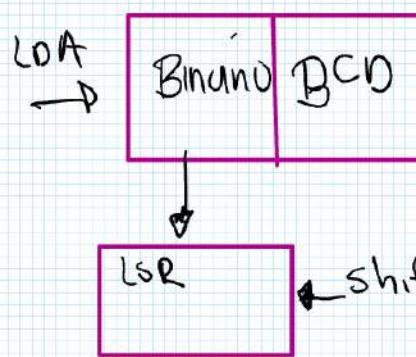
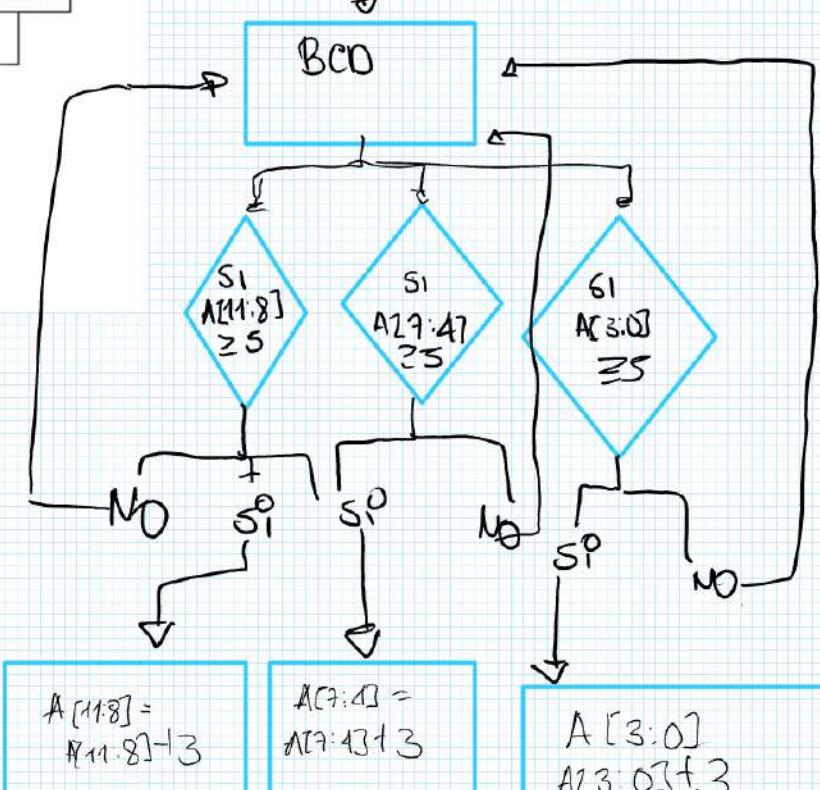
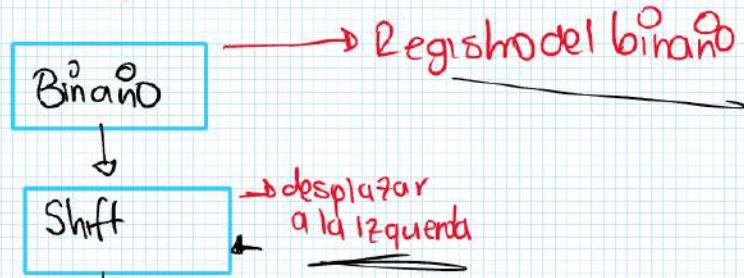
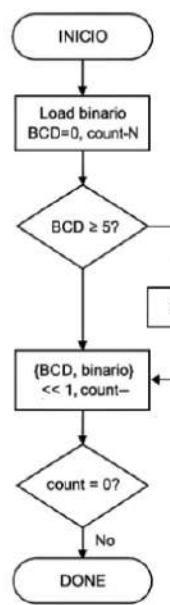


$$\begin{array}{l} \xrightarrow{2^0 = 1} \\ \xrightarrow{2^1 = \frac{4}{5}} \end{array}$$

$A[11:0]$

10

Lapash



Ejemplo 2

Explicación del punto

contenemos
Registro de sumando grande
Las salidas NO se pueden unir \rightarrow corto circuito

Figura 1.16 Identificación de componentes del camino de datos para la obtención de resultados finales.

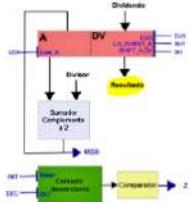


Figura 1.17 Implementación informática de datos para la obtención de resultados finales.

el multiplicador no afecta
nada

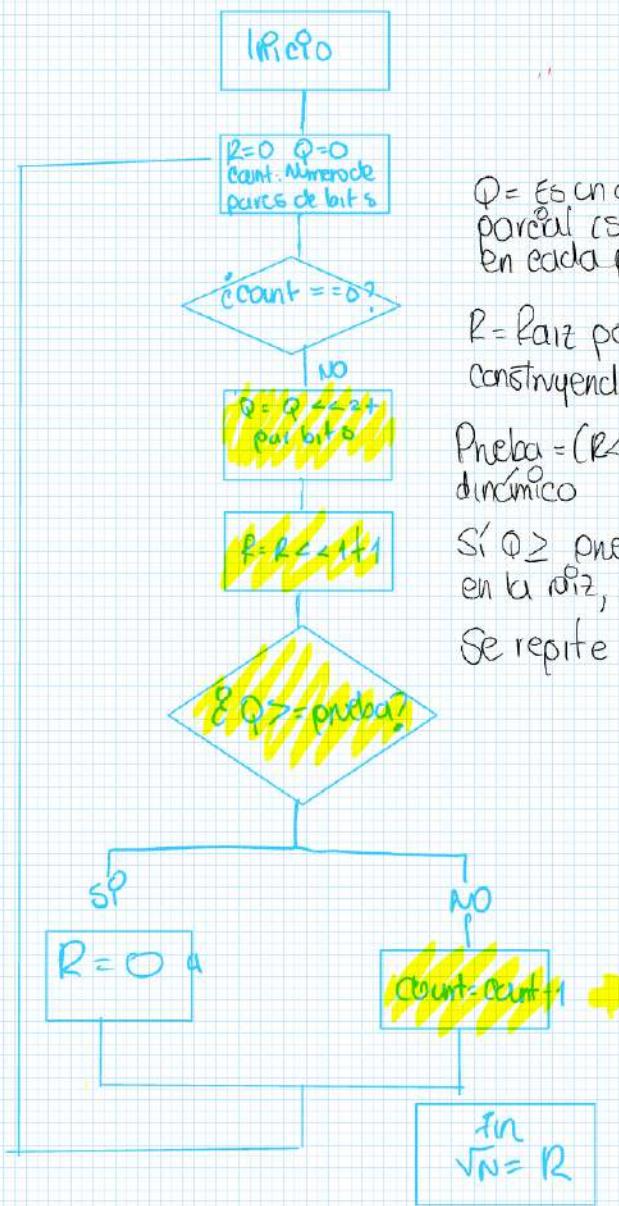
Complemento 2

$- \leftarrow 0001$

$+ \leftarrow 0001$

las entradas del dato float no son las mismas de la máquina de estado

Data path del algoritmo de la raíz.



$Q =$ Es un acumulador parcial (se va restando en cada paso)

R = Raíz parcial se va construyendo bit a bit

Prueba = $(R < 1) + 1$ es el divisor dinámico

Sí $Q \geq$ prueba, se resta y se pone en la raíz, si no se deja 0

Se repite hasta que count = 0

Un Data path muestra los registros, multiplexores, ALU y operaciones necesarias para implementar el Algoritmo en Hardware

Elementos principales del data path

① Registro del Radicando

Shift \rightarrow registrador de 2 bits

② Registro Q (Acumulador parcial)

Guarda el residuo
se actualiza en cada ciclo con $Q \ll 2$ y sumando los nuevos 2 bits

③ Registro R parcial

se desplaza a $R \ll 1$ en cada iteración
se actualiza en el resultado de la resta $Q -$ prueba

④ Generador de prueba

⑤ Sumador y restador

⑥ Comparador

⑦ Acumulador

- ② Bus de datos → Transporta los valores que se leen o escriben
 ③ Bus de control → Indica si se va a leer o escribir (sendles rd, wr)

SQR	0X440000
UART	0X450000
GPIO	0 X46.0000
RAM	0X3FFF 0x0000

Es una dirección de memoria, cada periférico tiene su propia memoria. Cuando el procesador Risc-V se comunica con sus periféricos, usa cables separados para cada uno. En su lugar, usa un solo bus de direcciones que accede exclusivamente a una región de memoria.

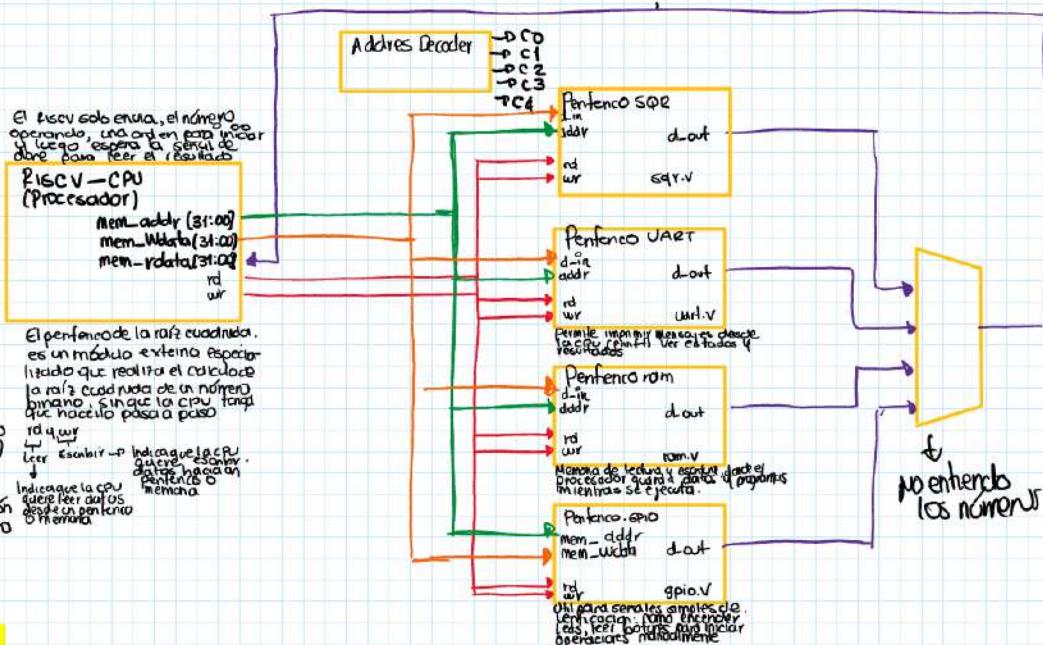
always @* begin
 case(mem_addr[31:16]) // direcciones chips-select

```
16'h0044 : cs = 6'b100000; // SQR
16'h0045 : cs = 6'b010000; // UART
16'h0046 : cs = 6'b001000; // GPIO
default : cs = 6'b000100; // RAM
endcase
end
```

CS = señales que activan o seleccionan un periférico específico

6'b0 = indica cuantos bits tiene el valor 000000

Addr = Indica la dirección de memoria o periférico que el procesador



Tarea: Hay que simularlo

¿Cómo se coloca ese archivo dentro del procesador?

https://github.com/cicamargoba/digital_UN/tree/main/femtoRV/simulation



SoC_femto_V
 Procesor: Es el núcleo RISC-V que ejecuta instrucciones, genera direcciones de memoria (mem_addr), datos (mem_udata), señales de lectura y escritura (mem_rd, mem_wdata).

Corazón del sistema
 define System on Chip que integra la CPU RISC-V, la memoria, UART y los periféricos.

Memory: Simula la RAM del sistema, donde estás el programa.

I/O (Entrada, salida): Define los periféricos mapeados en memoria, como los lejos y el UART.

Para agregar el periférico de la cinta:

① Definir el bit del periférico: Debajo de los locales: localparam I_O_SQRT_bit = 3; // nuevo periférico de rotación cuadrada (30,129)

② Crear el módulo sqrt.v



Multiplicación

lunes, noviembre 24, 2025

10:08 AM

Diagrama de flujo

