

$$2^0 = 1$$

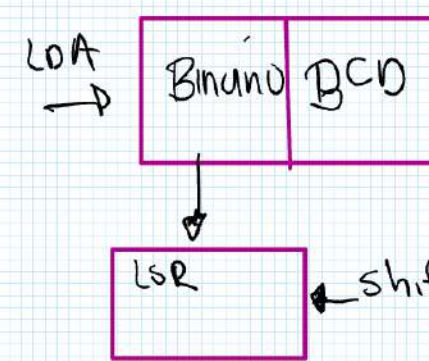
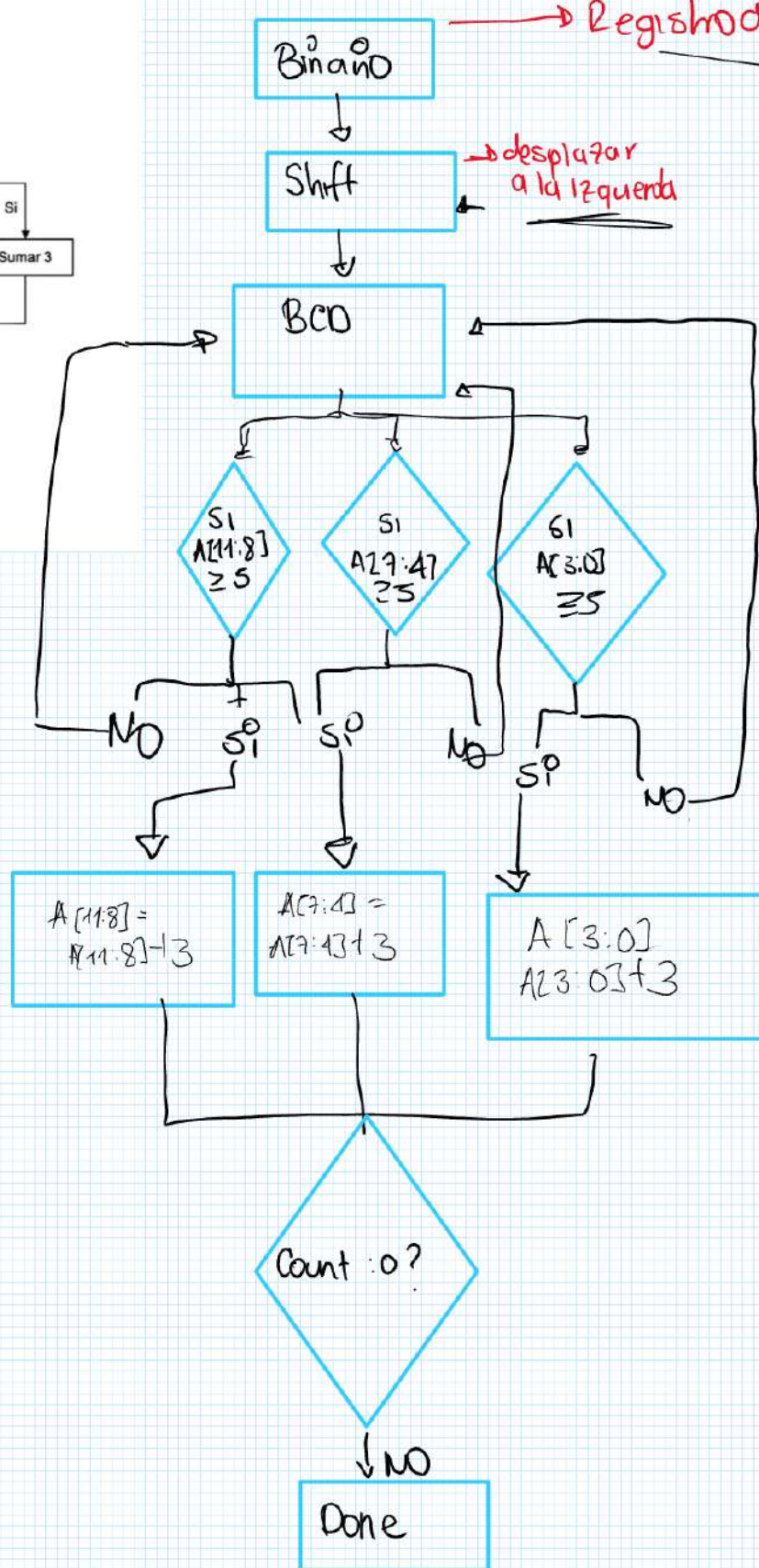
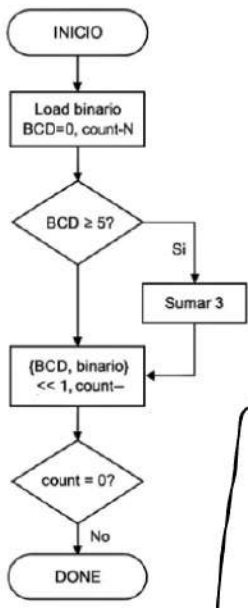
$$2^2 = \frac{4}{5}$$

$A[11:0]$

1 aapah

Registro del binario

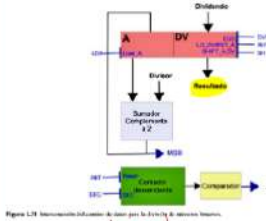
desplazar a la izquierda



Sumado 2

Explicación del profe

contatenamos \Rightarrow
Registro de comprobante grande
Las salidas no se pueden unir \rightarrow corto circuito



El autor no afecta a nadie.

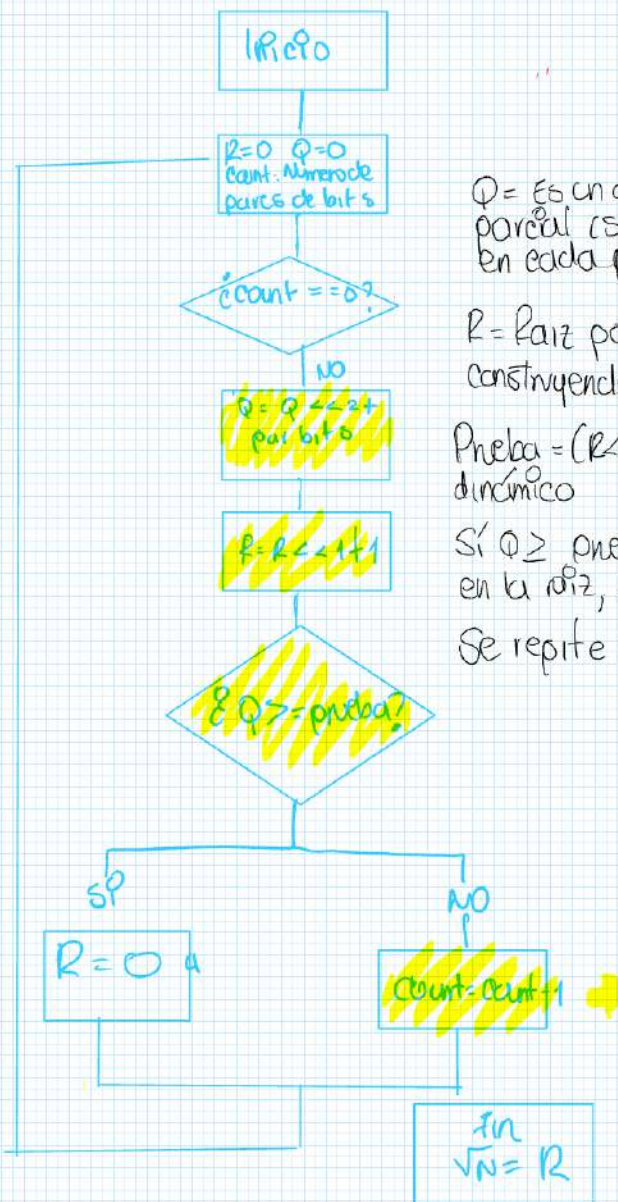
Complemento 2

• $\leftarrow \textcircled{1001}$

1-0001

las entidades del departamento son las mismas de la máquina de estado

Data positiu del algoritmo de la raíz.



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

R = Raíz parcial se va construyendo bit a bit

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

Si $\emptyset \geq$ prueba, se resta y se prueba en la raíz, si no se deja \emptyset .

Se repite hasta que $\text{count} = 0$

Un Datapath muestra los registros, multiplexores, ALU y conexiones necesarias para implementar el Algoritmo en Hardware

Elementos principales del data path

- 1 Registro del Radicando
Shift → Desplazador de 2 bits
- 2 Registro Q (A acumulador por
Guarda el residuo
se actualiza en cada ciclo
con $Q \ll 2$ y sumando los
nuevos 2 bits
- 3 Registro R parcial
se desplaza $\ll 1$
en cada iteración
se actualiza con el resultado
de la resta Q -pnebo
- 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 (señales rd, wr)

SQR	0X440000
UART	0X450000
GPIO	0X460000
RAM	0X3FFF

0X0000

Es una dirección de memoria, cada periférico tiene un rango de memoria. Cuando el procesador RISC-V se comunica con sus periféricos, no usa cables separados para cada uno. En su lugar, usan solo bus de direcciones y a cada periférico se le da una región de memoria exclusiva, llamada *mapa 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

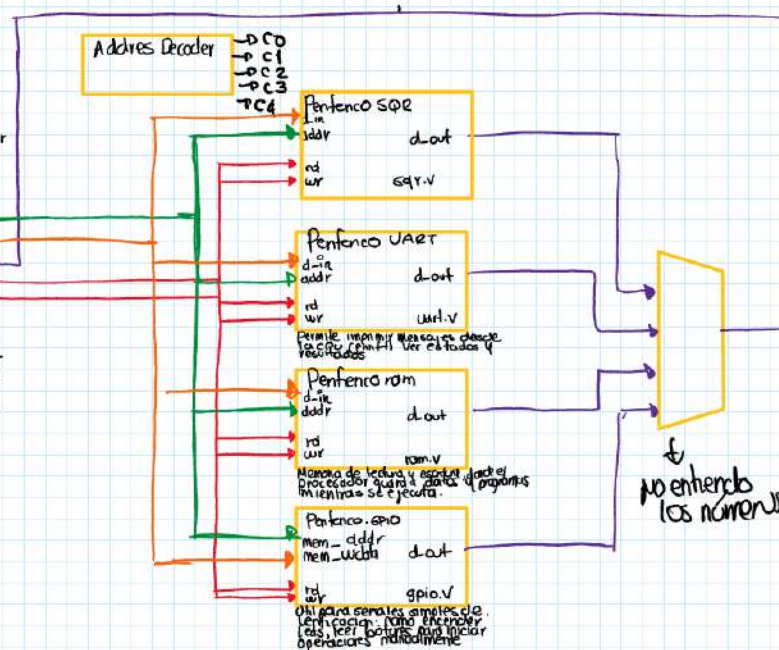
- CS = Señales que activan o seleccionan un periférico específico
- 6'b0 = Indica cuántos bits tiene el valor 000000
- Addr = Indica a qué dirección de memoria o periférico quiere el procesador

- Bus de dirección (mem_addr)**
(Indica que registro se está accediendo (OP_A, C12, RESULT))
Bus de datos de salida (mem_vdata)
(Lleva el número que la CPU escribe)
- Bus de datos de entrada (mem_rdata)**
(Devuelve el resultado o el estado (data))
- Bus de control (wr, rd)**
(Define si la operación es lectura o escritura)

El RISC-V solo envía, el número de operación, una vez en cada ciclo y luego, espere la señal de done para leer el resultado

RISC-V - CPU (Procesador)
mem_addr [31:00]
mem_wdata [31:00]
mem_rdata [31:00]
rd
wr

El periférico de la raíz cuadrada es un módulo externo especializado que realiza el cálculo de la raíz cuadrada de un número binario. Sin que la CPU tenga que hacerlo paso a paso.
rd y wr
Leer
Escribir → Indica que la CPU quiere leer datos o escribir datos en memoria



no entiendo los números

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
→ Organización del sistema
define System on Chip que integra la CPU RISC-V, la memoria, UART y los periféricos
- Processor: Es el núcleo RISC-V que ejecuta instrucciones y genera direcciones de memoria (mem_addr), datos (mem_wdata), señales de lectura y escritura (mem_rdata, mem_wdata)
- Memory: Simula la RAM del sistema, donde está el programa
- Io (Entrada, salida): Define los periféricos mapeados en memoria, como los LEO y el UART

Para agregar el periférico de raíz cuadrada

1. Definir el bit del periférico: Debajo de los localparam: localparam IO_SQR_bit: 3; // nuevo periférico de raíz cuadrada (30, 429)
2. crear el módulo sqrt.v

Multiplicación

lunes, noviembre 24, 2025

10:08 AM

Diagrama de flujo

