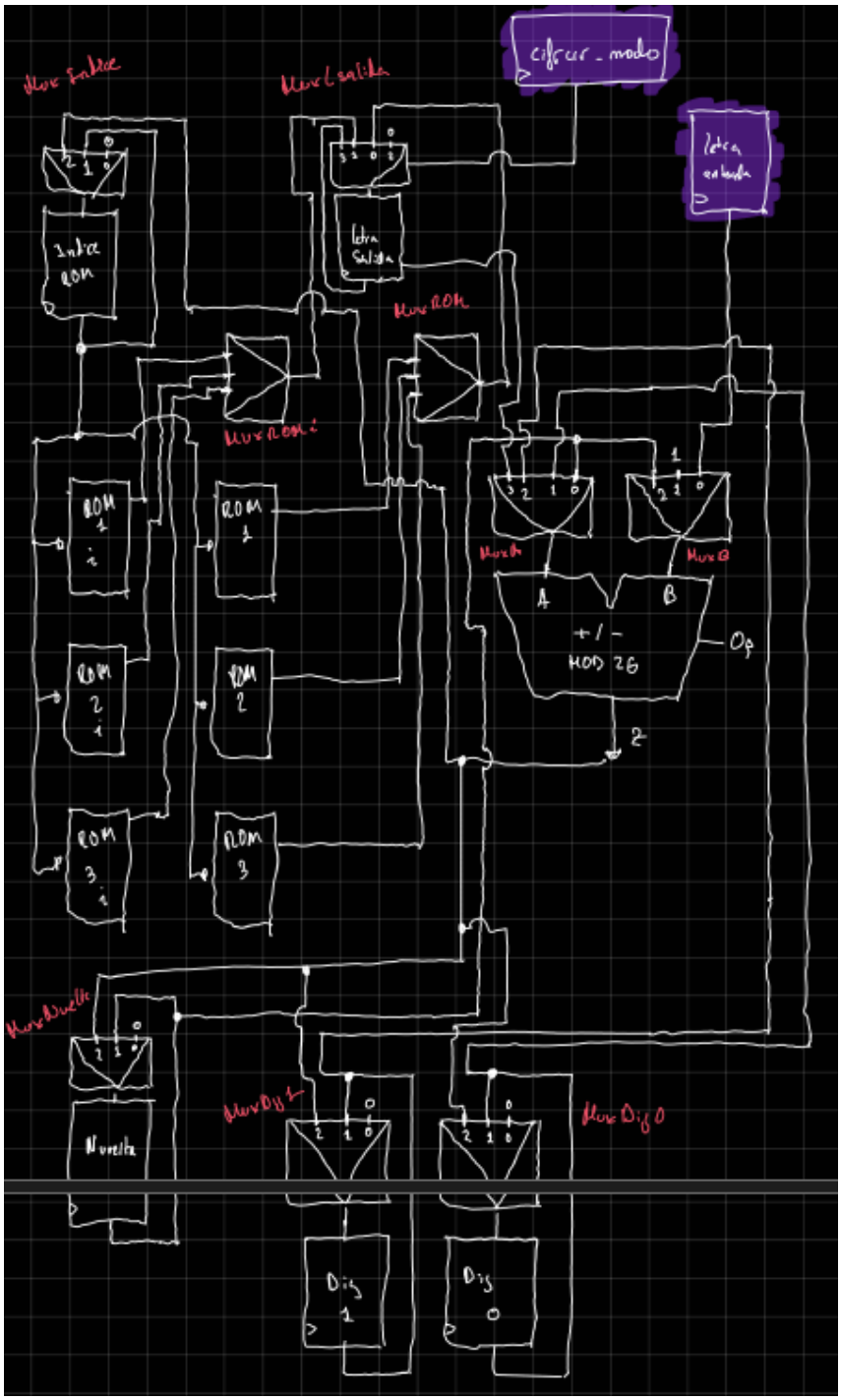
Implementación Hardware de Máquina Enigma

Ingeniería de Computadores - Diseño de Sistemas Digitales

Autores:
Juan Pastrana García
Omar Ouahri Vigil



Plataforma: FPGA Xilinx Artix-7 (Basys 3)
Lenguaje: VHDL

Índice

1. Introducción y Objetivos

El objetivo de este proyecto es recrear el funcionamiento criptográfico de la histórica Máquina Enigma utilizando tecnologías de hardware reconfigurable (FPGA). Inspirado en los principios electromecánicos estudiados por Alan Turing, este diseño traslada los rotores físicos, el reflector y el panel de conexiones a lógica digital síncrona descrita en VHDL.

El sistema permite cifrar y descifrar mensajes carácter a carácter, manteniendo un estado interno (memoria) que evoluciona con cada pulsación, garantizando así la naturaleza polialfabética del cifrado.

2. Arquitectura del Sistema

2.1. Diseño Conceptual (Datapath)

Antes de la implementación, se definió un esquema de ruta de datos (Figura 1) que modela el flujo de la información. La letra de entrada es procesada secuencialmente por etapas de transformación que simulan el paso de la corriente eléctrica a través de los rotores mecánicos.

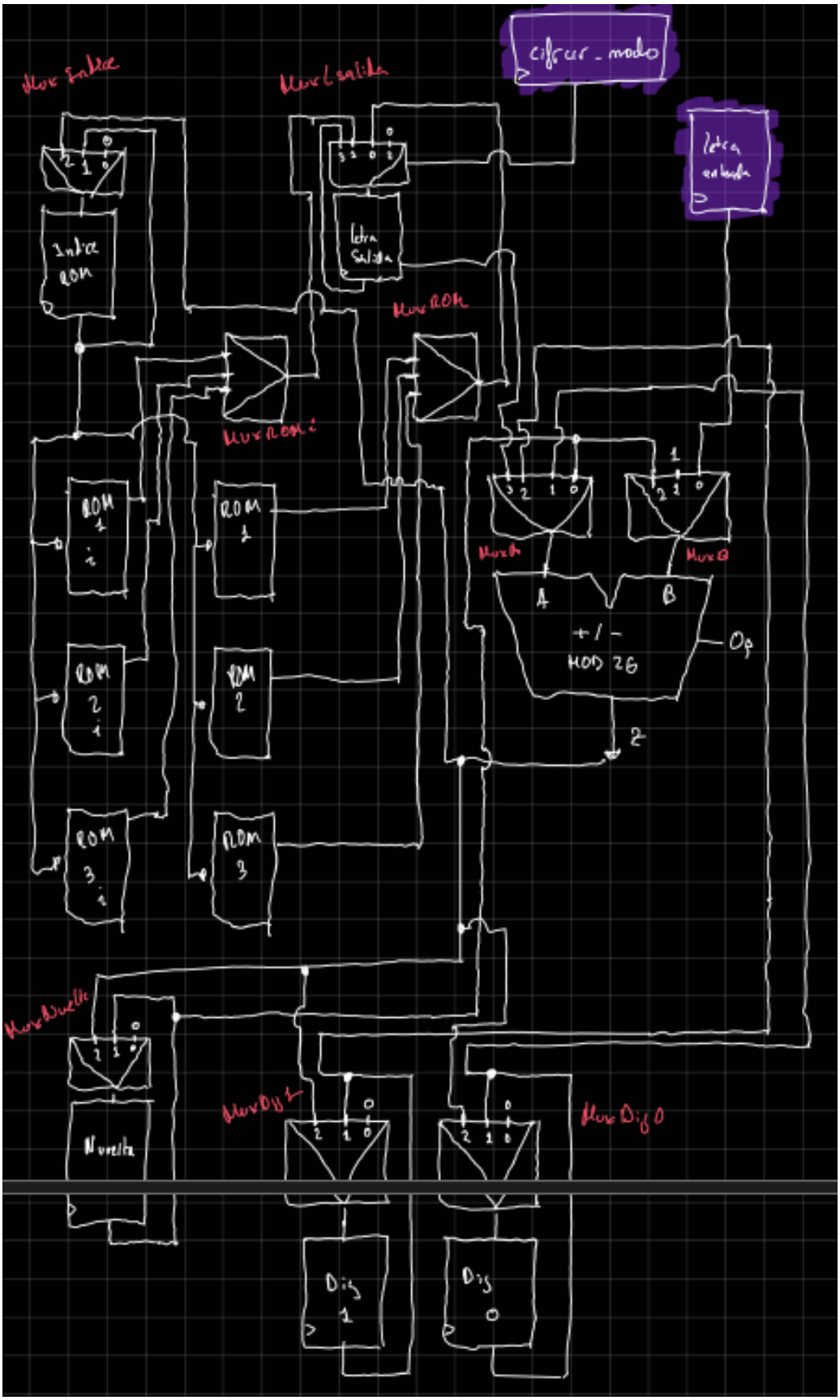


Figura 1: Diagrama esquemático original del Datapath. Se aprecia la estructura de multiplexores y realimentación necesaria para el cifrado.

2.2. Lógica de Control (FSM)

La coordinación entre el usuario y el hardware es gestionada por una Unidad de Control basada en una Máquina de Moore. Esta unidad (Figura 2) es responsable de:

- Detectar la pulsación del botón de cifrado.
- Activar las señales de control de los multiplexores.
- Calcular el siguiente estado de los rotores (lógica de trinquete).

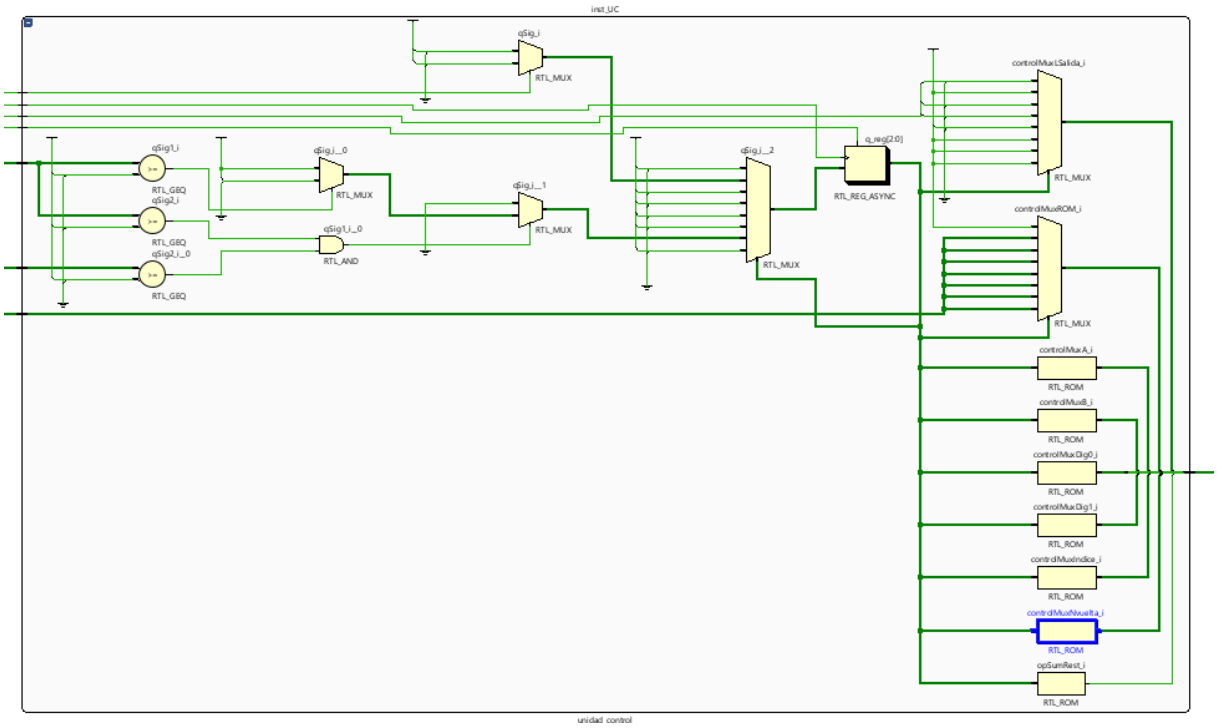


Figura 2: Esquemático sintetizado de la Unidad de Control (Vivado RTL). Se observan las señales de estado y la lógica de decodificación de salida.

3. Detalles de Implementación RTL

El diseño se ha sintetizado exitosamente utilizando Xilinx Vivado. A continuación se detallan los módulos críticos.

3.1. Top Level y Jerarquía

La Figura 3 muestra la interconexión global. Se destaca la separación entre el `debouncer` (izquierda) para filtrar la entrada mecánica, y el núcleo de procesamiento `datapath`.

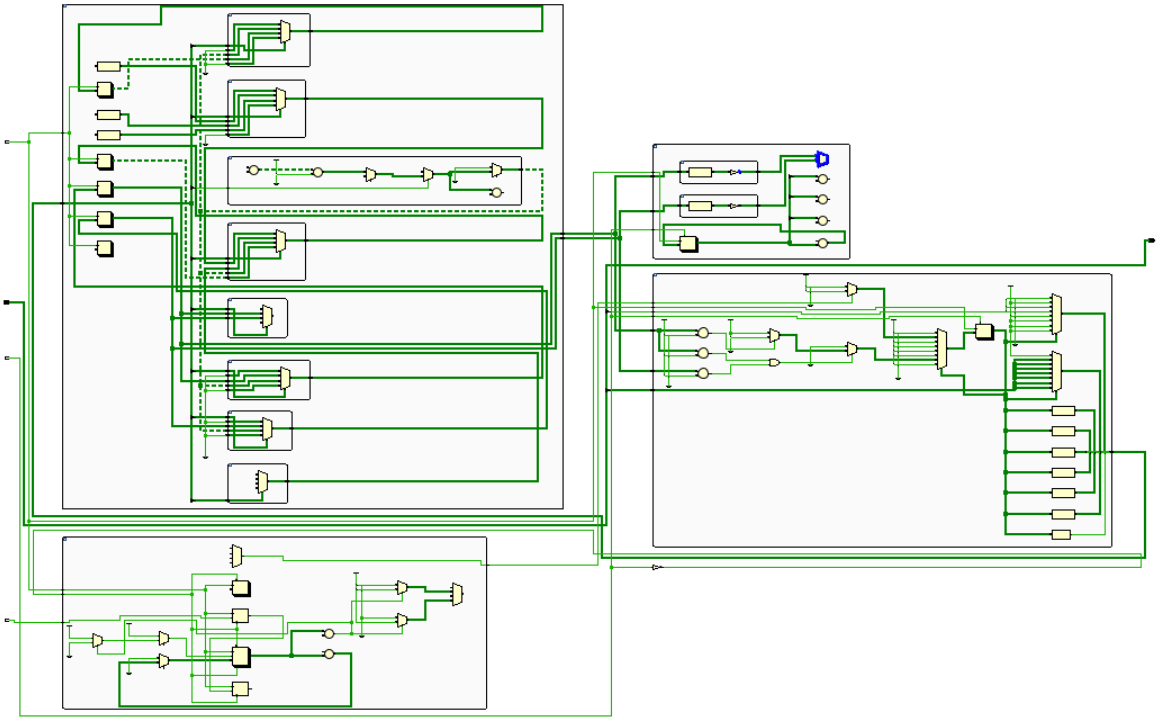


Figura 3: Vista RTL Top-Level. Los buses verdes representan las conexiones de datos de 5 bits (letras).

3.2. Núcleo Aritmético (ALU)

La sustitución de caracteres se realiza mediante aritmética modular. El módulo `instSumRest` (Figura 4) implementa la operación $Y = (A \pm B) \pmod{26}$, esencial para calcular los desplazamientos relativos de los anillos.

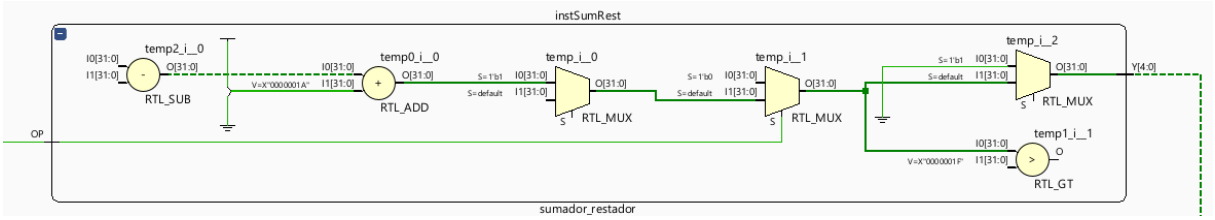


Figura 4: Detalle del Sumador/Restador Modular. Incluye comparadores para manejar el desbordamiento del alfabeto (26 letras).

3.3. Integridad de Señal (Debouncer)

Dado que el cifrado depende de pulsaciones físicas, se implementó un circuito antirrebotes (Figura 5) que introduce un retardo de 50ms para estabilizar la señal de entrada, evitando saltos múltiples de rotor indeseados.

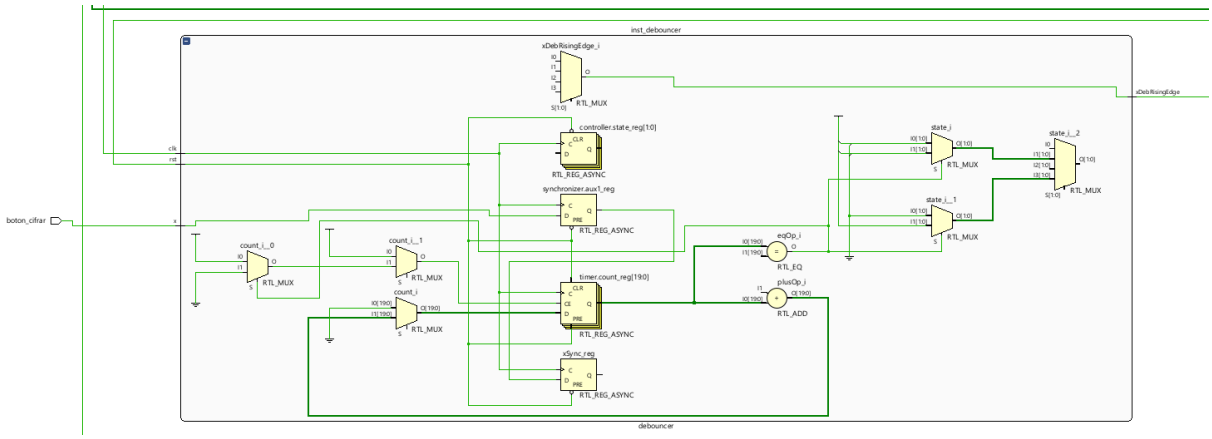


Figura 5: Circuito Debouncer con temporizador y registro de desplazamiento.

4. Manual de Usuario

Para operar la máquina en la placa Basys 3, siga las instrucciones de la Tabla 1.

Control	**Puerto Físico**	**Acción**
Reset	BTN Central	**Pulsar al inicio**. Reinicia rotores a 00.
Cifrar	BTN Derecho	Ejecuta el cifrado y mueve rotores.
Selección Letra	SW [4:0]	Entrada binaria (A=00000 ... Z=11001).
Selección Rotor	SW [14:13]	Cambia la configuración interna de cableado.
Modo	SW [15]	`0`=Cifrar (Down), `1`=Descifrar (Up).

Cuadro 1: Mapa de Interfaz de Usuario

5. Conclusiones

Se ha logrado implementar una máquina de cifrado funcional que respeta los principios teóricos de la Enigma. La combinación de un diseño conceptual claro a mano y su posterior traducción a VHDL sintetizable ha permitido crear un sistema robusto, verificable mediante los esquemáticos RTL generados por la herramienta de síntesis.