

Bitácora

Proyecto Grupal 1

José Bernardo Barquero Bonilla

2023150476

Jose Eduardo Campos Salazar

2013135620

Jimmy Feng Feng

2023060347

Alexander Montero Vargas

2023166058

Instituto Tecnológico de Costa Rica

Escuela de Ingeniería en Computadores

CE1107 - Fundamentos de Arquitectura de Computadores

Profesor

Luis Chavarría Zamora

20 de Mayo de 2025

Desarrollo por Días

01/05/2025

Encargados

Jose Barquero, Jose Edo Campos, Jimmy Feng y Alexander Montero

Objetivo

Comprender mejor lo solicitado en el enunciado del proyecto y realizar una distribución de las tareas.

Procedimiento

- Se realizó una reunión virtual mediante Google Meet para entender lo solicitado.
- Discusión de posibles formas de solucionar el proyecto.
- División equitativa del trabajo.

Resultados y Observaciones

Todos los estudiantes comprendieron mejor lo solicitado en el proyecto y a su vez, ya saben qué tareas les quedó asignado a cada uno para la realización del proyecto.

01/05/2025

Encargados

Jimmy Feng

Objetivo

Avanzar parte de los módulos asignados en SystemVerilog. Específicamente para el display de 7 segmentos.

Procedimiento

- Se comprendió la tarea asignada del display 7 segmentos.
- Se pensaron en diferentes soluciones para implementar este módulo.
- Se realizó la implementación en SystemVerilog de este módulo.

Resultados y Observaciones

Se logró crear exitosamente el módulo *HEX_SevenSeg* el cual toma el resultado de 4 bits, obtenido mediante una ecuación booleana, directamente lo pasa al display de 7 segmentos en forma hexadecimal. Es un funcionamiento similar a un BCD, sin embargo en Hexadecimal [1].

08/05/2025

Encargados

Jose Barquero, Jose Edo Campos, Jimmy Feng y Alexander Montero

Objetivo

Comprender de mejor forma lo solicitado en el proyecto.

Procedimiento

- Los estudiantes se pusieron en contacto mediante whatsapp para entender de mejor forma el proyecto.
- Se realizaron consultas al profesor sobre cosas que aún no se entendían.
- También, se le hizo la consulta al profesor de realizar algo extra al proyecto y si aceptó.

Resultados y Observaciones

Los estudiantes comprendieron de mejor forma lo solicitado en el enunciado del proyecto, gracias a las consultas realizadas al profesor, además de la retroalimentación dada por cada miembro del mismo. También, se llegó a un acuerdo con el profesor para hacer algo extra en el proyecto. Ese extra consiste en usar una conexión WiFi o Bluetooth (aún por definir) entre una app mobile y un NodeMCU para enviar datos mediante el protocolo UART a la FPGA.

08/05/2025

Encargados

Jose Barquero

Objetivo

Avanzar con la propuesta de investigación del proyecto.

Procedimiento

- El estudiante comprendió lo solicitado en el enunciado del proyecto para la realización de la estructura de la propuesta de investigación.
- Se hizo uso de la herramienta ChatGPT para generar una estructura adecuada a lo solicitado.
- Se hizo la parte de introducción, identificación del problema y la metodología de la propuesta.

Resultados y Observaciones

Gracias a la estructura brindada por la herramienta IA, se creó una propuesta de investigación completa, donde abarcase los puntos más adecuados. También, se creó de forma exitosa la parte de introducción, identificación del problema y metodología del documento. Esto tomando en cuenta un buen manejo del lenguaje técnico y comunicación de ideas.

09/05/2025

Encargados

Jose Barquero, Jose Edo Campos, Jimmy Feng y Alexander Montero

Objetivo

Entender dudas que se tienen del proyecto.

Procedimiento

- Los estudiantes le realizaron consultas al profesor, debido a una confusión surgida debido a las entradas del módulo de la FPGA y las salidas del mismo módulo.

Resultados y Observaciones

Las dudas fueron atendidas por el profesor, el cual aclaró todo y dejó más claro cómo es que funcionan las entradas y salidas del módulo a implementar en la FPGA.

09/05/2025

Encargados

Jose Edo Campos

Objetivo

Realizar un prototipo de la app mobile.

Procedimiento

- El estudiante se encargó en realizar una aplicación móvil, según las entradas que se deben enviar a la FPGA, es decir, la entrada del acumulado (4 bits), además de la operación a realizar.
- Se definió la estructura de la comunicación UART.

Resultados y Observaciones

Se obtuvo un prototipo de la app mobile que se puede apreciar en la Figura 1. Además, se definió una estructura a utilizar del protocolo UART, donde los últimos cuatro bits representan el acumulado, y los cuatro primeros bits representan el CLK, RESET y la operación a escoger.

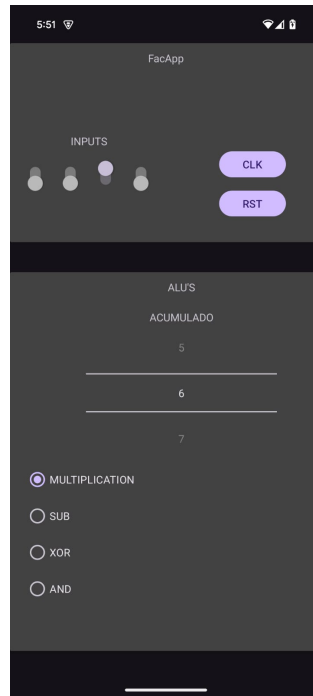


Figure 1: Prototipo visual de la aplicación Mobile

09/05/2025-10/05/2025

Encargados

Jose Barquero

Objetivo

Realizar un diseño de la ALU a implementar en la FPGA usando System Verilog.

Procedimiento

- El estudiante comprendió el comportamiento que deben tener las operaciones a realizar.
- Se diseñó según la especificación el realizar todo con compuertas la parte de un selector de operaciones, el cual se describió como un MUX 4:1.
- El estudiante realizó la operación de AND y XOR tomando en cuenta un concepto que se llama replicación para la entrada de dos bits[2] [3].
- El estudiante también realizó la operación de SUB tomando en cuenta que la resta se puede apreciar como una suma y también tomando en consideración que debe ser circular [4].

- El estudiante realizó una versión preliminar de la multiplicación, tomando en consideración los productos parciales y las sumas hechas por un full adder considerando carry [5] [6] [7]

Resultados y Observaciones

Se logró realizar un diseño preliminar de la ALU utilizando solo compuertas lógicas en base a las ecuaciones booleanas obtenidas de las tablas de verdad de cada parte de la ALU realizada.

En la Figura 2 se puede visualizar el diseño realizado para esta parte.

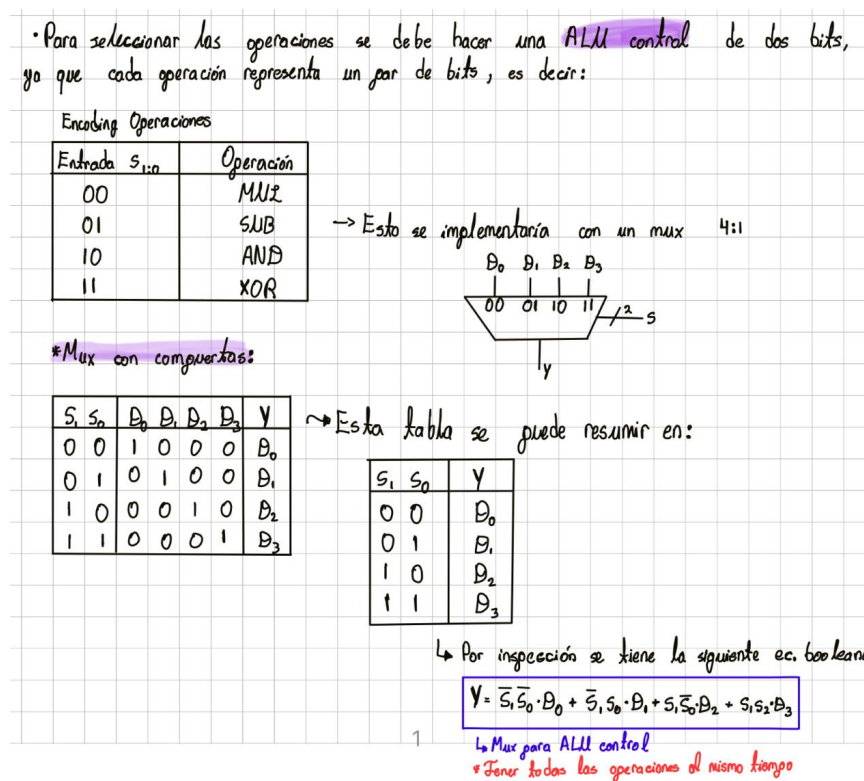


Figure 2: Diseño del multiplexor a implementar

En la Figura 3 se puede visualizar el diseño realizado para esta operación.

Esta operación como tal es una compuerta. La tabla de verdad se vería de la siguiente forma:						
A_3	A_2	A_1	A_0	B_1	B_0	Y
0	0	0	0	0	0	
0	0	0	1	0	1	
0	0	1	0	1	0	
0	0	1	1	1	1	
0	1	0	0	0	0	
0	1	0	1	0	1	
0	1	1	0	1	0	
0	1	1	1	1	1	
1	0	0	0	0	0	
1	0	0	1	0	1	
1	0	1	0	1	0	
1	0	1	1	1	1	
1	1	0	0	0	0	
1	1	0	1	0	1	
1	1	1	0	1	0	
1	1	1	1	1	1	

Esta forma no es la más adecuada

Usando como referencia:

"Logic and Computer Design fundamentals ed 5th"

"ALM Design - ECE 270"

Así, se puede ampliar el número de dos bits (B) a uno de 4 bits usando replicación, es decir:

$B = B_1 B_0 \Rightarrow B = \overset{B_1 \ B_1 \ B_1 \ B_1}{B_1 B_1 B_1 B_1} B_0 B_0$

↳ Representación de B en 4 bits

Esto se presta para operaciones simples y simétricas

Con esto, se obtendría lo siguiente:

$$Y_3 = A_3 B_1$$

$$Y_2 = A_2 B_1$$

$$Y_1 = A_1 B_0$$

$$Y_0 = A_1 B_0$$

Figure 3: Diseño de la operación AND a implementar

En la Figura 4 se puede visualizar el diseño realizado para esta operación.

* Lo XOR se puede implementar como:

A	B	Y
0	0	0
0	1	1
1	0	1
1	1	0

$\rightarrow Y = \bar{A}B + A\bar{B} = A \oplus B$

De forma similar al AND, se hará uso de la replicación para $B: B = B_1 B_1 B_0 B_0$, y se hará bit por bit. Entonces

Tabla Parcial XOR

A_3	A_2	A_1	A_0	B_1	B_0	Y_3	Y_2	Y_1	Y_0
0	0	0	0	0	0	0	0	0	0
1	0	1	0	0	0	1	0	0	0
1	1	0	1	1	0	0	0	0	1
1	1	1	1	1	1	0	0	0	0

Así:

$$Y_3 = A_3 \oplus B_1 = \bar{A}_3 B_1 + A_3 \bar{B}_1$$

$$Y_2 = A_2 \oplus B_1 = \bar{A}_2 B_1 + A_2 \bar{B}_1$$

$$Y_1 = A_1 \oplus B_0 = \bar{A}_1 B_0 + A_1 \bar{B}_0$$

$$Y_0 = A_0 \oplus B_0 = \bar{A}_0 B_0 + A_0 \bar{B}_0$$

$$Y_3 = \bar{A}_3 B_1 + A_3 \bar{B}_1$$

$$Y_2 = \bar{A}_2 B_1 + A_2 \bar{B}_1$$

$$Y_1 = \bar{A}_1 B_0 + A_1 \bar{B}_0$$

$$Y_0 = \bar{A}_0 B_0 + A_0 \bar{B}_0$$

Figure 4: Diseño de la operación XOR a implementar

En la Figura 5 se puede visualizar el diseño realizado para esta operación.

La resta se puede implementar de la siguiente forma:

$$A - B = A + \bar{B} + 1 \rightarrow \text{Utilizando complemento a base dos}$$

La entrada "B" de 2 bits se debe escribir en 4 bits. Entonces:

$$B = 00B_1B_2 \rightarrow \bar{B} = 11\bar{B}_1\bar{B}_2$$

Como la resta se puede escribir como una suma. Entonces:

Ejm:

C	B	Y	$\rightarrow Y = C \oplus B$
0	0	0	
0	1	1	
1	0	1	
1	1	0	Carry $\rightarrow C_{out} = C \cdot B$

Figure 5: Primera parte del diseño de la operación SUB a implementar

En la Figura 6 se puede visualizar el diseño realizado para esta operación.

Para esta implementación se sumará bit por bit y se considerará el acarreo

$$\begin{aligned} Y_0 &= A_0 \oplus \bar{B}_0 \\ Y_1 &= (A_1 \oplus \bar{B}_1) \oplus (A_0 \cdot \bar{B}_0) \\ Y_2 &= (A_2 \oplus 1) \oplus (A_1 \cdot \bar{B}_1) \\ Y_3 &= (A_3 \oplus 1) \oplus (A_2 \cdot 1) \end{aligned}$$

Pero, a este resultado debe sumarse 0001, es decir al valor Y_0 se le suma 1 y a los demás cero. Así:

$\begin{aligned} Y_0 &= A_0 \oplus \bar{B}_0 \oplus 1 \\ C_1 &= A_0 \cdot \bar{B}_0 \end{aligned}$
$\begin{aligned} Y_1 &= A_1 \oplus \bar{B}_1 \oplus C_1 \\ C_2 &= (A_1 \cdot \bar{B}_1) + (A_1 \cdot C_1) + (\bar{B}_1 \cdot C_1) \end{aligned}$
$\begin{aligned} Y_2 &= (A_2 \oplus 1) \oplus (A_1 \cdot \bar{B}_1) \\ C_3 &= A_2 \cdot C_2 \end{aligned}$
$Y_3 = A_3 \oplus 1 \oplus C_3$

Figure 6: Segunda parte del diseño de la operación SUB a implementar

En la Figura 7 se puede visualizar el diseño realizado para esta operación.

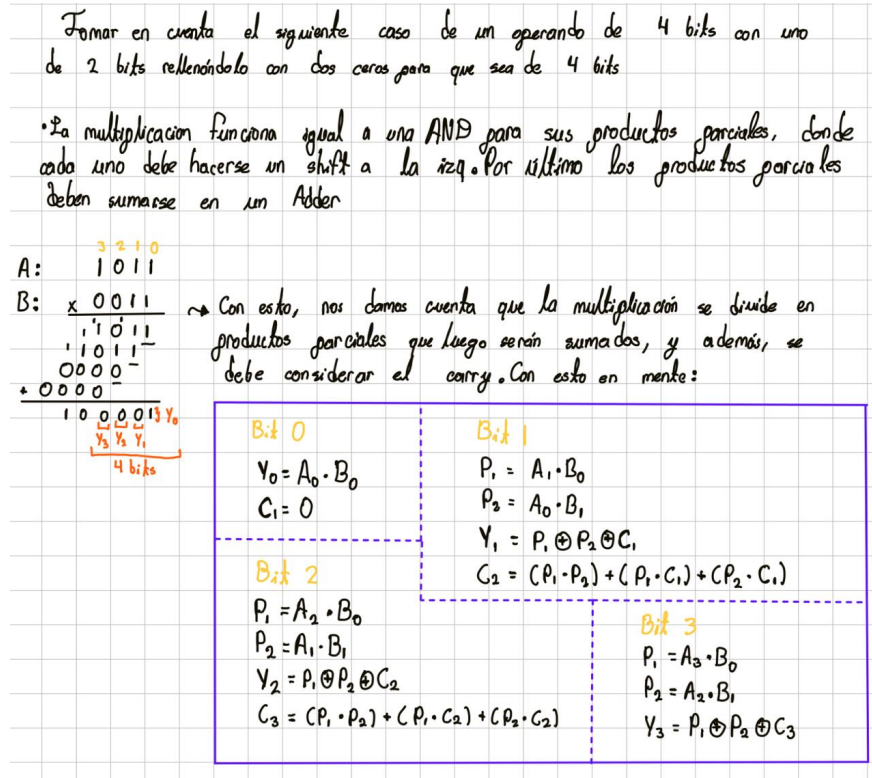


Figure 7: Diseño de la operación MUL a implementar

10/05/2025

Encargados

Jose Edo Campos

Objetivo

Mejorar la parte gráfica de la aplicación mobile.

Procedimiento

- Se consideró realizar unos cambios visuales en la interfaz gráfica de la app mobile, esto para que se viera de mejor forma.

Resultados y Observaciones

La aplicación mobile obtenida, presenta mejoras en su apartado gráfico. En la Figura 8 se puede apreciar este cambio realizado.

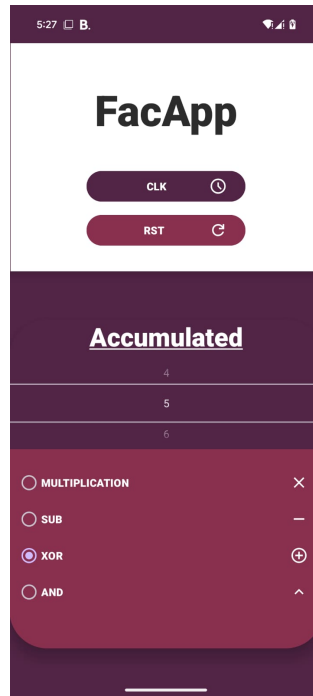


Figure 8: Interfaz gráfica de la app mobile mejorada

11/05/2025

Encargados

Jose Edo Campos y Jimmy Feng

Objetivo

Creación de la comunicación serial utilizando UART.

Procedimiento

- Se tomó como base un código utilizado para el curso de taller de diseño digital, donde la FPGA recibe los mensajes de un NodeMCU mediante el protocolo UART.
- Los datos que envía el NodeMCU son recibidos desde la app mobile.

Resultados y Observaciones

Se obtuvo una comunicación exitosa entre la aplicación mobile a la FPGA mediante el NodeMCU y usando la comunicación UART. Para las pruebas se utilizaron los LEDs propios de la FPGA para visualizar si los datos estaban llegando de forma correcta.

11/05/2025

Encargados

Jose Barquero, Jose Edo Campos y Alexander Montero

Objetivo

Mejoras en el módulo de la ALU.

Procedimiento

- Se realizaron pruebas en la FPGA del módulo de la ALU, esto para visualizar cómo estaba funcionando y si lo estaba haciendo de forma adecuada.
- Se notaron algunos errores respecto a los LEDs y el display de 7 segmentos.
- Se crearon los testbench de algunos módulos creados.
- Se realizó un cambio en cómo se hacían las operaciones AND y XOR, donde el operando B dejada de usar replicación y ahora solo sería tendría dos ceros en sus dos bits más significativos.

Resultados y Observaciones

A pesar de los errores que poseía el código de la ALU, se lograron corregir, así logrando que se mostrasen los resultados adecuados. También, la creación de los testbench permitió una mejor visualización de cómo estaba funcionando cada módulo.

12/05/2025

Encargados

Jose Edo Campos y Jimmy Feng

Objetivo

Implementación completa entre la app mobiel y el módulo de la ALU.

Procedimiento

- Se hizo la conexión entre estos dos módulos independientes para demostrar el funcionamiento del sistema.

Resultados y Observaciones

Se logró de manera exitosa la conexión entre la aplicación mobile y las salidas generadas de la ALU, según correspondiese.

12/05/2025

Encargados

Jose Barquero, y Alexander Montero

Objetivo

Creación del módulo PWM.

Procedimiento

- Se tomó en cuenta la definición de lo que es PWM para la implementación en system verilog sobre este módulo [8] [9].
- Con el conocimiento adquirido, se realizó la implementación en el HDL sobre este módulo en la FPGA, tomando en cuenta el uso de solo compuertas lógicas y haciendo las implementaciones de forma estructural [10] [11] [12].

Resultados y Observaciones

Se logró crear de manera exitosa los módulos correspondientes al módulo de PWM, donde se encuentra un contador y un comparador, esto con el fin de hacer modular y estructural el código.

13/05/2025-15/05/2025

Encargados

Jose Barquero, Jose Edo Campos, Jimmy Feng y Alexander Montero

Objetivo

Realizar ajustes en todo lo realizado en el proyecto.

Procedimiento

- Se realizaron cambios en la lógica de los contadores, esto debido a que no eran estructurales y no usaban compuertas, si no estructuras de alto nivel.
- Se creó el circuito de desacople para la parte del motor, esto tomando en cuenta que las fotoresistencias entregan valores analógicos y se deben convertir a digital.
- Se mejoró un poco la comunicación UART.
- Se realizaron cambios en la comunicación, específicamente en al FPGA para hacerlo de forma estructural y usando una FSM, sin embargo esta implementación no resultó de buena forma.

Resultados y Observaciones

El desacople eléctrico funciona hasta cierto punto de buena forma, no obstante no es tan notorio los cambios según el resultado de la ALU. Los cambios hechos para ser de forma estructural todos los módulos, no resultaron en su mayoría, específicamente para la comunicación, pero para lo demás si funcionó de forma adecuada.

16/05/2025

Encargados

Jimmy Feng

Objetivo

Mejoras y refactor de la comunicación UART.

Procedimiento

- Se realizó unos cambios en el módulo UART para que cumpliera lo solicitado en el proyecto.
- La comunicación UART ahora solo trabaja con compuertas y utiliza una FSM con sus respectivas ecuaciones booleanas para las transiciones y salidas.

Resultados y Observaciones

El resultado obtenido de este refactor del módulo UART funcionó de buena forma, además de la implementación de la FSM para tener un mejor flujo y control durante la comunicación permitió tener una mejora en la comunicación UART.

Tabla de transición de estados:

S1	S0	rx	ch	cf	b7	NS1	NS0
0	0	0	X	X	X	0	1
0	0	1	X	X	X	0	0
0	1	0	0	X	X	0	1
0	1	0	1	X	X	1	0
0	1	1	X	X	X	0	1
1	0	X	X	0	X	1	0
1	0	X	X	1	0	1	0
1	0	X	X	1	1	1	1
1	1	X	X	0	X	1	1
1	1	X	X	1	X	0	0

Table 1: Tabla de transición de estados para la FSM UART

Tabla de salidas:

S1	S0	rx	ch	cf	b7	LD	IC	RC	IB	RB	DV	CDR
0	0	0	X	X	X	0	0	1	0	0	0	1
0	0	1	X	X	X	0	0	0	0	0	0	1
0	1	0	0	X	X	0	1	0	0	0	0	0
0	1	0	1	X	X	0	0	1	0	1	0	0
0	1	1	X	X	X	0	0	0	0	0	0	0
1	0	X	X	0	X	0	1	0	0	0	0	0
1	0	X	X	1	0	1	0	1	1	0	0	0
1	0	X	X	1	1	1	0	1	0	0	0	0
1	1	X	X	0	X	0	1	0	0	0	0	0
1	1	X	X	1	X	0	0	1	0	1	1	0

Table 2: Tabla de señales de salida de la FSM UART

Ecuaciones de transición de estados:

$$\begin{aligned}
 next_state_1 = & (\overline{s1} \cdot s0 \cdot \overline{rx} \cdot ch) + \\
 & (s1 \cdot \overline{s0} \cdot cf \cdot \overline{b7}) + \\
 & (s1 \cdot \overline{s0} \cdot cf \cdot b7) + \\
 & (s1 \cdot \overline{s0} \cdot \overline{cf}) + \\
 & (s1 \cdot s0 \cdot \overline{cf})
 \end{aligned}$$

$$\begin{aligned}
 next_state_0 = & (\overline{s1} \cdot \overline{s0} \cdot \overline{rx}) + \\
 & (\overline{s1} \cdot s0 \cdot \overline{rx} \cdot \overline{ch}) + \\
 & (s1 \cdot \overline{s0} \cdot cf \cdot b7) + \\
 & (s1 \cdot s0 \cdot \overline{cf})
 \end{aligned}$$

Ecuaciones de salidas:

$$load_data = (s1 \cdot \overline{s0} \cdot cf \cdot \overline{b7}) + (s1 \cdot \overline{s0} \cdot cf \cdot b7)$$

$$inc_clk = (\overline{s1} \cdot s0 \cdot \overline{rx} \cdot \overline{ch}) + (s1 \cdot \overline{s0} \cdot \overline{cf}) + (s1 \cdot s0 \cdot \overline{cf})$$

$$\begin{aligned}
 rst_clk = & (\overline{s1} \cdot \overline{s0} \cdot \overline{rx}) + (\overline{s1} \cdot s0 \cdot \overline{rx} \cdot ch) + (s1 \cdot \overline{s0} \cdot cf \cdot \overline{b7}) + \\
 & (s1 \cdot \overline{s0} \cdot cf \cdot b7) + (s1 \cdot s0 \cdot cf)
 \end{aligned}$$

$$inc_bit = (s1 \cdot \overline{s0} \cdot cf \cdot \overline{b7})$$

$$rst_bit = (\overline{s1} \cdot s0 \cdot \overline{rx} \cdot ch) + (s1 \cdot s0 \cdot cf)$$

$$data_valid = (s1 \cdot s0 \cdot cf)$$

$$clr_data_ready = (\overline{s1} \cdot \overline{s0} \cdot \overline{rx}) + (\overline{s1} \cdot \overline{s0} \cdot rx);$$

17/05/2025

Encargados

Jose Edo Campos

Objetivo

Hacer cambios en los módulos para que cumplan con lo solicitado en el proyecto.

Procedimiento

- En base a las especificaciones del proyecto, se realizaron algunos cambios en módulos que aún estaban usando estructuras de alto nivel.
- Se realizó la implementación estructural de cada módulo a cambiar, y se validó que funcionara de forma adecuada.

Resultados y Observaciones

La implementación estructural funcionó de forma adecuada para estos módulos que se realizaron cambios.

17/05/2025

Encargados

Jose Barquero, Jose Edo Campos, Jimmy Feng y Alexander Montero

Objetivo

Finalizar con toda la documentación del proyecto.

Procedimiento

- Se realizó la propuesta de investigación del proyecto y se finalizó con la bitácora del mismo.

Resultados y Observaciones

Se terminó la propuesta de investigación en base a lo solicitado en el proyecto, y también se finalizó la bitácora del mismo.

Referencias

- [1] Bhimsen, *7 Segment HEX Decoder*, <https://electronics-fun.com/7-segment-hex-decoder/>, Digital Electronics, n.d. (visitado 02-05-2025).
- [2] D. of Electrical y U. o. M. Computer Engineering, *ALU Design - ECE 270*, Online lecture notes, 2020. dirección: https://engineering.purdue.edu/~meyer/DDU270/Notes/PDF/2-Mod4_NQ_2019.pdf.
- [3] M. M. Mano, C. R. Kime y T. Martin, *Logic and Computer Design Fundamentals*, 5th. Pearson, 2017, ISBN: 9780131989269. dirección: [https://wp.kntu.ac.ir/dfard/ebook/lc_ds1/M.%20Morris%20R.%20Mano,%20Charles%20R.%20Kime,%20Tom%20Martin%20-%20Logic%20and%20computer%20design%20fundamentals-Prentice%20Hall%20\(2015\).pdf](https://wp.kntu.ac.ir/dfard/ebook/lc_ds1/M.%20Morris%20R.%20Mano,%20Charles%20R.%20Kime,%20Tom%20Martin%20-%20Logic%20and%20computer%20design%20fundamentals-Prentice%20Hall%20(2015).pdf).
- [4] D. Harris y S. Harris, *Digital Design and Computer Architecture*, 2ª ed. Morgan Kaufmann, 2015. dirección: <https://www.elsevier.com/books/digital-design-and-computer-architecture/harris/978-0-12-800056-4>.
- [5] GeeksforGeeks, *Sequential Binary Multiplier*, Consultado en mayo de 2025, 2024. dirección: <https://www.geeksforgeeks.org/sequential-binary-multiplier/>.
- [6] N. Matloff, *Digital Design for Multiplication*, University of California, Davis. Consultado en mayo de 2025, 2003. dirección: <https://heather.cs.ucdavis.edu/Mult.pdf>.
- [7] RealDigital, *Multipliers - Arithmetic Circuits*, Consultado en mayo de 2025, s.f. dirección: <https://www.realdigital.org/doc/42f40f5d5d502c3672477a2bbd78b697>.
- [8] S. Brown y Z. Vranesic, *Fundamentals of Digital Logic with Verilog Design*, 3rd. McGraw-Hill Education, 2012, Chapter on Sequential Circuits and FSMs relevant for PWM design, ISBN: 9780073380544.
- [9] Xilinx Inc., *Pulse Width Modulation (PWM) Tutorial using Vivado and Verilog*, Xilinx, 2021.
- [10] Intel Corporation, *PWM Motor Control with FPGAs*, <https://www.intel.com>, Application Note describing PWM motor control with FPGAs and Quartus, 2018.
- [11] D. D. Vachhani y R. P. Deshmukh, "PWM based Speed Control of DC Motor using FPGA," en *2011 International Conference on Emerging Trends in Electrical and Computer Technology*, IEEE, 2011, págs. 626-629. DOI: [10.1109/ICETECT.2011.5760144](https://doi.org/10.1109/ICETECT.2011.5760144).
- [12] S. Ghosh y A. Prakash, "Design and Implementation of PWM Controller Using FPGA," *International Journal of Engineering Trends and Technology (IJETT)*, vol. 4, n° 9, págs. 4021-4024, 2013, ISSN: 2231-5381.