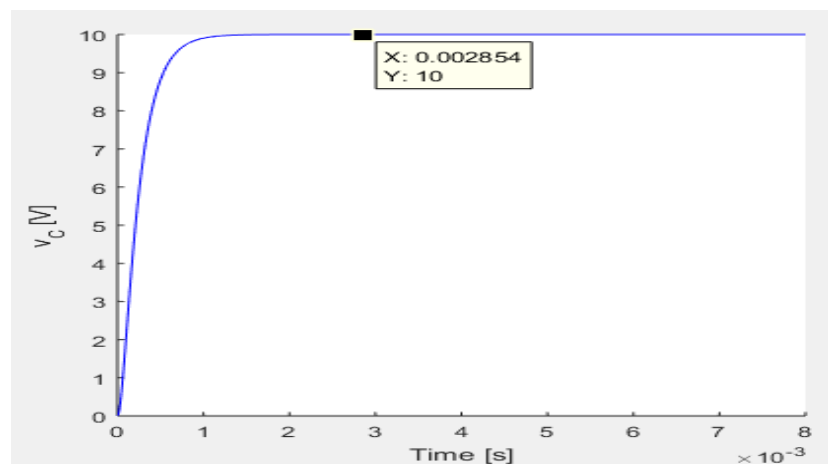


Figura 83. Duty = 0.5

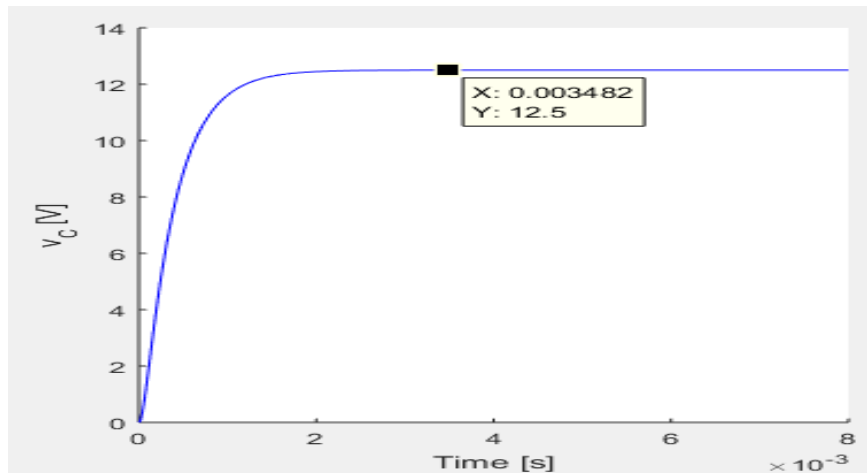


Figura 84. Duty = 0.6

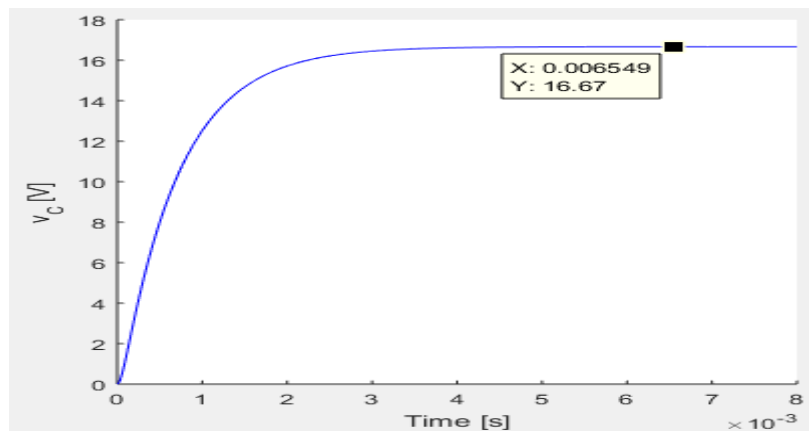


Figura 85. Duty = 0.7

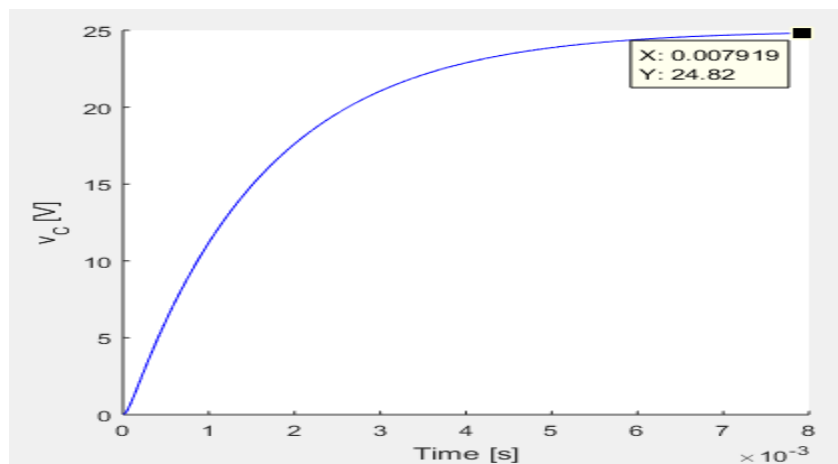


Figura 86. Duty = 0.8

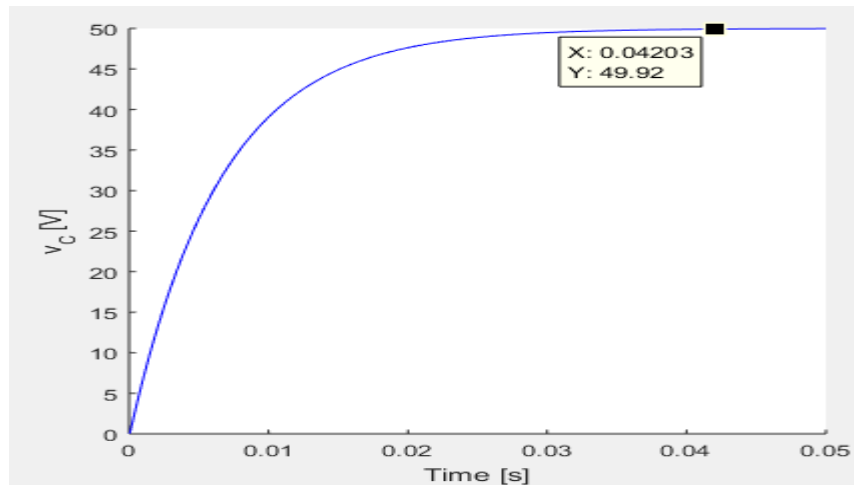


Figura 87. Duty = 0.9

Calculo porcentaje error: Se realiza la comparación de los resultados arrojados entre la Simulación en Matlab y el valor que arroja el osciloscopio del voltaje promedio de salida. El valor obtenido por el instrumento de medida es escalizado, por lo que, para poder hacer la comparación, es necesario que este valor se convierta al voltaje exacto que arroja el M (D) de cada convertidor, esto, mediante la siguiente formula.

$$V_o = \frac{V_f * 50}{3.3} \text{ Ecuación (17)}$$

Donde V_f es la señal escalizada

El cálculo del porcentaje error está dado por la siguiente formula:

$$\% \text{ error} = \left| \frac{V_{\text{experimental}} - V_{\text{teorico}}}{V_{\text{teorico}}} \right| * 100$$

La tabla 2 muestra los valores promedio V_o de cada duty hallados, el % error y los valores obtenidos en las figuras 70 a 87.

Duty	Vf, señal escalizada (V)	Vo, Voltaje promedio, Ecuación 17 (V)	Vo, Simulacion Matlab (V)	% error (%)
0.1	0.357	5.40	5.56	2.87
0.2	0.424	6.42	6.25	2.72
0.3	0.476	7.21	7.14	0.98
0.4	0.547	8.28	8.33	0.60
0.5	0.659	9.98	10	0.2
0.6	0.820	12.42	12.5	0.64
0.7	1.17	17.72	16.67	6.29
0.8	1.63	24.69	24.82	0.52
0.9	3.24	49.09	49.92	1.62

Tabla 3. % error, Señal osciloscopio vs Simulacion Matlab

Respuesta Transitoria: La grafica obtenida desde el osciloscopio es realizada a través de las coordenadas que surgen de ella. Este medidor toma todos los puntos dentro de la gráfica, por lo que el grosor y los destellos que aparecen en la señal de color rojo, se visualizan de esa forma. La señal simulada por Matlab se representa de color azul.

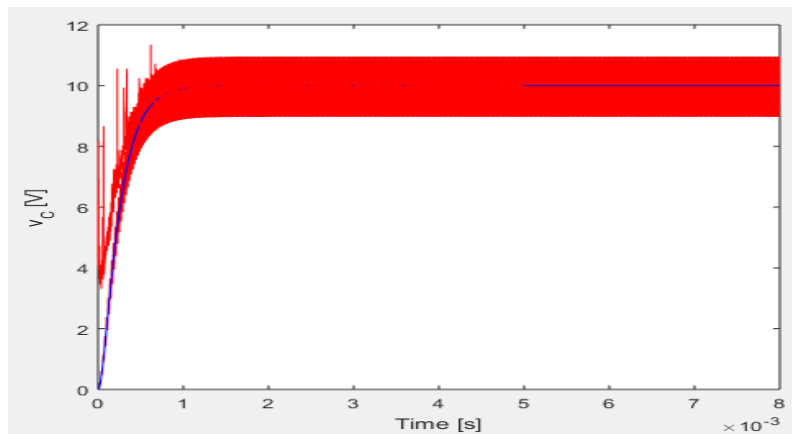


Figura 88. Respuesta transitoria, Matlab vs osciloscopio, Duty = 0.5

4.3 CONVERTIDOR BUCK - BOOST

Simulación vivado:

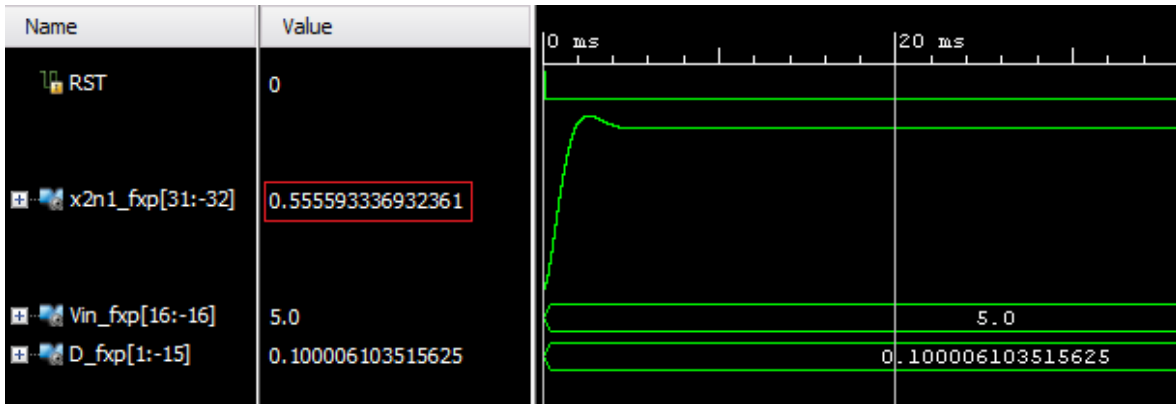


Figura 89. Duty = 0.1

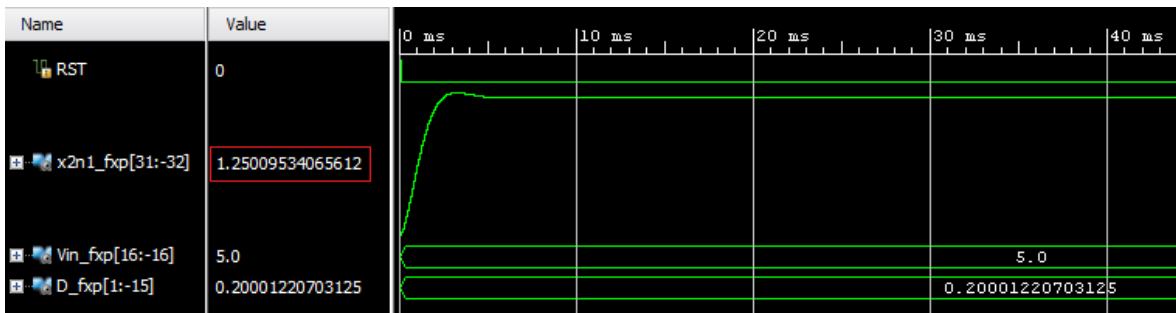


Figura 90. Duty = 0.2

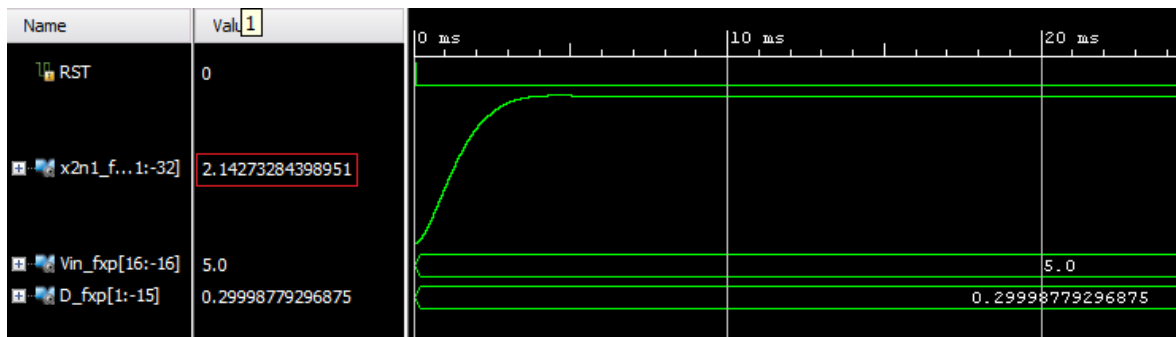


Figura 91. Duty = 0.3

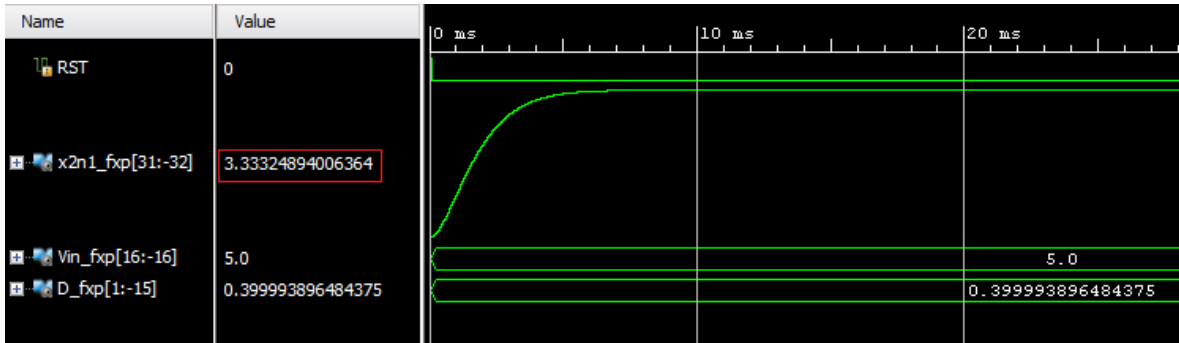


Figura 92. Duty = 0.4

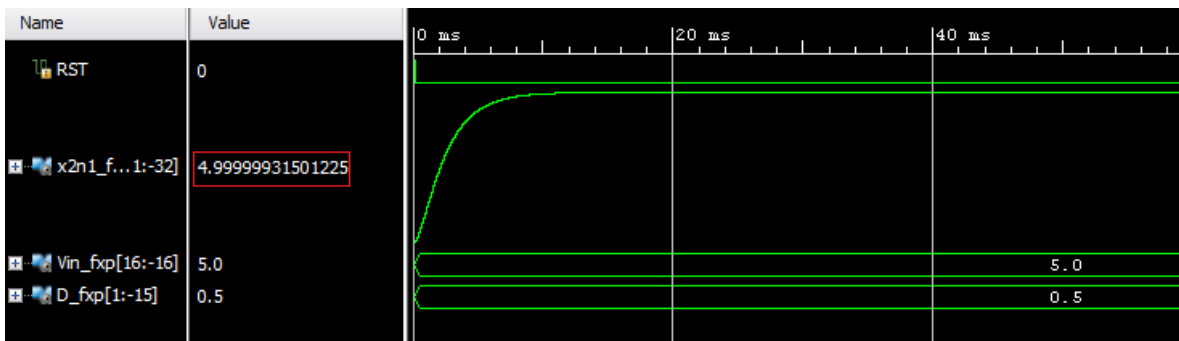


Figura 93. Duty = 0.5

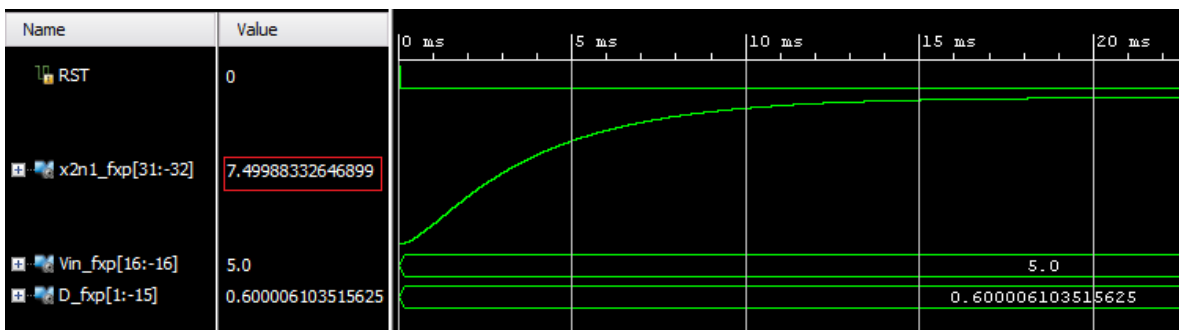


Figura 94. Duty = 0.6

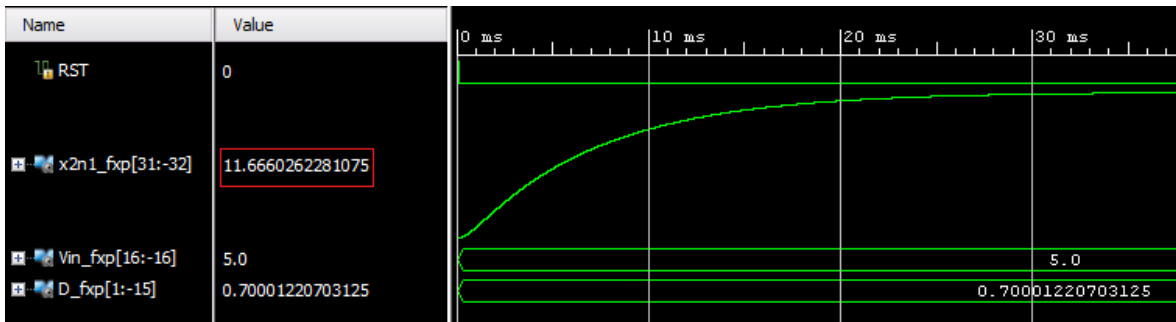


Figura 95. Duty = 0.7

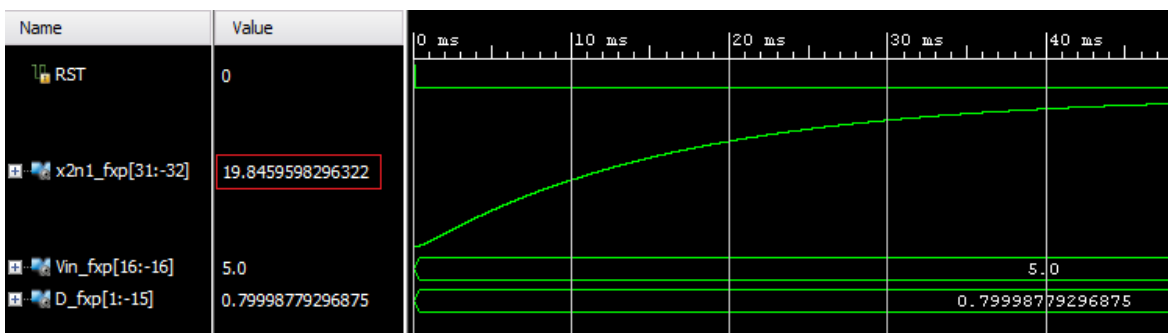


Figura 96. Duty = 0.8

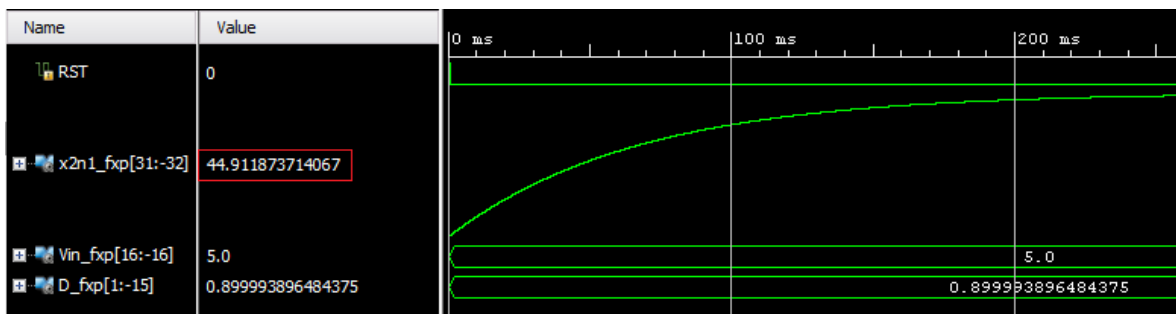


Figura 97. Duty = 0.9

Osciloscopio (voltaje promedio): Teniendo en cuenta que la salida máxima arrojada por la FPGA es 3.3 V y valor máximo del Boost es de 45 (duty de 0.9) es necesario realizar una escalización de la señal mediante la siguiente regla de tres:

$$V_f = \frac{V_o * 3.3}{45}$$

Donde V_f es el valor final escalizado y mostrado en el osciloscopio.

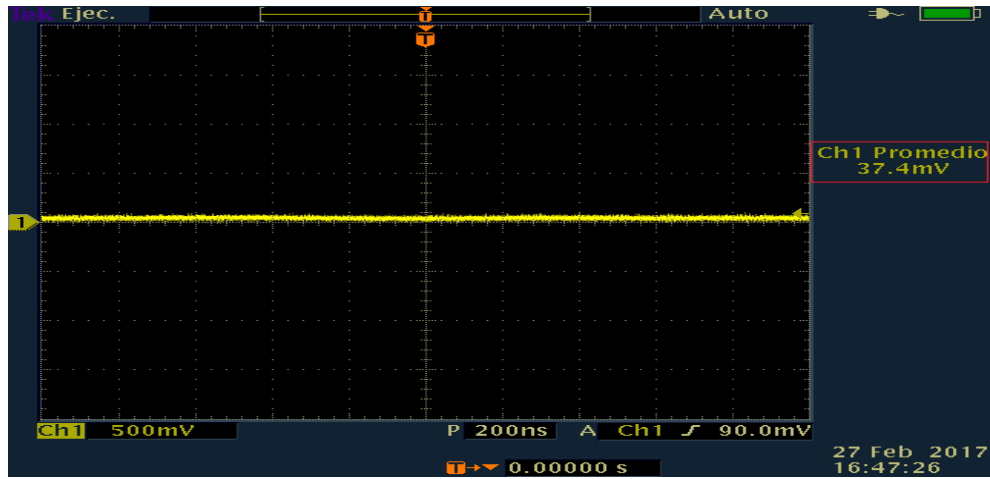


Figura 98. Duty = 0.1

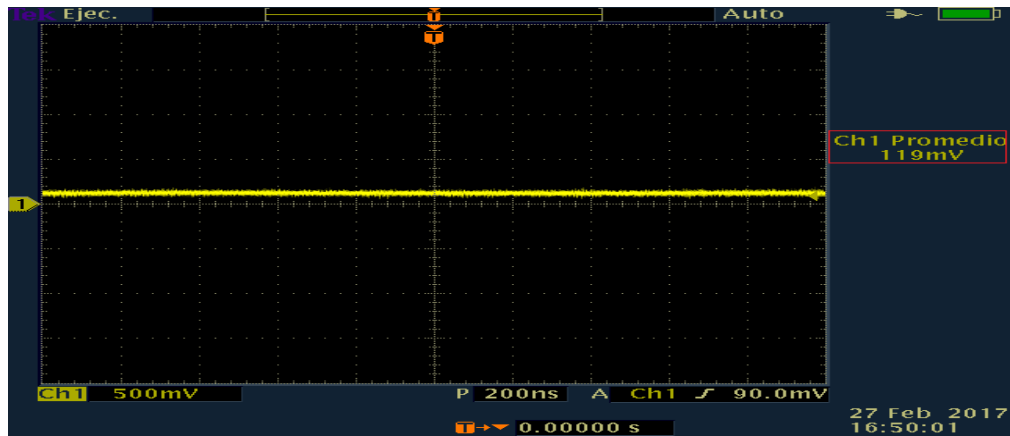


Figura 99. Duty = 0.2

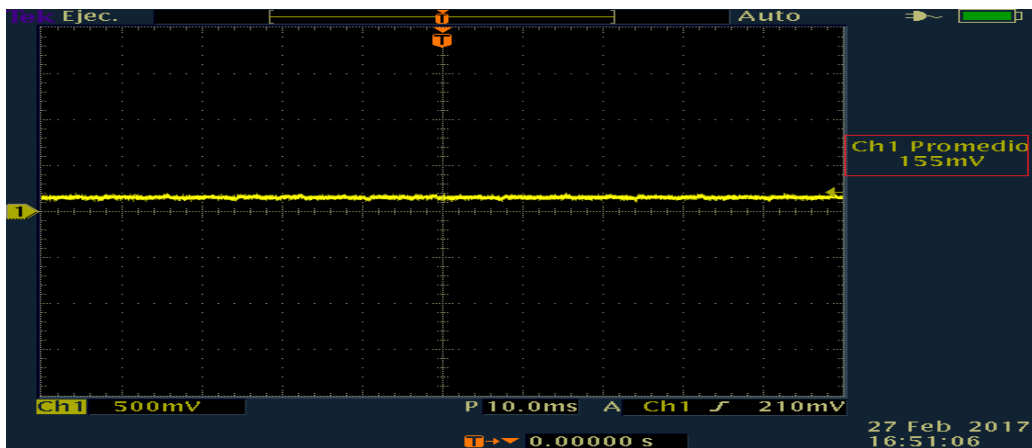


Figura 100. Duty = 0.3

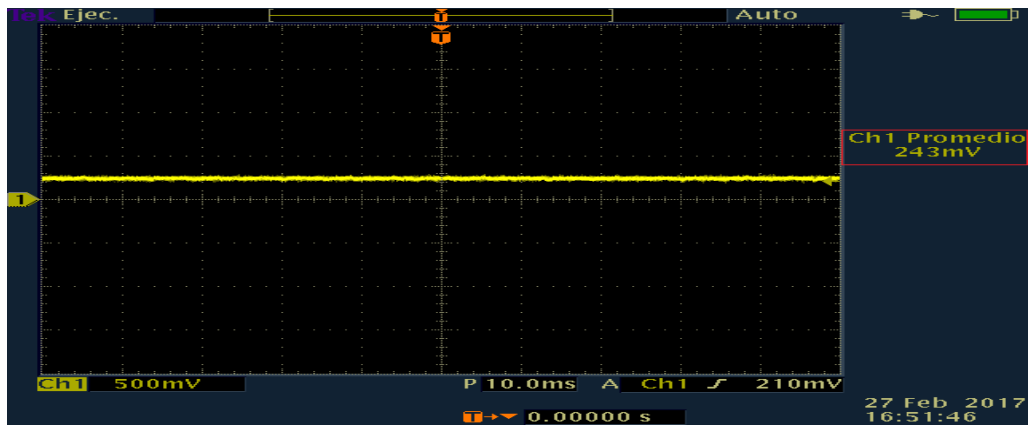


Figura 101. Duty = 0.4

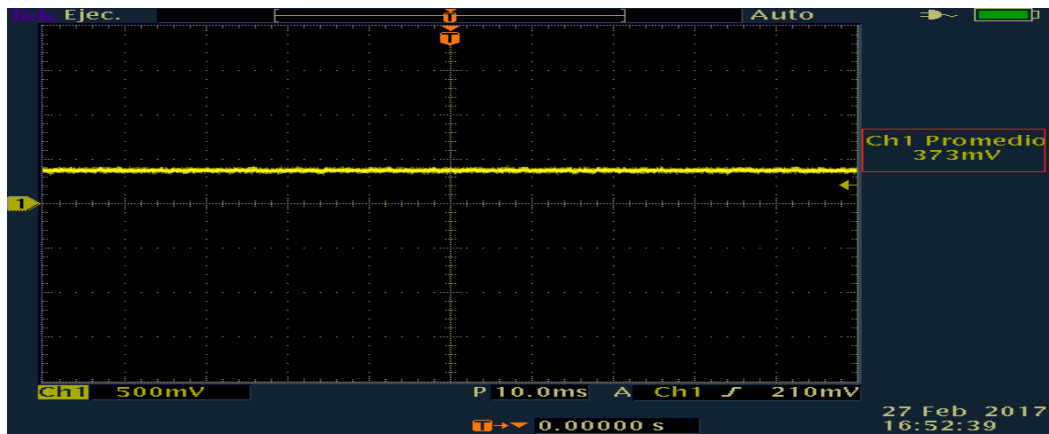


Figura 102. Duty = 0.5

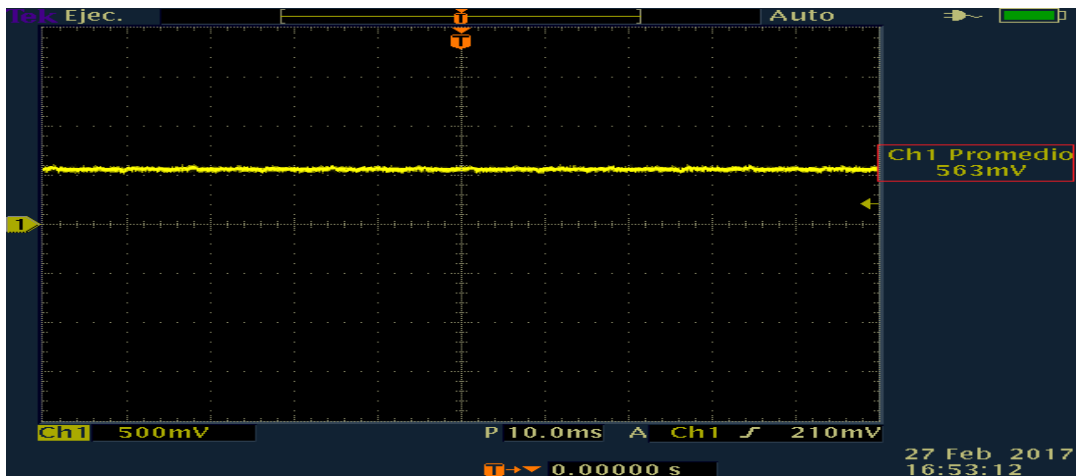


Figura 103. Duty = 0.6

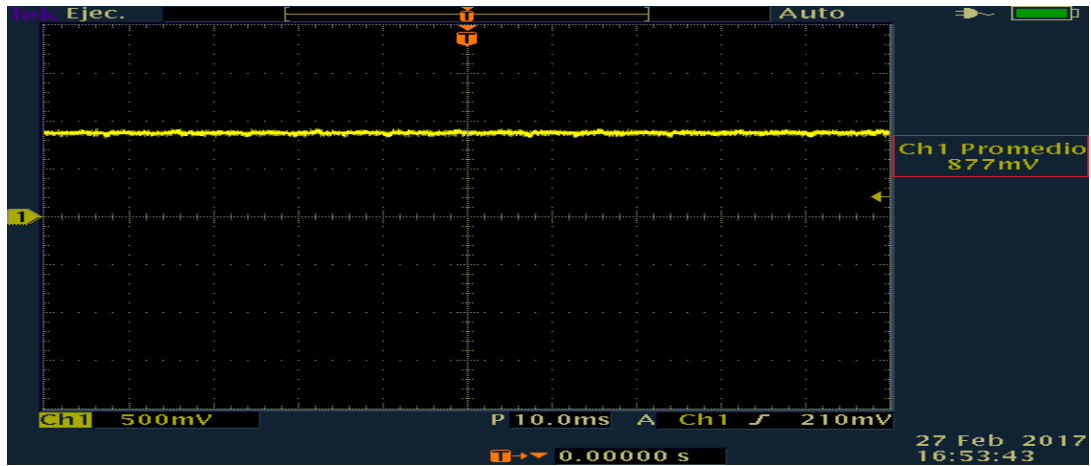


Figura 104. Duty = 0.7

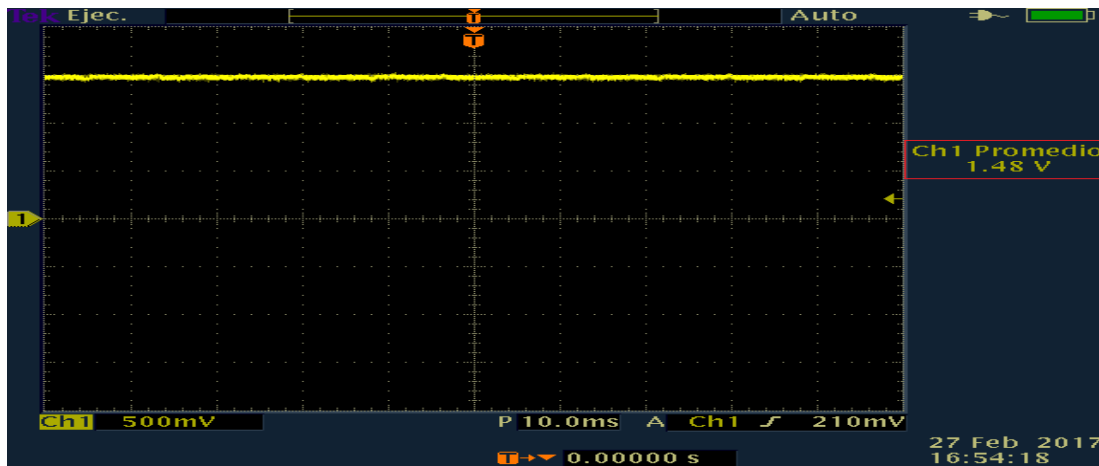


Figura 105. Duty = 0.8

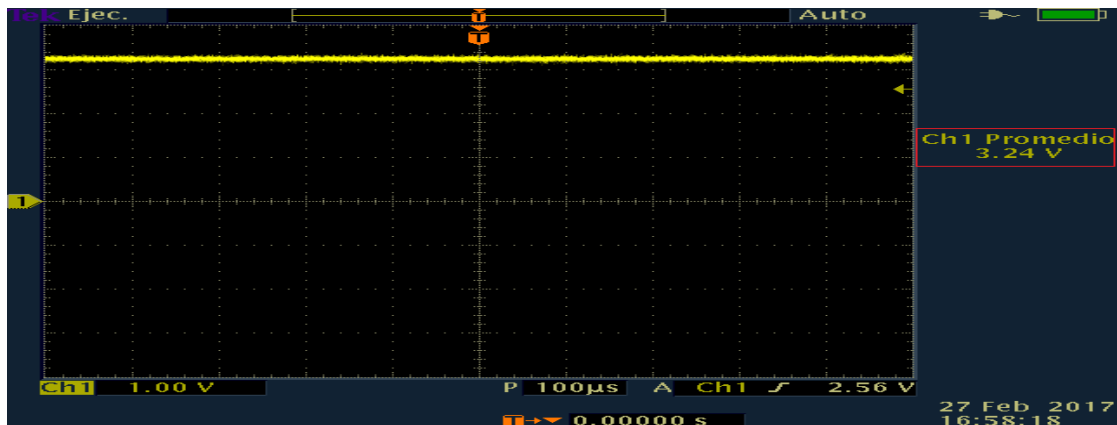


Figura 106. Duty = 0.9

Simulación Matlab (voltaje promedio):

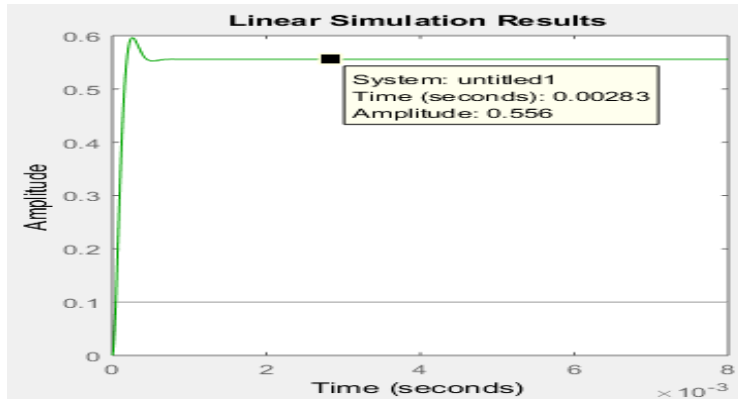


Figura 107. Duty = 0.1

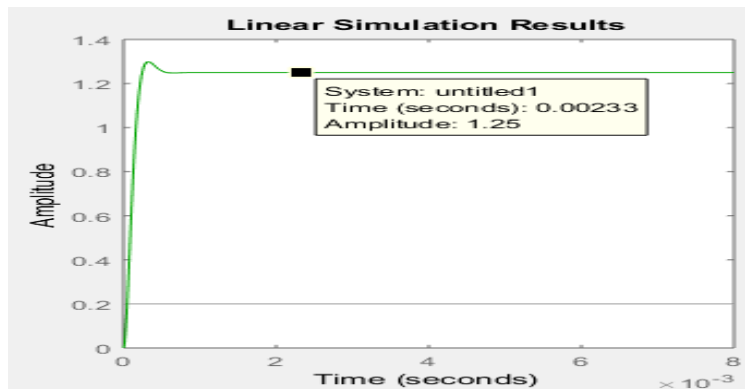


Figura 108. Duty = 0.2

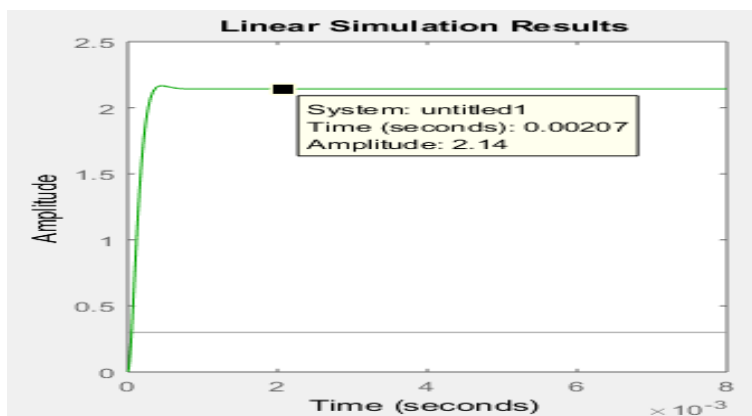


Figura 109. Duty = 0.3

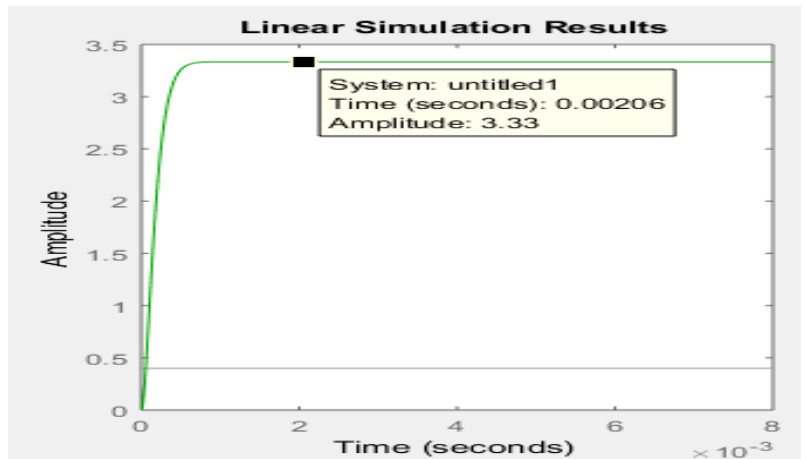


Figura 110. Duty = 0.4

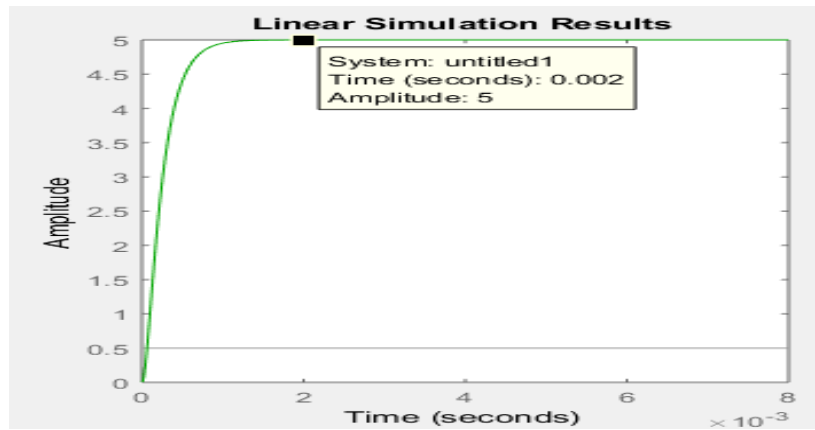


Figura 111. Duty = 0.5

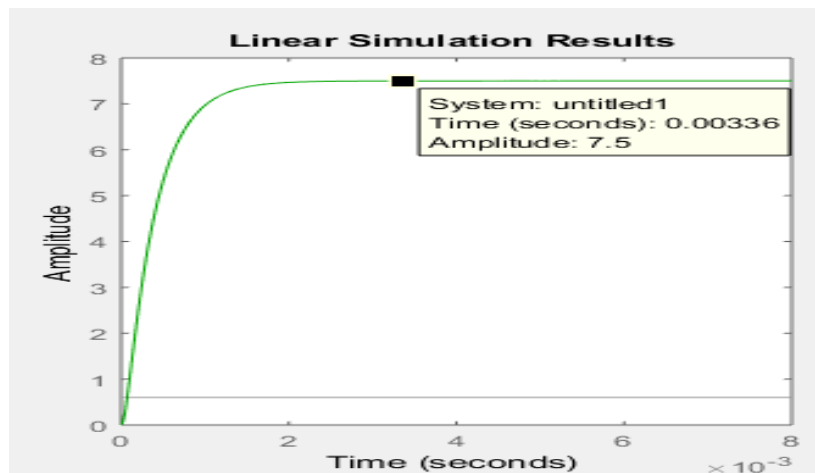


Figura 112. Duty = 0.6

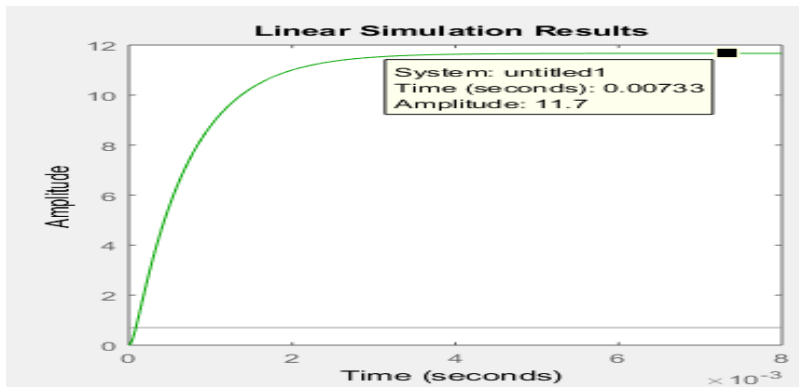


Figura 113. Duty = 0.7

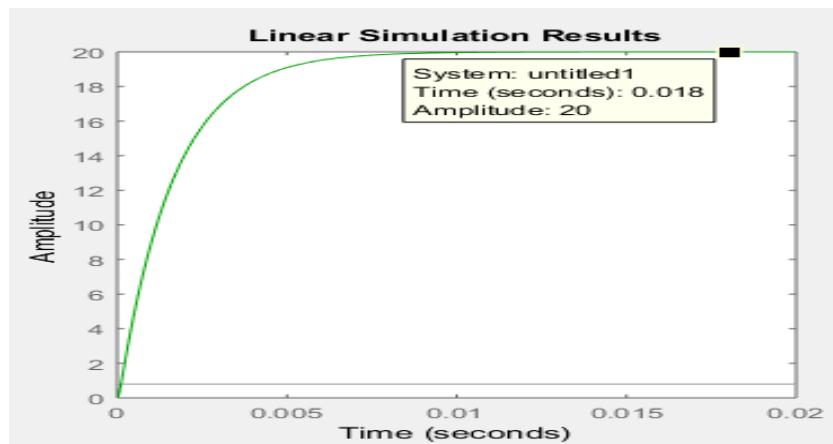


Figura 114. Duty = 0.8

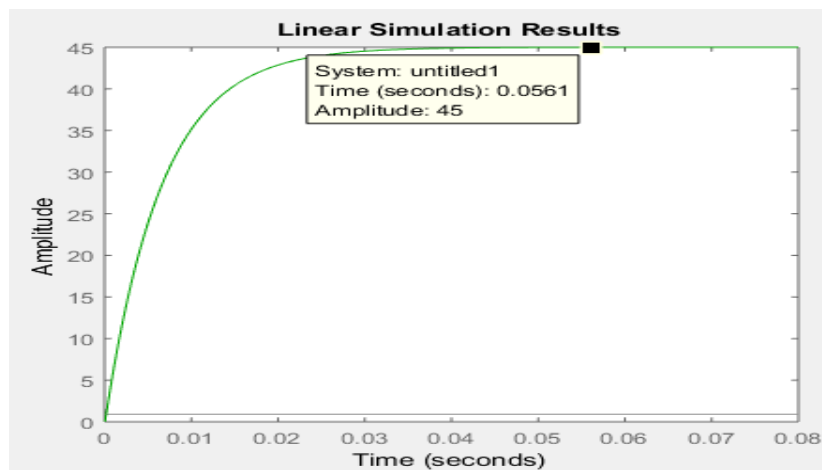


Figura 115. Duty = 0.9

	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

Calculo porcentaje error: Se realiza la comparación de los resultados arrojados entre la Simulación en Matlab y el valor que arroja el osciloscopio del voltaje promedio de salida. El valor obtenido por el instrumento de medida es escalizado, por lo que, para poder hacer la comparación, es necesario que este valor se convierta al voltaje exacto que arroja el M (D) de cada convertidor, esto, mediante la siguiente formula.

$$V_o = \frac{V_f * 45}{3.3} \text{ Ecuación (18)}$$

Donde V_f es la señal escalizada

El cálculo del porcentaje error está dado por la siguiente formula:

$$\% \text{ error} = \left| \frac{V_{\text{experimental}} - V_{\text{teorico}}}{V_{\text{teorico}}} \right| * 100$$

La tabla 2 muestra los valores promedio V_o de cada duty hallados, el % error y los valores obtenidos en las figuras 98 a 115.

Duty	V_f , señal escalizada (V)	V_o , Voltaje promedio, Ecuación 17 (V)	V_o , Simulación Matlab (V)	% error (%)
0.1	0.0374	0.51	0.55	7.27
0.2	0.119	1.62	1.25	29.6
0.3	0.155	2.11	2.14	1.40
0.4	0.243	3.31	3.33	0.60
0.5	0.373	5.08	5	1.6
0.6	0.563	7.67	7.5	2.26
0.7	0.877	11.9	11.7	1.70
0.8	1.48	20.18	20	0.9
0.9	3.24	44.18	45	1.82

Tabla 4. % error, Señal osciloscopio vs Simulación Matlab

Respuesta transitoria: Señal del osciloscopio en rojo y simulación de Matlab en Azul

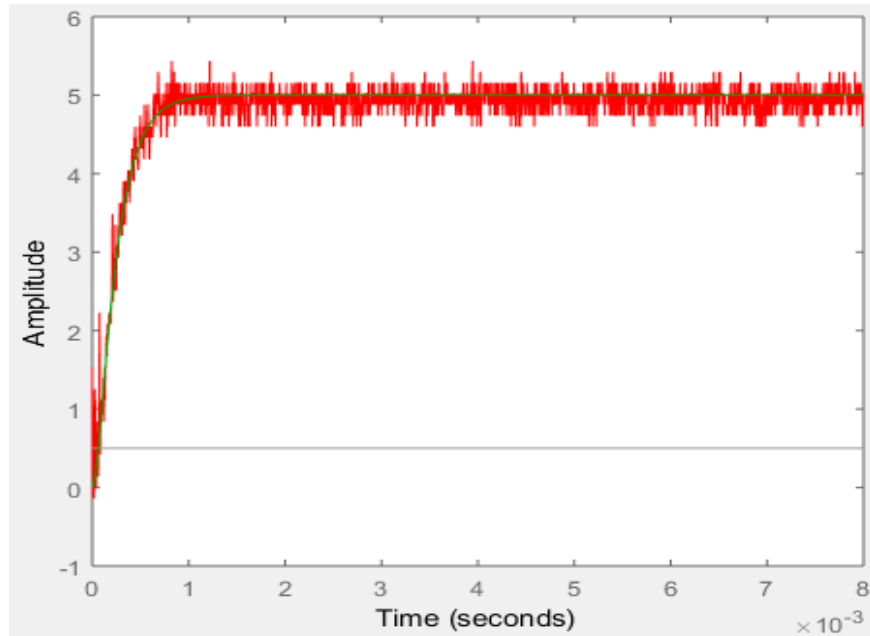


Figura 116. Respuesta transitoria, Osciloscopio vs Matlab, Duty = 0.5

Los resultados obtenidos en el cálculo del porcentaje error fueron relativamente bajos. De las 27 medidas (los tres convertidores), el 96.3 % de estas, se encontró con un error por debajo del 8.8 %. El 3.7 % faltante (una medida) se da con un error del 29.6 %, insignificante por la precisión que se logró en el resto de las mediciones.

Los valores del voltaje promedio obtenidos en la Simulación de Vivado y Matlab fueron prácticamente exactos, demostrando que es necesario poseer dos o más herramientas que permitan verificar el correcto diseño del sistema a implementar.

Finalmente, las respuestas transitorias experimentales de cada convertidor obtenidas por el osciloscopio estuvieron muy similares en comparación con las halladas en la Simulación de Matlab. Cabe aclarar que, para poder obtener la gráfica arrojada por el medidor, se realizó la extracción de las coordenadas que generaba dicha gráfica, haciendo que, al graficar estos puntos en Matlab, la imagen se visualizara con cualquier tipo de ruido. Esto se notó más específicamente en la señal capturada en el convertidor Boost.

	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

5 CONCLUSIONES, RECOMENDACIONES Y TRABAJO FUTURO

La ingeniería es un área que desde sus inicios ha venido evolucionando exponencialmente a tal punto de que el desarrollo de cualquier tipo de sistemas se puede diseñar, verificar, comprobar y mejorar mediante muchas herramientas, técnicas de modelamiento, lenguajes de programación, entre otros. El diseño y desarrollo elaborado en Vivado comparado con Matlab, arroja datos demasiado exactos, permitiendo que se genere confianza en la eficiencia y eficacia del progreso y resultado final del proyecto.

Por otro lado, la electrónica de potencia viene contribuyendo notoriamente en el desarrollo de nuevas estructuras para el procesamiento de la energía. Es un tema que es muy importante y que merece ser estudiado profundamente. Al ser una materia de estudio interesante, prácticamente es necesario realizar un prototipo que muestre su principal funcionamiento. Gracias a VHDL, Hardware in the Loop HIL y a la tarjeta de desarrollo FPGA, sumado a los resultados obtenidos, es de concluir que se obtiene un prototipo casi exacto de los tres convertidores DC a DC que son el Buck, Boost y Buck Boost.

En conclusión, es posible obtener los resultados a una escala menor de cualquier tipo de planta o sistema si se realiza una investigación de la temática necesaria para ejecutar el proyecto. Es importante no solo recurrir a textos en internet, sino que también es significativo acudir a personas con grandes conocimientos en el Tema. Después de tener claro la temática, usar una metodología precisa que implemente el paso correcto a realizar será de gran ayuda.

	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

Se logran mediciones muy cercanas a las ideales, pero, es de mencionar que dichos resultados pueden ser más precisos si se tienen en cuenta algunos factores como:

- 6 Implementar una técnica de modelamiento y desarrollo de este más exacto
- 7 Tener en cuenta que existen perdidas en los elementos del medidor a la hora de realizar las mediciones, de igual forma, utilizar un osciloscopio que permita obtener graficas con trazados más finos
- 8 Emplear un conversor digital – analógico serial con una mayor frecuencia de trabajo que permita obtener una conversión más precisa.
- 9 Al utilizarse el Método de Euler, el cual no es el más preciso de todos. Para este método en particular, quizás, sea indispensable usar un hardware con una mayor frecuencia de muestreo.

Como trabajo futuro, se halla actualmente el desarrollo de un control PI para los tres convertidores ya mencionados. Se tiene finalizado el código en VHDL, pero que por cuestiones de muestreo aun no es posible obtener los resultados esperados. Seguirá en estudio e investigación hasta que el control del prototipo ya implementado sea correctamente finalizado.

	<p style="text-align: center;">INFORME FINAL DE TRABAJO DE GRADO</p>	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

REFERENCIAS

- Fundamental of power Electronics. Autores: Erickson, Robert W, Maksimovic, Dragan. 2001
- Bose Power Electronics and Motor Driver advanced and trends. Autores: Bimal K. 2006
- Blog: Que son las FPGA, Como funcionan, Para qué sirven, quien debería usarlos. <http://nodoelectronico.com/que-son-los-fpgas-como-funcionan-para-que-sirven-quien-deberia-utilizarlos/>. Autores: Emmanuel Rojas. 2016
- Blog: What is Hardware in the Loop. Why Perform Hardware in the Loop Simulation. https://es.mathworks.com/help/physmod/simscape/ug/what-is-hardware-in-the-loop-simulation.html?s_tid=gn_loc_drop. Autores: MathWorks. 2016
- Blog: VHDL, Lenguaje de descripción de Hardware. https://rua.ua.es/dspace/bitstream/10045/3928/1/S2_2_ESTRUCTURA_BASICA_DE_DISEÑO.pdf. Autores: Universidad de Alicante. 2007

 Institución Universitaria	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

APÉNDICE

APENDICE A

Código VHDL Convertidor Buck

```

library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.STD_LOGIC_UNSIGNED.ALL;
use IEEE.STD_LOGIC_ARITH.ALL;
library IEEE_proposed;
use IEEE_proposed.fixed_pkg.all;
entity BUCK is
    Port (CLK: in STD_LOGIC; RST: in STD_LOGIC; D: in sfixed (1 downto -15);
          Vo : out STD_LOGIC_VECTOR (7 downto 0) := (others => '0'));
end BUCK;
architecture Behavioral of BUCK is
type estados is (e0, e1);
signal estado: estados;
signal x1n_fxp: sfixed (31 downto -32):= to_sfixed (0,31, -32);
signal x1n1_fxp: sfixed (31 downto -32):= to_sfixed (0,31, -32);
signal x2n_fxp: sfixed (31 downto -32):= to_sfixed (0, 31, -32);
signal x2n1_fxp: sfixed (31 downto -32):= to_sfixed (0, 31, -32);
signal Vin_fxp: sfixed (16 downto -16);
signal R_fxp: sfixed (31 downto -8);
signal L_fxp: sfixed (16 downto -16);
signal C_fxp: sfixed (1 downto -32);
signal D_fxp: sfixed (1 downto -15);
signal h_fxp: sfixed (1 downto -32);
signal clk_base: sfixed (1 downto -32);
signal n_fxp: sfixed (31 downto 0);
signal k, n: std_logic_vector (31 downto 0):= (others => '0');
signal clk_sys: std_logic:= '0';
signal k1,k2,k3: sfixed (16 downto 0);
begin
Vin_fxp <= to_sfixed (5,1 6, -16);
R_fxp<= to_sfixed (5, 31, -8);
L_fxp <= to_sfixed (0.000330, 16, -16);
C_fxp<= to_sfixed (0.000010, 1, -32);

```

 Institución Universitaria	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

```

D_fxp<= D;
h_fxp<= to_sfixed (0.00000008, 1, -32);
clk_base <= to_sfixed (0.00000004 ,1, -32);
n_fxp<= resize (h_fxp/clk_base, n_fxp);
n<= to_slv (n_fxp);
k1<= to_sfixed (4.5, 16, 0);
k2<= to_sfixed (255, 16, 0);
process (clk, k, n)
begin
  if (rising_edge(CLK)) then
    if (k < n-1) then
      k <= k + '1';
    else
      clk_sys <= not(clk_sys);
      k <= (others => '0');
    end if;
  end if;
end process;
process (Vin_fxp, R_fxp, C_fxp, h_fxp, x1n_fxp, x1n1_fxp, x2n_fxp, x2n1_fxp, clk_sys)
begin
  if (RST = '1') then
    x1n_fxp <= to_sfixed (0, 31, -32); x2n_fxp <= to_sfixed (0, 31, -32);
    x1n1_fxp <= to_sfixed (0, 31, -32);
    x2n1_fxp <= to_sfixed (0, 31, -32);
  else
    if (rising_edge(clk_sys)) then
      case estado is
        when e0 =>
x1n1_fxp <= resize (((vin_fxp*d_fxp)/l_fxp) *h_fxp) - ((x2n_fxp*h_fxp)/l_fxp) + x1n1_fxp,
x1n1_fxp); --Corriente Bobina
x2n1_fxp <= resize (((x1n_fxp/c_fxp) *h_fxp) - ((x2n1_fxp*h_fxp)/(r_fxp*c_fxp)) +
x2n_fxp, x2n1_fxp); --Voltaje Condensador
estado <= e1;
          when e1 =>
            x1n_fxp <= x1n1_fxp; x2n_fxp <= x2n1_fxp;
            estado <= e0;
          end case;
        end if;
      end if;
    end process;
    k3 <= resize((x2n1_fxp*k2)/k1, k3); Vo <= to_slv (k3(7 downto 0));
  end Behavioral;

```

 Institución Universitaria	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

APENDICE B

Código VHDL Convertidor Boost

```

library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.STD_LOGIC_UNSIGNED.ALL;
use IEEE.STD_LOGIC_ARITH.ALL;
library IEEE_proposed;
use IEEE_proposed.fixed_pkg.all;
entity Boost is
    Port (CLK: in STD_LOGIC; RST: in STD_LOGIC; D: in sfixed (1 downto -15);
          Vo: out STD_LOGIC_VECTOR (7 downto 0):= (others => '0'));
end Boost;
architecture Behavioral of Boost is
type estados is (e0, e1);
signal estado: estados;
signal x1n_fxp: sfixed (31 downto -32):= to_sfixed (0, 31, -32);
signal x1n1_fxp: sfixed (31 downto -32):= to_sfixed (0, 31, -32);
signal x2n_fxp: sfixed (31 downto -32):= to_sfixed (0, 31, -32);
signal x2n1_fxp: sfixed (31 downto -32):= to_sfixed (0, 31, -32);
signal Vin_fxp: sfixed (16 downto -16);
signal R_fxp: sfixed (31 downto -8);
signal L_fxp: sfixed (16 downto -16);
signal C_fxp: sfixed (1 downto -32);
signal D_fxp: sfixed (1 downto -15);
signal h_fxp: sfixed (1 downto -32);
signal clk_base: sfixed (1 downto -32);
signal n_fxp: sfixed (31 downto 0);
signal k, n: std_logic_vector (31 downto 0):= (others => '0');
signal clk_sys: std_logic:= '0';
signal k1,k2,k3: sfixed (16 downto 0);
begin
Vin_fxp<= to_sfixed (5, 16, -16);
R_fxp <= to_sfixed (5, 31, -8);
L_fxp<= to_sfixed (0.000330, 16, -16);
C_fxp <= to_sfixed (0.000010, 1, -32);
D_fxp<= D;
h_fxp<= to_sfixed (0.00000008, 1, -32);
clk_base<= to_sfixed (0.00000004, 1, -32);
n_fxp<= resize (h_fxp/clk_base, n_fxp);
n<= to_slv(n_fxp);

```

 Institución Universitaria	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

```

k1<= to_sfixed (50, 16 ,0);
k2<= to_sfixed (255, 16, 0);
process (clk, k, n)
begin
  if (rising_edge(CLK)) then
    if (k < n-1) then
      k <= k + '1';
    else
      clk_sys <= not(clk_sys);
      k <= (others => '0');
    end if;
  end if;
end process;
process (Vin_fxp, R_fxp, C_fxp, h_fxp, x1n_fxp, x1n1_fxp, x2n_fxp, x2n1_fxp, clk_sys)
begin
  if (RST = '1') then
    x1n_fxp <= to_sfixed (0, 31, -32); x2n_fxp <= to_sfixed (0, 31, -32);
    x1n1_fxp <= to_sfixed (0, 31, -32);
    x2n1_fxp <= to_sfixed (0, 31, -32);
  else
    if (rising_edge(clk_sys)) then
      case estado is
        when e0 =>
          x1n1_fxp <= resize (((Vin_fxp * h_fxp)/ L_fxp) - (((x2n_fxp * (1-D_fxp))/L_fxp) *h_fxp) +
          x1n1_fxp, x1n1_fxp); --Corriente Bobina
          x2n1_fxp <= resize(-((x2n1_fxp*h_fxp)/(R_fxp*C_fxp)) + (((x1n_fxp * (1-D_fxp))/C_fxp)
          *h_fxp) + x2n_fxp, x2n1_fxp); --Voltaje Condensador
          estado <= e1;
          when e1 =>
            x1n_fxp <= x1n1_fxp;
            x2n_fxp <= x2n1_fxp;
            estado <= e0;
          end case;
        end if;
      end if;
    end process;
    k3 <= resize((x2n1_fxp*k2)/k1, k3);
    Vo <= to_slv (k3(7 downto 0));

end Behavioral;

```

 Institución Universitaria	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

APENDICE C

Código Convertidor Buck – Boost

```

library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.STD_LOGIC_UNSIGNED.ALL;
use IEEE.STD_LOGIC_ARITH.ALL;
library IEEE_proposed;
use IEEE_proposed.fixed_pkg.all;
entity BUCK_BOOST is
  Port (CLK: in STD_LOGIC; RST: in STD_LOGIC;
        D: in sfixed (1 downto -15);
        Vo: out STD_LOGIC_VECTOR (7 downto 0):= (others => '0'));
end BUCK_BOOST;
architecture Behavioral of BUCK_BOOST is
type estados is (e0, e1);
signal estado: estados;
signal x1n_fxp: sfixed (31 downto -32):= to_sfixed (0, 31, -32);
signal x1n1_fxp: sfixed (31 downto -32):= to_sfixed (0, 31, -32);
signal x2n_fxp: sfixed (31 downto -32):= to_sfixed (0, 31, -32);
signal x2n1_fxp: sfixed (31 downto -32):= to_sfixed (0, 31, -32);
signal Vin_fxp: sfixed (16 downto -16);
signal R_fxp: sfixed (31 downto -8);
signal L_fxp: sfixed (16 downto -16);
signal C_fxp: sfixed (1 downto -32);
signal D_fxp: sfixed (1 downto -15);
signal h_fxp: sfixed (1 downto -32);
signal clk_base: sfixed (1 downto -32);
signal n_fxp: sfixed (31 downto 0);
signal k, n: std_logic_vector (31 downto 0):= (others => '0');
signal clk_sys: std_logic:= '0';
signal k1,k2,k3: sfixed (16 downto 0);
begin
Vin_fxp<= to_sfixed (5, 16, -16);
R_fxp<= to_sfixed (5, 31, -8);
L_fxp<= to_sfixed (0.000330 ,16, -16);
C_fxp<= to_sfixed (0.000010, 1, -32);
D_fxp<= D;
h_fxp<= to_sfixed (0.00000008, 1, -32);

```

 Institución Universitaria	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

```

clk_base<= to_sfixed (0.00000004 ,1, -32);
n_fxp<= resize (h_fxp/clk_base, n_fxp);
n<= to_slv(n_fxp);
k1<= to_sfixed (45, 16, 0);
k2<= to_sfixed (255, 16, 0);
process (clk, k, n)
begin
  if (rising_edge(CLK)) then
    if (k < n-1) then
      k <= k + '1';
    else
      clk_sys <= not(clk_sys);
      k <= (others => '0');
    end if;
  end if;
end process;
process (Vin_fxp, R_fxp, C_fxp, h_fxp, x1n_fxp, x1n1_fxp, x2n_fxp, x2n1_fxp, clk_sys)
begin
  if (RST = '1') then
    x1n_fxp<= to_sfixed (0, 31, -32); x2n_fxp<= to_sfixed (0, 31, -32);
    x1n1_fxp <= to_sfixed (0, 31, -32);
    x2n1_fxp<= to_sfixed (0, 31, -32);
  else
    if (rising_edge(clk_sys)) then
      case estado is
        when e0 =>
x1n1_fxp <= resize (((vin_fxp*d_fxp)/l_fxp) *h_fxp) - (((x2n_fxp*(1-d_fxp))/l_fxp)
*h_fxp) + x1n1_fxp, x1n1_fxp); --Corriente Bobina
x2n1_fxp <= resize (((x1n_fxp*(1-d_fxp))/c_fxp) *h_fxp) -
((x2n1_fxp*h_fxp)/(r_fxp*c_fxp)) + x2n_fxp, x2n1_fxp); --Voltaje Condensador
estado <= e1;
          when e1 =>
            x1n_fxp <= x1n1_fxp; x2n_fxp <= x2n1_fxp;
            estado <= e0;
          end case;
        end if;
      end if;
    end process;
    k3 <= resize((x2n1_fxp/K1) *k2, k3);
    Vo <= to_slv (k3(7 downto 0));

end Behavioral;

```

	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

FIRMA ESTUDIANTES

Joaquín Gujo

Sergio Ignacio San

FIRMA ASESOR _____

FECHA ENTREGA: 3 de abril de 2017

FIRMA COMITÉ TRABAJO DE GRADO DE LA FACULTAD _____

RECHAZADO___ ACEPTADO___ ACEPTADO CON MODIFICACIONES___

ACTA NO. _____

FECHA ENTREGA: _____

FIRMA CONSEJO DE FACULTAD _____

ACTA NO. _____

FECHA ENTREGA: _____