Universidad
Politécnica
de Madrid

Programación
Hardware
Reconfigurable

Escuela Técnica Superior
de Ingeniería de
Sistemas Informáticos

PROYECTO FINAL DE LA ASIGNATURA

Cifrado Afín con FPGA

(CAFPGA)

Nombre del equipo: PHR22-M-08

Espacio de trabajo (sharepoint) del grupo.

Miembros del equipo:

Líder	Alumno
X	Mariano Ulloa, Joao Sebastian
	Butsa, Zhanna
	Dotor Ruiz, Joaquín
	Quiñonero González, Alberto

ÍNDICE

ÍNDICE DE FIGURAS

ÍNDICE DE TABLAS

OBJETIVO

Vamos a implementar el funcionamiento de un sistema de cifrado y descifrado afín en una FPGA con el objetivo de trasladar una base de nuestras carreras a un ejemplo físico. Pretendemos que esta herramienta sea usada tanto por docentes como por alumnos de nuevo ingreso para facilitar la comprensión de dicho cifrado.

Alumnos de Álgebra, Fundamentos de Computadores y Fundamentos de Seguridad verán de primera mano que el cifrado afín, aunque ya en desuso, sigue siendo útil con fines introductorios a la enseñanza de la criptografía además de ser uno de sus primeros contactos con dispositivos programables. También, en la segunda parte del proyecto, se desarrollará una funcionalidad wifi para controlar el sistema a través de dispositivos móviles.

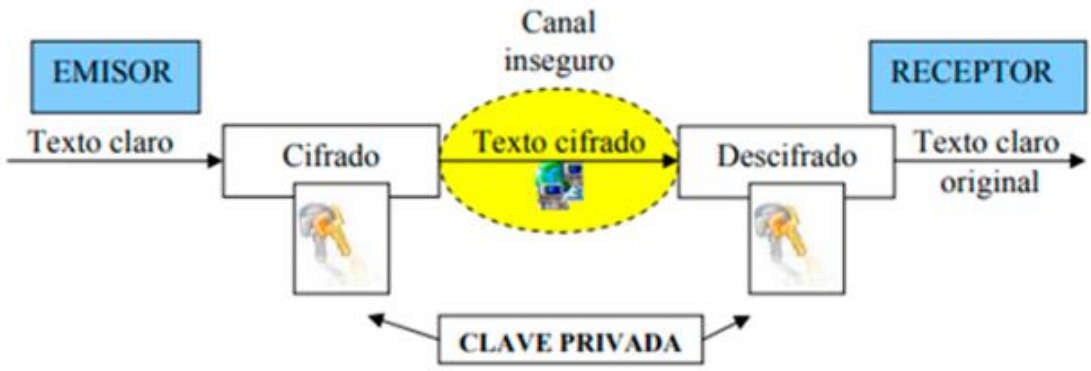

Figura 1: Ejemplo cifrado

1. MODIFICACIÓN OBJETIVO

El proyecto no ha sufrido modificaciones en cuanto al objetivo. La idea principal descrita originalmente es la misma con la que se ha ido trabajando aunque sí se experimentaron cambios en otros aspectos.

INTRODUCCIÓN

Este proyecto realiza el cifrado y descifrado afín de tal forma que mediante un teclado se introducen los datos y con displays de 7 segmentos se muestra el resultado de la operación. Este proyecto se ha realizado con el objetivo de facilitar el aprendizaje de este cifrado con un ejemplo práctico, está enfocado a que las futuras generaciones vean una cohesión entre asignaturas como Fundamentos de Computadores, Álgebra y Fundamentos de Seguridad.

El único proyecto encontrado con un cierto parecido al nuestro es el proyecto realizado en el año 2019-2020 titulado como Cifrado de bloque mediante puertas lógicas, lo cual nos parece una idea excelente si tienes el objetivo de añadir un mayor nivel de seguridad a tus dispositivos, pero nuestro proyecto no tiene exactamente esa función, por lo que para el objetivo para el cual nosotros realizamos nuestro proyecto pensamos que nuestro teclado de cifrado y descifrado afín es más útil y práctico.

En la siguiente figura se muestra el proceso a seguir para obtener un mensaje cifrado haciendo uso del cifrado afín y la operación inversa para descifrarlo.

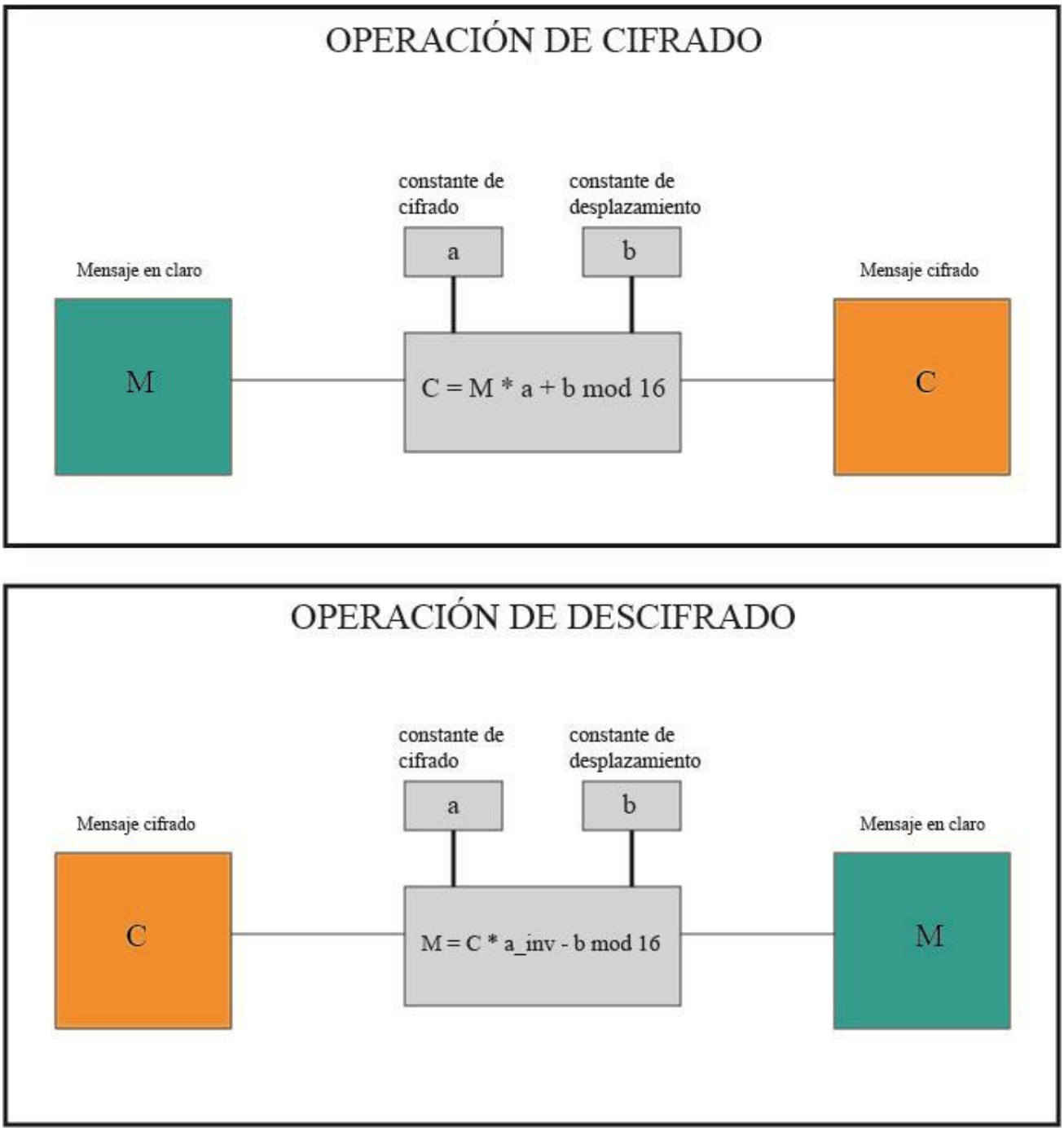

Figura 2: Cifrado afín (Autor Joaquín Dotor)

DEFINICIÓN DEL PROBLEMA

El problema identificado es la tasa de abandono de estudiantes sobre todo en etapas tempranas de la carrera. Estos son datos de las tres asignaturas con las que más relación tiene nuestro proyecto.

Tasas de rendimiento en convocatoria ordinaria:

-Fundamentos de Computadores

Convocatoria ordinaria

Curso	Nº de alumnos matriculados	Nº de alumnos aprobados	Tasa de rendimiento (%)
2020-21	177	33	18.64
2019-20	190	63	33.16
2018-19	188	24	12.77
2017-18	179	27	15.08
2016-17	150	16	10.67

Figura 3: Tasa de rendimiento en Fundamentos de Computadores en convocatoria ordinaria (UPM)

-Álgebra

Convocatoria ordinaria

Curso	Nº de alumnos matriculados	Nº de alumnos aprobados	Tasa de rendimiento (%)
2019-20	152	72	47.37
2018-19	164	55	33.54
2017-18	152	58	38.16
2016-17	121	39	32.23
2015-16	123	43	34.96

Figura 4: Tasa de rendimiento en Álgebra en convocatoria ordinaria (UPM)

-Fundamentos de Seguridad

Convocatoria ordinaria

Curso	Nº de alumnos matriculados	Nº de alumnos aprobados	Tasa de rendimiento (%)
2019-20	123	91	73.98
2018-19	138	78	56.52
2017-18	134	68	50.75
2016-17	112	60	53.57
2015-16	139	55	39.57

Figura 5: Tasa de rendimiento en Fundamentos de Seguridad en convocatoria ordinaria (UPM)

Tasas de absentismo en convocatoria ordinaria:

-Fundamentos de Computadores

Convocatoria ordinaria

Curso	Nº de alumnos matriculados	Nº de alumnos no presentados	Tasa de absentismo (%)
2020-21	177	0	0.00
2019-20	190	10	5.26
2018-19	188	55	29.26
2017-18	179	37	20.67
2016-17	150	39	26.00

Figura 6: Tasa de absentismo en Fundamentos de Computadores en convocatoria ordinaria (UPM)

-Álgebra

Convocatoria ordinaria

Curso	Nº de alumnos matriculados	Nº de alumnos no presentados	Tasa de absentismo (%)
2019-20	152	30	19.74
2018-19	164	31	18.90
2017-18	152	49	32.24
2016-17	121	36	29.75
2015-16	123	51	41.46

Figura 7: Tasa de absentismo en Álgebra en convocatoria ordinaria (UPM)

-Fundamentos de Seguridad

Convocatoria ordinaria

Curso	Nº de alumnos matriculados	Nº de alumnos no presentados	Tasa de absentismo (%)
2019-20	123	4	3.25
2018-19	138	19	13.77
2017-18	134	22	16.42
2016-17	112	34	30.36
2015-16	139	17	12.23

Figura 8: Tasa de absentismo en Fundamentos de Seguridad en convocatoria ordinaria (UPM)

Estos datos son recogidos de la plataforma GAUSS de la Universidad Politécnica de Madrid.

Se buscará disminuir o ayudar a mantener estos porcentajes y su tendencia.

DISEÑO DE LA SOLUCIÓN PROPUESTA

Para ayudar a ilustrar a nuevos alumnos hemos realizado un sistema de displays y teclado que cifra y descifra mediante cifrado afín. Herramienta con la que podrán interactuar siempre que quieran.

Los hitos que hemos utilizado para realizar dicho sistema son los siguientes:

Hito 1: Realización del algoritmo de cifrado mediante puertas lógicas.

Hito 2 : Realización del algoritmo de descifrado mediante puertas lógicas.

Hito 3: Implementación en VHDL.

Hito 4: Generar una conexión de entrada a través del teclado hexadecimal.

Hito 5: Generar una conexión de salida cifrada a través de pantalla oled.

Hito 6 : Estudiar el componente ESP-32.

Hito 7: Lograr la conexión entre la ESP-32 para recepción de datos.

Hito 8 : Lograr la conexión entre la ESP-32 para el envío de datos.

Hito 9: Comprobación y testeo del total funcionamiento del sistema.

El proyecto es similar a una aplicación desarrollada por Danil Tkachenko para la universidad checa Tomas Bata University. Affine Cipher fue lanzada a Google Play Store el 22 de septiembre de 2021. La principal diferencia entre la aplicación del señor Tkachenko y el proyecto aquí desarrollado es la implementación hardware.

A continuación una imagen de Affine Cipher:

Figura 9: Affine Cipher (Danil Tkachenko)

HARDWARE EMPLEADO

Para el desarrollo de este proyecto se han empleado diferentes componentes:

Figura 10: FPGA Basys Artix-7 (Digilent, 2022)

La FPGA fue provista por la asignatura. Además de cargarle el programa y utilizar sus puertos de entrada y salida de datos, utilizamos un switch para diferenciar entre cifrado y descifrado, otro para indicar que la última tecla pulsada hay que guardarla en el registro que corresponda a continuación y otro para activar el proceso de operar con las teclas mensaje, dos LEDS, uno para indicar si se cifra o descifra y otro para indicar si el proceso se ha completado, el puerto PMOD A para conectar el teclado, el puerto PMOD B para conectar la salida de tecla modificada a los displays y el puerto PMOD C para la salida del decodificador 3 a 8 que es la que activa a nivel bajo la pantalla que recibe señal. También se usaron los displays de la FPGA para hacer pruebas en la implementación física.

Figura 11: Pmod KYBD: teclado 16 botones (Digilent, 2022)

Este componente sirve para introducir la cadena a cifrar/descifrar así como los valores necesarios para las operaciones.

Figura 12: Pantalla OLED Digilent Activa matrix 128 x 32 pixels SPI Interface (Digilent, 2022)

La pantalla OLED se iba a utilizar para la representación del resultado. Debido a la complejidad que suponía se optó por otra ruta que se estimó más sencilla.

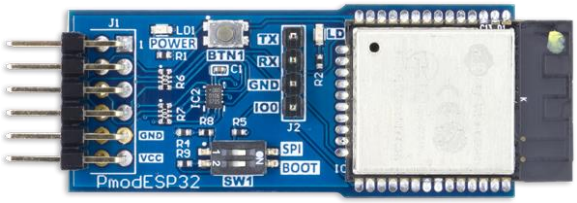

Figura 13: Pmod ESP32: Wireless Communication Module (Digilent, 2022)

El ESP32 corresponde a la segunda parte del proyecto. Su inclusión estaba en preparación pero no se incluyó por falta de tiempo y se prescindió de su uso.

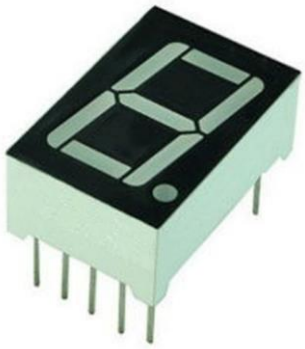

Figura 14: Display LED 7 segmentos

Figura 15: Resistencia 1/4W 1K Ohmios

Estos componentes se usaron como alternativa a la pantalla OLED. Con los displays se representa el resultado de las operaciones y se usaron resistencias para regular la corriente recibida por los displays.

HERRRAMIENTAS SOFTWARE EMPLEADAS

Se ha utilizado vivado para la programación completa del circuito, simulación e implementación en FPGA y Adobe Illustrator para elaborar una explicación del funcionamiento del cifrado afín y un esquema del circuito que nos facilite la implementación y que sea más visual.

DESARROLLO SOFTWARE REALIZADO

Con esta ilustración del circuito simplificado, el desarrollo del software es mucho más visual y fácil de comprender. Para ver con más detalle acceder al siguiente enlace: Ilustración.pdf

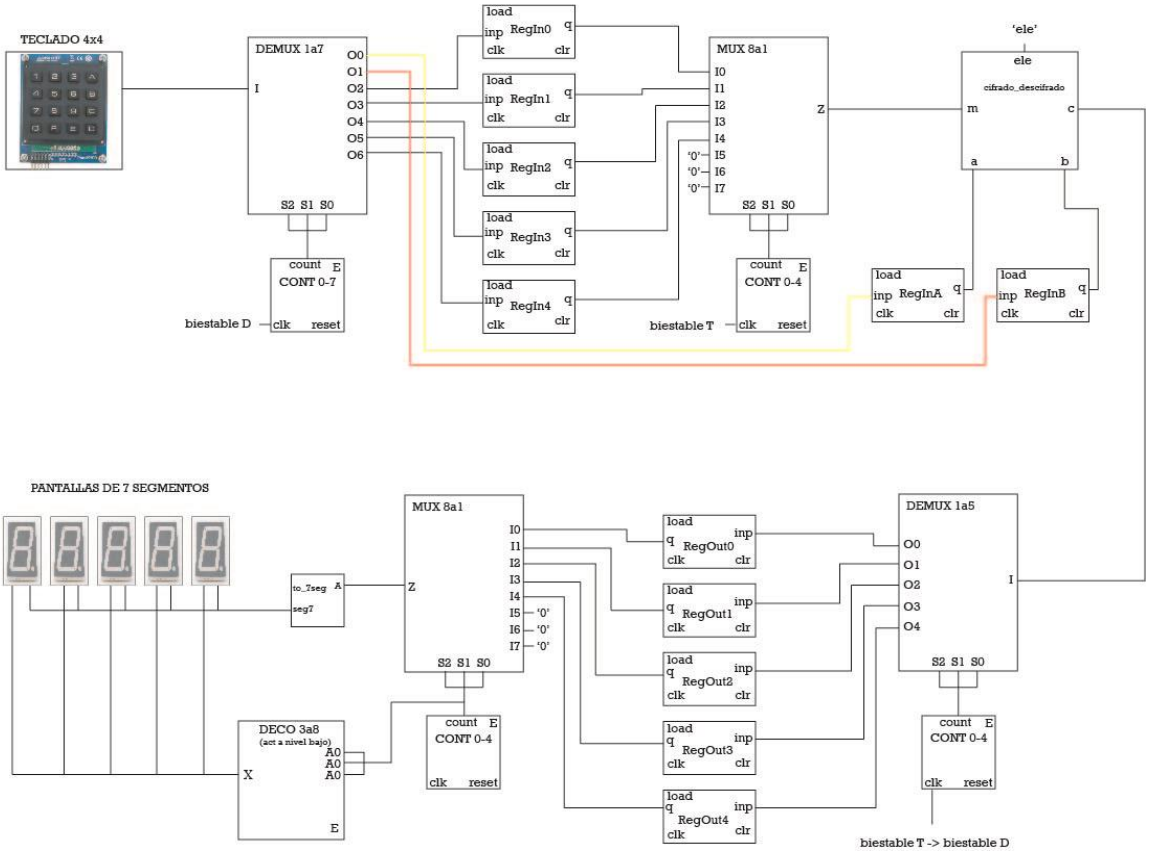

Figura 16: Ilustración circuito master. (Autor Joaquín Dotor)

El algoritmo se basa en el uso de componentes básicos conectados para conseguir el objetivo que se plantea.

El primer componente en nuestro proyecto es el teclado, cuyos puertos son, entrada de reloj, entrada de fila (4 bits), salida de columna (4 bits) y tecla, que es la tecla pulsada.

- Para el funcionamiento del teclado hacemos uso de otro componente que recibe la fila y la columna y va alimentando las columnas para obtener cuál es la tecla pulsada recibiendo la fila. Devuelve la tecla pulsada.
- El circuito general es el más grande, que alberga la funcionalidad principal del cifrado. Sus puertos son, switches de reset, activar, elegir (cif o descif) y pulsación (registrar una tecla), leds que indican si se cifra o descifra y si se ha finalizado el

proceso, entrada de reloj, 4 bits de entrada que indican la tecla pulsada y 5 vectores de 4 bits para las salidas de tecla ya modificada.

El funcionamiento de este circuito consiste en que un demultiplexor recibe las teclas pulsadas y un contador de 0 a 6 va contando cada vez que se activa y desactiva el switch de pulsación, con lo que conseguimos que la tecla pase a los registros de entrada y de ellos, los que son mensaje pasan a un multiplexor que los va enviando en orden a el circuito que cifra o descifra, cuyas entradas son a (constante de cifrado), b (constante de desplazamiento) y el switch de elección.

Las teclas ya modificadas pasan a un demultiplexor que las va guardando en los registros de salida, pasan a un multiplexor cuya salida está conectada a un
conversor de hexadecimal a 7 segmentos el cual tiene conectada su salida a las 5 pantallas de 7 segmentos, alimentadas con un decodificador activo a nivel bajo.

- El funcionamiento de las pantallas de 7 segmentos (recibe vector de 4 bits con la tecla y devuelve un vector de 7 bits que son los segmentos a encender) se hace con los componentes ya mencionados conversor de hexadecimal a 7 segmentos, con el que conseguimos representar una tecla en 7 segmentos, y decodificador 3 a 8, con el que activamos la pantalla que va a recibir la señal y la representará.

Durante la programación del teclado surgió un inconveniente:

Al pulsar una tecla, el teclado manda señal constantemente de la última tecla pulsada, por lo que no existe un valor que indique que no se está pulsando ninguna tecla. Para solucionar este inconveniente se añadió un switch que su función es incrementar el contador que gestiona los registros que almacenan las teclas, funcionando de manera que cada vez que se activa y desactiva el switch, la última tecla pulsada se almacena en el registro correspondiente.

Otro problema se debía a que todos los contadores menos el del primer demultiplexor al estar su enable conectados al switch de activación, necesitábamos un pequeño delay para que diera tiempo a que la señal estuviera disponible en los siguientes componentes, por lo que se tuvieron que añadir biestables.

A parte de esos, hubo pequeños problemas (los load de los registros, para almacenar un dato en el momento indicado, algún proceso que necesitaba una variable extra a evaluar para el correcto funcionamiento, etc.).

PRUEBAS Y TESTS

Simulación del circuito general cifrando, que recibe las teclas y las devuelve modificadas.

En la siguiente figura hay que observar la entrada input, que será F, 1, F, A, C, 0, B.

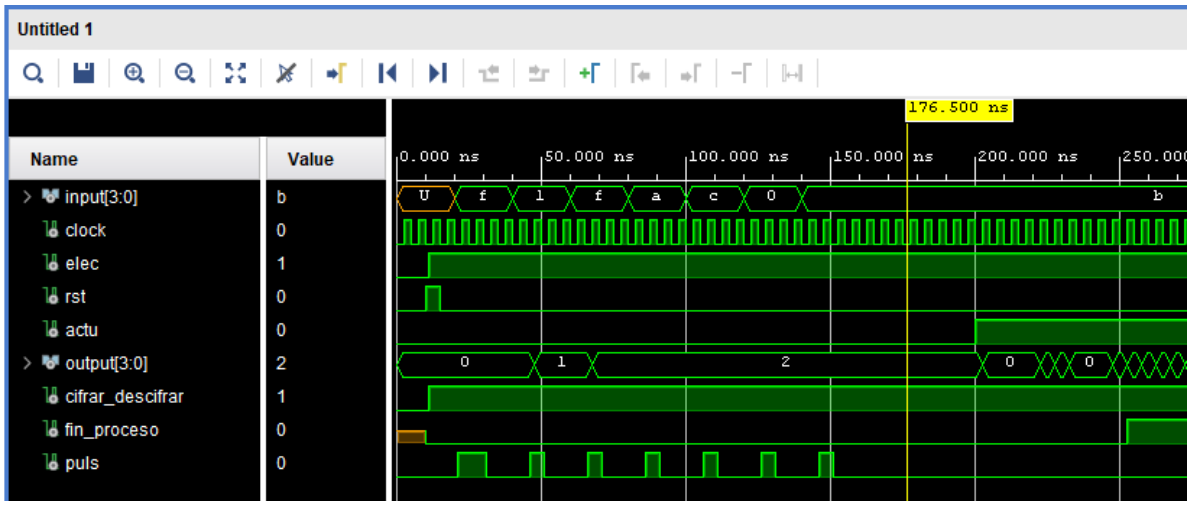

Figura 17: Simulación general cifrado, entradas. (Simulación vivado)

En la siguiente figura hay que observar que en la salida output, saldrán las teclas mensaje cifradas a partir del instante en que actu se active (2, 7, 5, 1, 6). La salida saldrá en bucle.

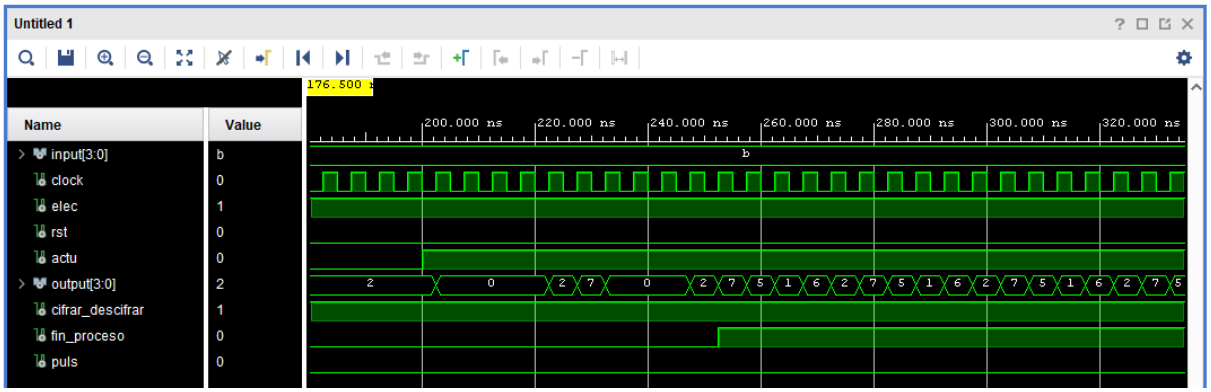

Figura 18: Simulación general cifrado, salidas. (Simulación vivado)

Simulación del circuito general descifrando, que recibe las teclas y las devuelve modificadas.

En la siguiente figura hay que observar la entrada input, que será 3, 2, 1, A, 4, C, F.

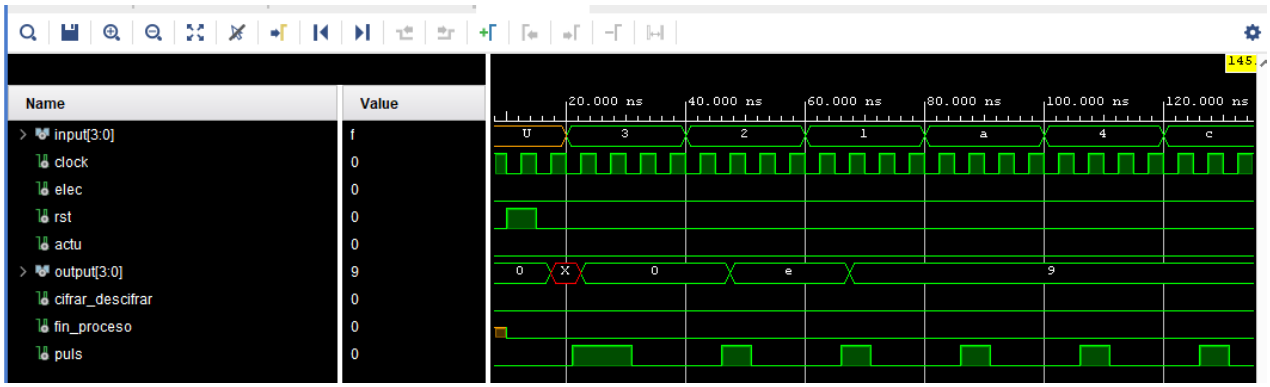

Figura 19: Simulación general descifrado, entradas. (Simulación vivado)

En la siguiente figura hay que observar que en la salida output, saldrán las teclas mensaje descifradas a partir del instante en que actu se active (). La salida saldrá en bucle.

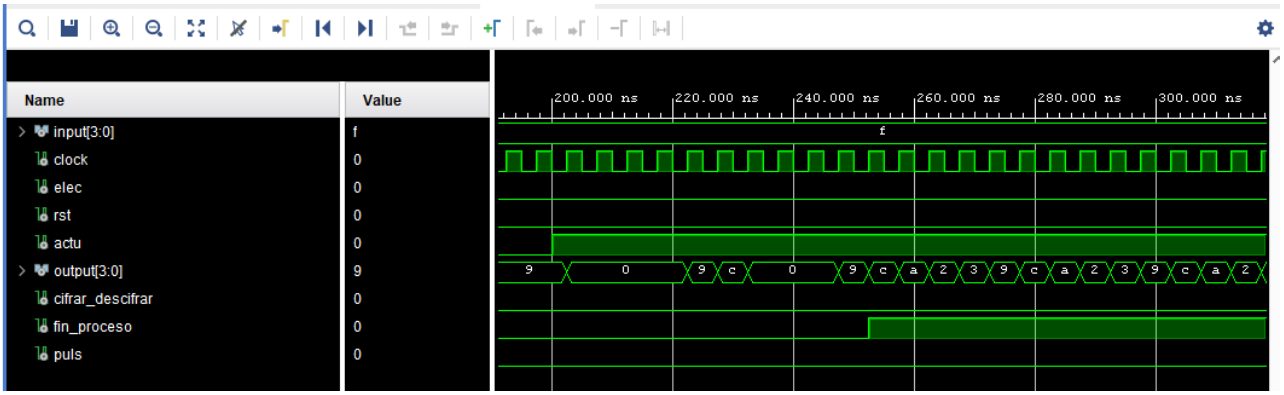

Figura 20: Simulación general descifrado, entradas. (Simulación vivado)

Simulación de la pantalla de 7 segmentos, que recibe la tecla en hexadecimal y la devuelve en formato 7 segmentos.

En la siguiente figura hay que observar que el componente recibe números hexadecimales y devuelve un vector de 7 bits indicando los segmentos que deben encenderse (activo a nivel bajo).

Ejemplos:

Recibe '0' devuelve 40 que en binario es (0)100 0000 y significa encender todos los segmentos excepto el número 7, que en el caso de '0' es el segmento del centro.

Recibe '4' devuelve 19 que en binario es (0)001 1001 y significa encender 2º 3º 6º y 7º segmentos.

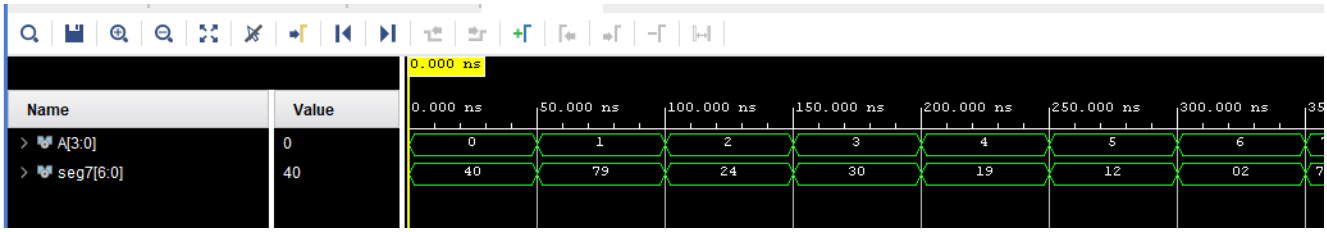

Figura 21: Simulación pantalla 7 segmentos. (Simulación vivado)

Para las pruebas físicas se probó el código del teclado conectándolo al puerto PMOD A de la FPGA y probando sus teclas como salida de 4 leds de la placa, comprobando así su correcto funcionamiento. Con las pantallas de 7 segmentos se hizo el mismo procedimiento, ambos experimentos fueron un éxito.

Más tarde se hicieron pruebas con el funcionamiento completo del circuito, no consiguiendo los resultados esperados.

PLANIFICACIÓN Y COSTES

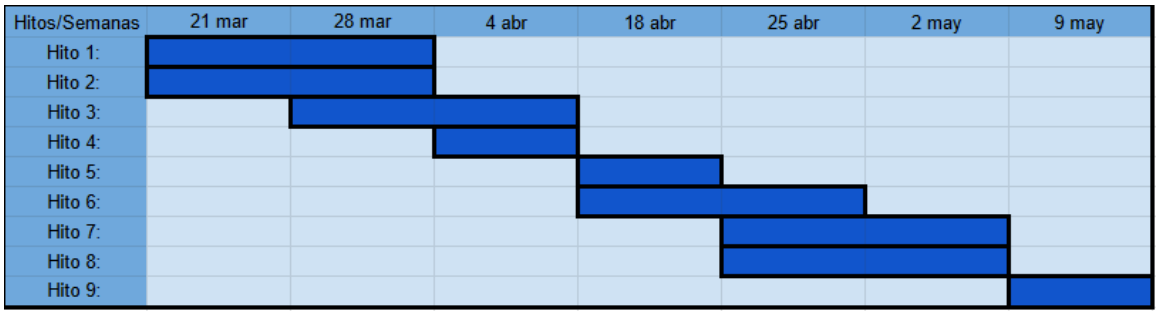

Figura 22: Diagrama de Gantt inicial

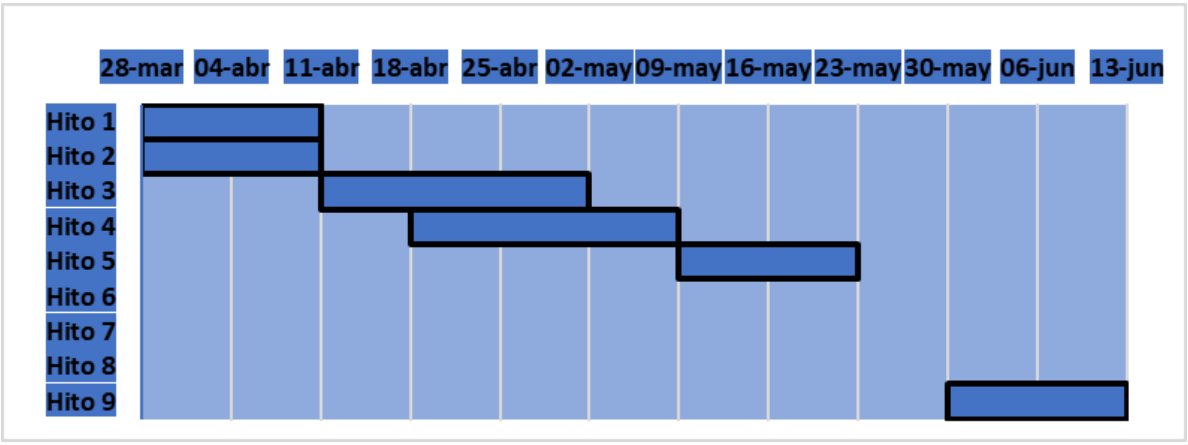

Figura 23: Diagrama de Gantt modificado

En relación a los hitos que hemos hecho hay que mencionar que algunos han sido modificados y otros eliminados, por diversas razones.

Los hitos 6, 7 y 8 se eliminaron, ya que por problemas de tiempo y la dificultad que esto suponía no pudimos añadir la ESP-32, además el hito 5 se comenzó pero no se finalizó, ya qué decidimos descartar la pantalla OLED y añadir los 5 displays.

La distribución final del tiempo en la realización del trabajo fue la siguiente:

Semana 1: esta semana nos dedicamos a entender y planear cómo íbamos a hacer nuestro proyecto, cada uno de nosotros proponía diversas ideas por lo que decidimos mezclar un poco y sacar una idea conjunta hacia delante.

Semana 2: En la segunda semana decidimos dividirnos el trabajo, mientras dos elaboraban el código de cifrado y los otros dos el código de descifrado.

Semana 3: En la tercera semana el código de cifrado ya estaba programado y listo el circuito que cifra, con el código de descifrado tuvimos dificultades y en esta semana nos dedicamos a resolver los errores surgidos.

Semana 4: En esta semana ya resolvimos los problemas del código de descifrado, y nos pusimos a elaborar la memoria, tuvimos bastantes problemas porque no sabíamos cómo

hacerla por lo que primero decidimos unir el código de descifrado y cifrado en un mismo proyecto de VHDL, para así tener un mini menú donde se pueda elegir mediante un switch cifrar o descifrar.

Semana 5 y 6: Implementación del teclado.

Semana 7 y 8: Finalmente terminamos de incorporar el teclado y nos pusimos a diseñar la memoria, fue un trabajo duro.

Semana 9: Finalmente comenzamos a trabajar con los displays ya que decidimos por la falta de tiempo no hacer uso de la pantalla OLED y tampoco incorporar la conexión WIFI.

Semana 10: Estuvimos trabajando en los fallos y errores que nos daba el código. Compramos más displays y resistencias porque 2 de los displays se estropearon.

Semana 11 y 12: Estas semanas las dedicaremos a resolver los errores de código y a ver si funcionaba todo bien.

Componente	Precio Unidad	Cantidad	Total
Placa de desarrollo FPGA Basys Artix-7 de Digilent	172,24 €	1	172,24 €
Pmod KYPD: teclado de 16 botones	14,50 €	1	14,50 €
Display LED 7 segmentos ánodo común	1,80 €	5	9€
Resistencia 1/4W 1K Ohmios (10 piezas)	0,14 €	1	0,14€
TOTAL PRESUPUESTO PROYECTO:			195,88 €

Tabla 1: Costes materiales del prototipo desarrollado.

ASPECTOS SOCIALES, AMBIENTALES, ÉTICOS Y LEGALES

El principal aspecto social que se busca tocar es el de la educación y motivación de los alumnos de nuevo ingreso.

CONCLUSIONES

Con esto además de aprender cómo sacar un proyecto hacia adelante y trabajar en equipo, se ve que todo esfuerzo tiene su recompensa y ha sido bastante duro pero gratificante ya que empiezas a confiar en ti mismo y a estar orgulloso de cualquier avance.

LÍNEAS FUTURAS

Sería posible incluir cualquier tipo de cifrado más complejo en este mismo proyecto y ampliar el tamaño del mensaje introducido dado que este proyecto es altamente escalable.

En cualquier caso, implementar una conexión WIFI con el Pmod ESP32 para enviar y recibir datos a través de un dispositivo móvil es el primer paso para expandir el proyecto.

REFERENCIAS

GAUSS UPM. (18 de 06 de 2022). Obtenido de https://upm.es/gauss

DIGILENT. (18 de 06 de 2022). Obtenido de Pmod ESP32: Wireless Communication Module: https://store.digilentinc.com/pmod-esp32-wireless-communication-module/

DIGILENT. (18 de 06 de 2022). Obtenido de Basys 3 Artix-7 FPGA Trainer Board: Recommended for Introductory Users: https://digilent.com/shop/basys-3-artix-7-fpga-trainer-board-recommended-for-introductory-users/

DIGILENT. (18 de 06 de 2022). Obtenido de Pmod KYPD: 16-button Keypad : https://digilent.com/shop/pmod-kypd-16-button-keypad/

Xilinx. (18 de 06 de 2022). Obtenido de http://www.xilinx.com

ANEXOS

1. Anexo I. Material entregado en Sharepoint

2. La entrega se divide en los siguientes archivos:

3. Este documento, tanto en formato word como pdf, que es la memoria del proyecto.
4. Un proyecto en vivado con el circuito al completo para la implementación física.
5. Un proyecto en vivado con el circuito modificado para poder simular su funcionamiento (ya que no es fácil simular un teclado).
6. Una ilustración que explica el cifrado afín.
7. Una ilustración del circuito de este proyecto simplificado.

Para simular el proyecto, abrir el proyecto simulación y en la entidad pruebas cambiar valores libremente.
input es la tecla introducida. (la primera es la constante de multiplicación y la segunda la de desplazamiento) Para la correcta simulación deben introducirse 7 teclas en hexadecimal.
puls es el switch que indica que la última tecla hay que registrarla, por lo que después de cada tecla introducida debe activarse y desactivarse (usar prueba actual cambiando solo el valor de las teclas introducidas).
elec es el switch que indica si cifrar '1' o descifrar '0'.
rst es el switch que resetea el sistema por completo. Debe activarse y desactivarse al empezar la simulación, tal y como está actualmente en la entidad pruebas.
actu es el switch que activa el cifrado (o descifrado si elec es 0) una vez que ya están todas las teclas en sus respectivos registros.

output es la salida que nos interesa, que es la que nos muestra las teclas ya modificadas. Hay que fijarse en los valores de output algunos ciclos de reloj más tarde de la activación de **actu,** que será a partir de entonces cuando empiece a mostrar las teclas modificadas en bucle.