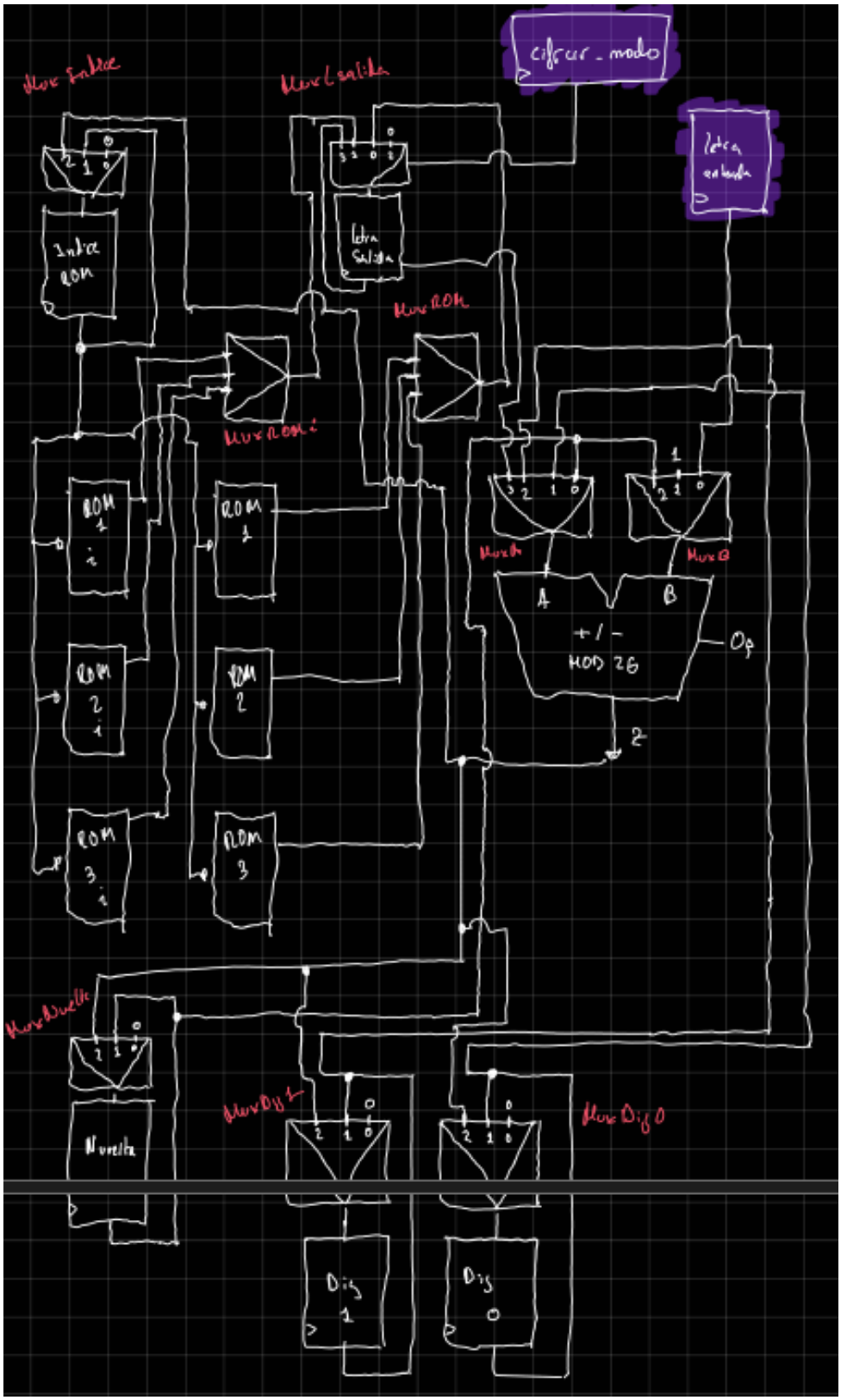
Sistema Criptográfico Hardware

Implementación de Máquina Enigma en FPGA Artix-7

Autores del Diseño:
Juan Pastrana García
Omar Ouahri Vigil



Resumen Técnico:
Diseño VHDL modular con separación estricta de Datapath y Unidad de Control.
Implementación física en Digilent Basys 3.

Índice

1. 1. Principios de Diseño

El objetivo del proyecto no es la simulación software, sino la **emulación hardware** de los procesos electromecánicos de la máquina Enigma. Se ha priorizado la estabilidad de la señal y la modularidad del código VHDL.

1.1. Arquitectura General

El sistema sigue una arquitectura de procesador dedicado, dividida en dos dominios principales visibles en el esquema RTL Top-Level (Figura 1):

1. **Datapath (Ruta de Datos):** Contiene los elementos de transformación (Multiplexores, ALU, ROMs). Es puramente combinacional excepto por los registros de estado de los rotores.
2. **Unidad de Control (FSM):** Gobierna la secuencia temporal. Decide cuándo se carga un dato y cuándo avanza un rotor.

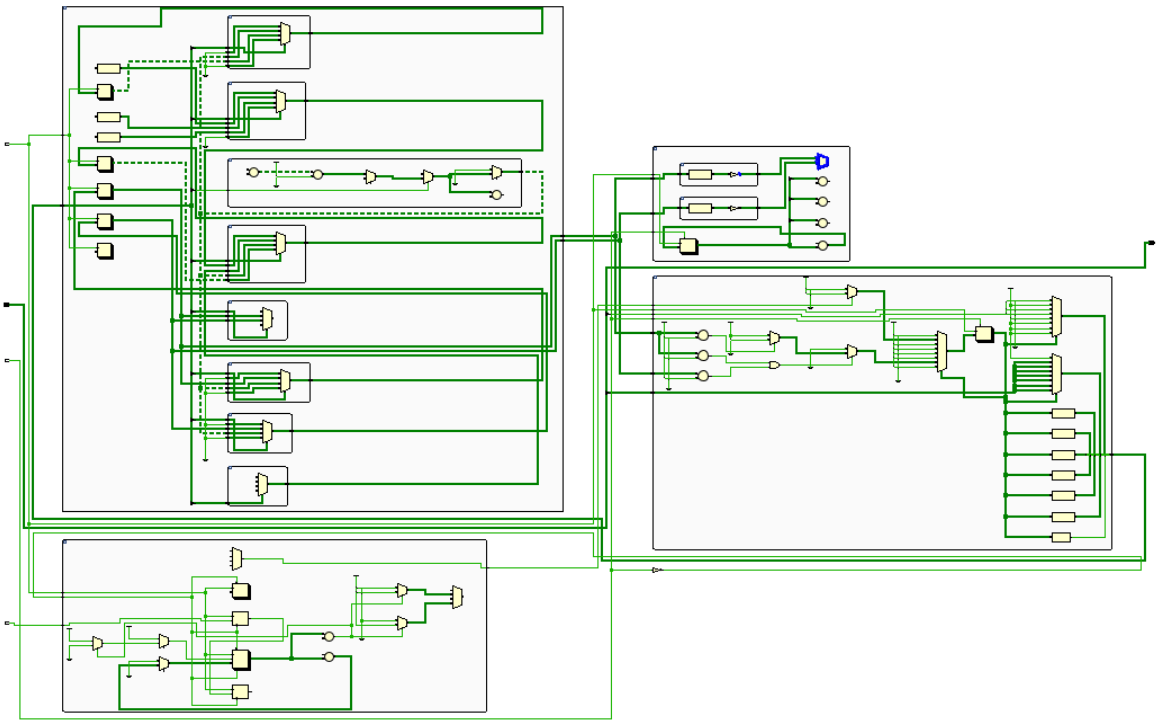


Figura 1: Arquitectura RTL sintetizada en Vivado. Se aprecia la interconexión entre la lógica de control, el datapath y los periféricos de I/O.

2. 2. Análisis del Datapath

2.1. Concepción Original vs Implementación

El diseño partió de un esquema conceptual dibujado a mano (Figura 2), donde se modeló el flujo de la letra a través de los rotores como una serie de sumas de offsets

y sustituciones.

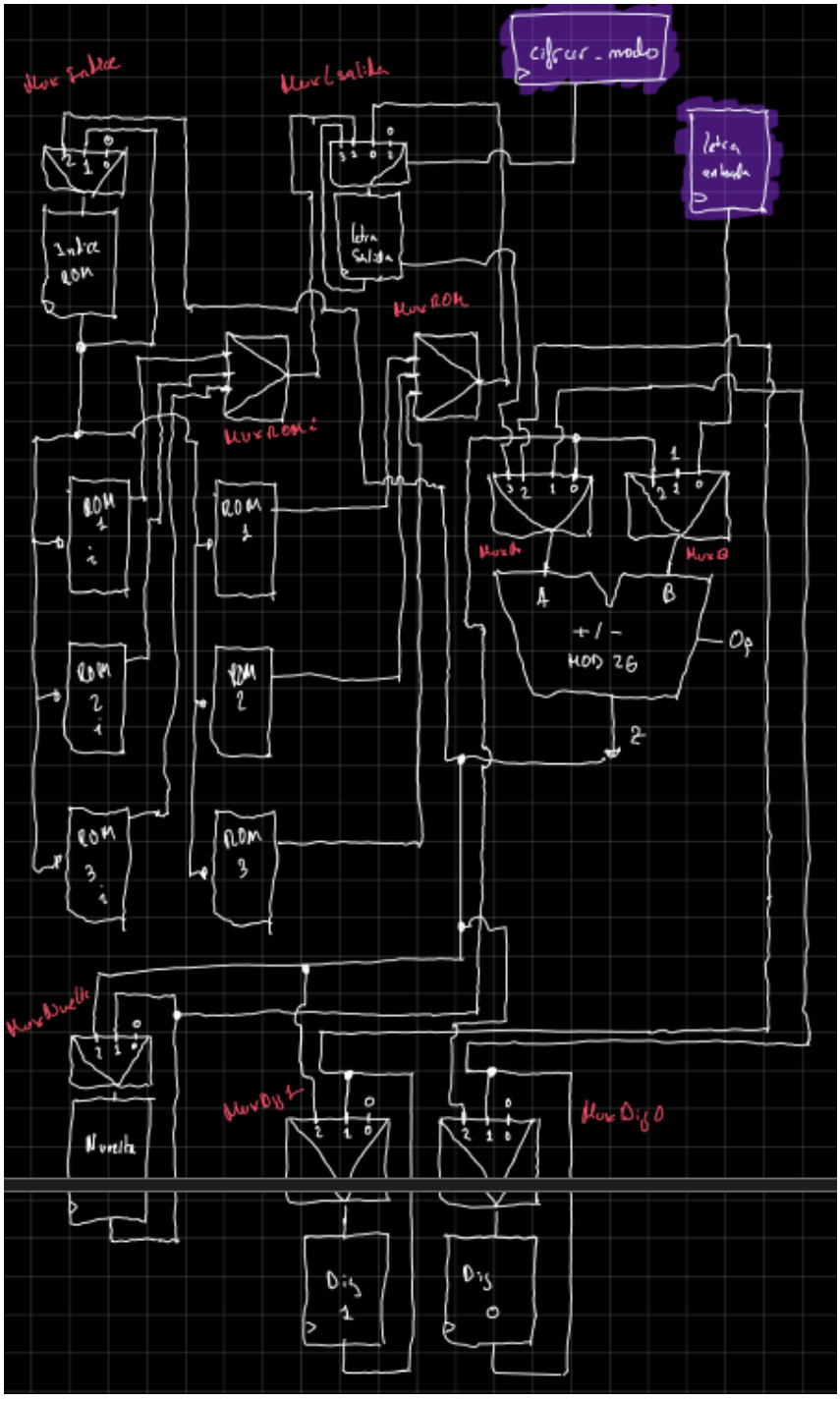


Figura 2: Diseño conceptual original del flujo de datos (Juan Pastrana & Omar Ouahri).

2.2. Aritmética Modular (El Núcleo Matemático)

A diferencia de una CPU de propósito general, esta máquina requiere una ALU especializada en aritmética de campo finito (Alfabeto de 26 caracteres). El módulo `sumador_restador` implementa la ecuación fundamental del cifrado Enigma:

$$C_i = (P_i + K_i) \pmod{26} \quad (1)$$

Donde P_i es la letra plana y K_i el desplazamiento del rotor. La implementación RTL (Figura 3) muestra el uso de comparadores para corregir el desbordamiento del módulo.

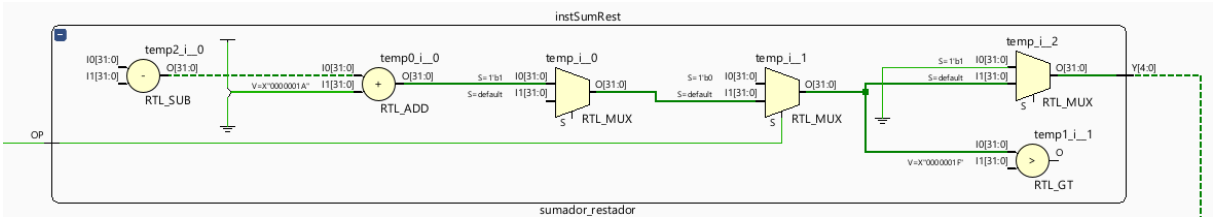


Figura 3: Detalle de la ALU Modular. Nótese la lógica de comparación para mantener los valores en el rango [0-25].

3. 3. Lógica de Control y Sincronización

3.1. Máquina de Estados (FSM)

El çorazón"del sistema es una Máquina de Moore compleja (Figura 4). Sus estados críticos son:

- **S1 (Idle):** Espera de evento de usuario.
- **S4-S6 (Mecánica):** Simulación del trinquete. Evalúa si el *Rotor 0* ha completado una vuelta para arrastrar al *Rotor 1*.

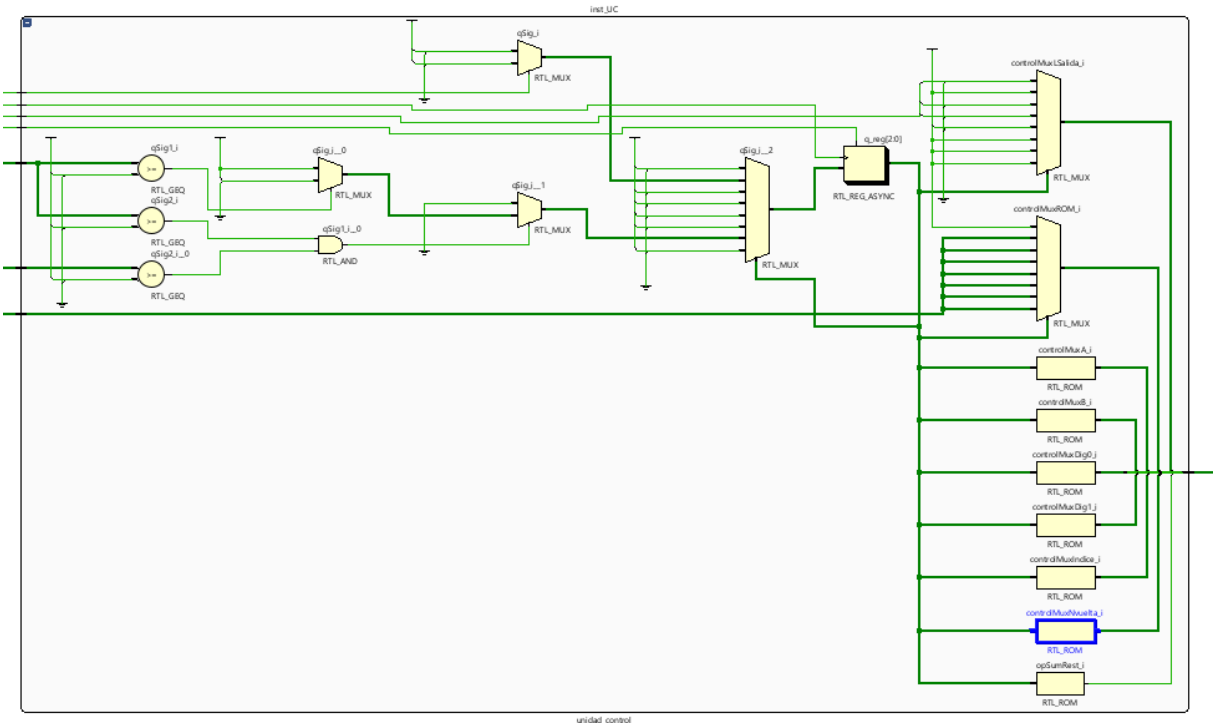


Figura 4: Lógica secuencial de la Unidad de Control. Gestiona los saltos de estado y las señales de habilitación de registros.

3.2. Integridad de Entrada (Debouncing)

Dado que la FPGA opera a 100MHz y los pulsadores mecánicos tienen rebotes de 10-20ms, se implementó un filtro digital (Figura 5). Este bloque asegura que una pulsación física se traduzca en exactamente un ciclo de cifrado.

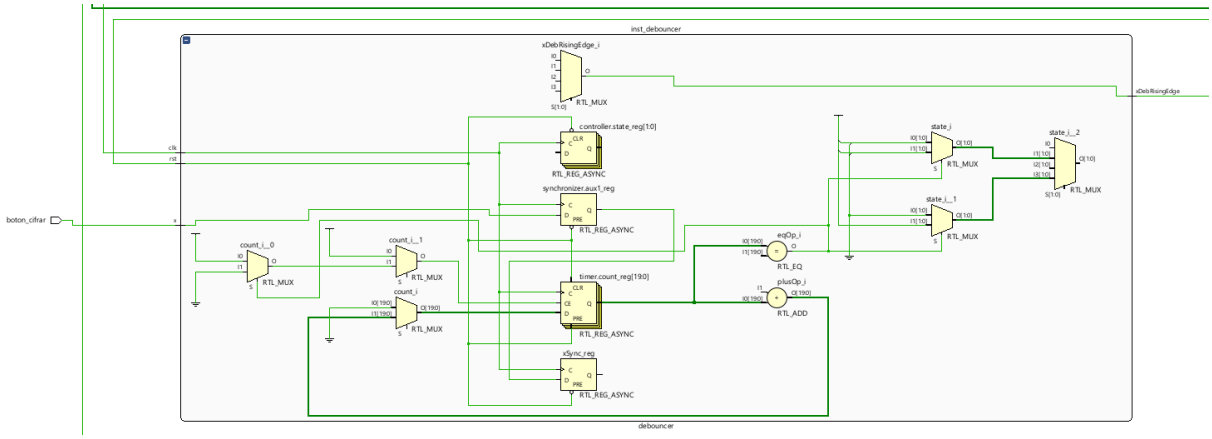


Figura 5: Circuito Debouncer. Utiliza un contador de temporización para filtrar ruido en la señal `btn_cifrar`.

4. 4. Conclusiones y Futuras Mejoras

La implementación en FPGA demuestra ser superior en determinismo y velocidad respecto a soluciones software. La arquitectura modular permite la fácil adición de más rotores o un panel de conexiones (*Plugboard*) en futuras iteraciones sin rediseñar el núcleo de control.