

Herramientas de Descripción

Descripción Funcional (Algebraica, Secuencial...)

Diagrama de Bloques

Máquina de Estados Finitos (Autómata)

Grafo de Flujo o de Precedencia

Programas:

Algoritmos --> C, ADA, Fortran,...

Componentes --> VHDL, Verilog,...

Combinándolos se facilita la Síntesis

Con Ejemplos se ilustrará el uso de estas herramientas

Ejemplo de Descripción

- *Cálculo aproximado de la raíz cuadrada de un número natural, x*

- **Método:**

r es la raíz (aprox) de x

generamos $2^2, 3^2, 4^2, \dots$ hasta que $(r + 1)^2 > x$

usamos d y s para calcular cómodamente el resultado

- **Algoritmo:**

input x ;

$r := 1$; $d := 2$; $s := 4$; $fin := 0$;

while $fin = 0$ **loop** $r := r + 1$; $d := d + 2$; $s := s + d + 1$;

if $x < s$ *then* $fin := 1$; *end if*; **end loop**;

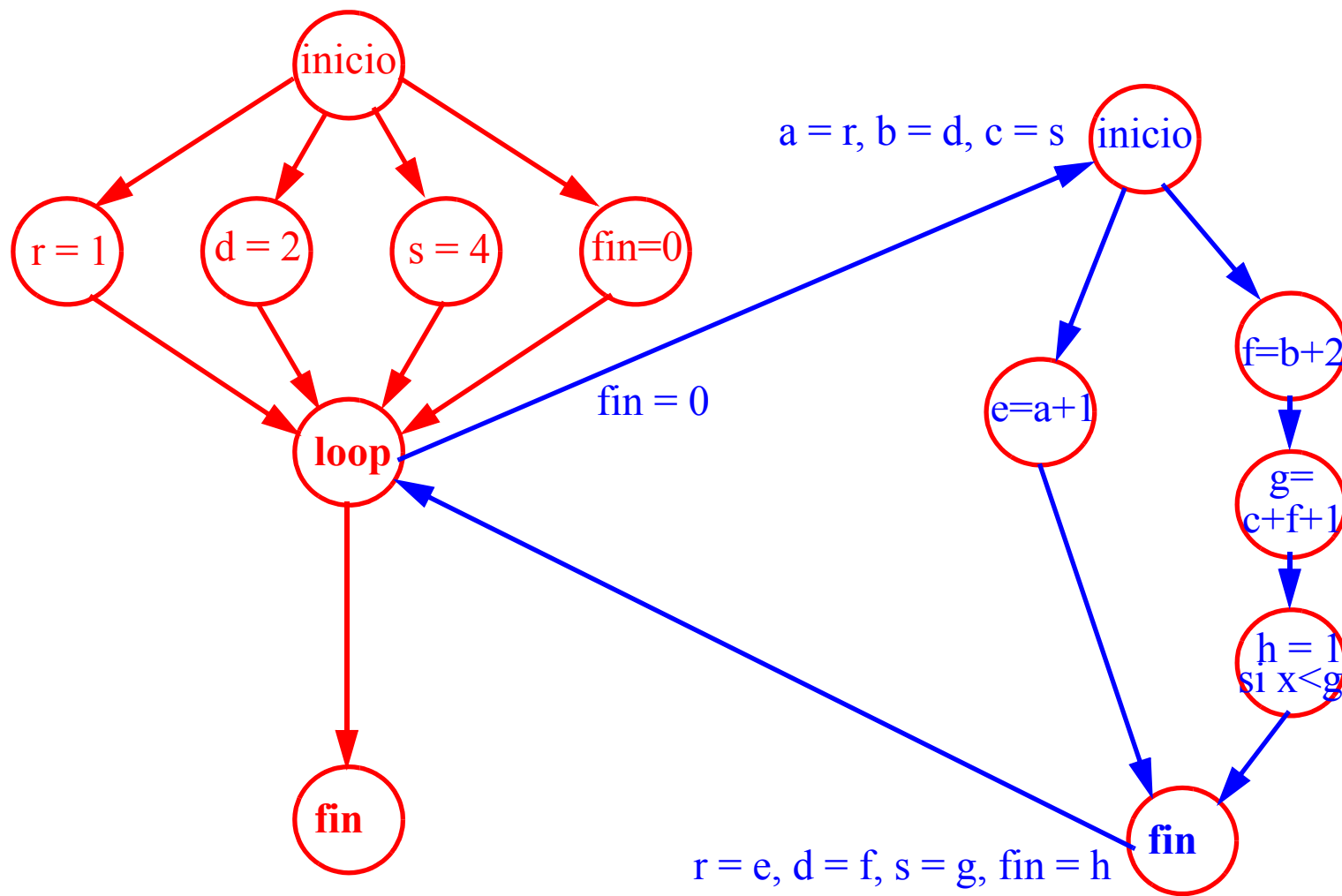
- **Recursos:**

Sumador completo de tres entradas

Registros

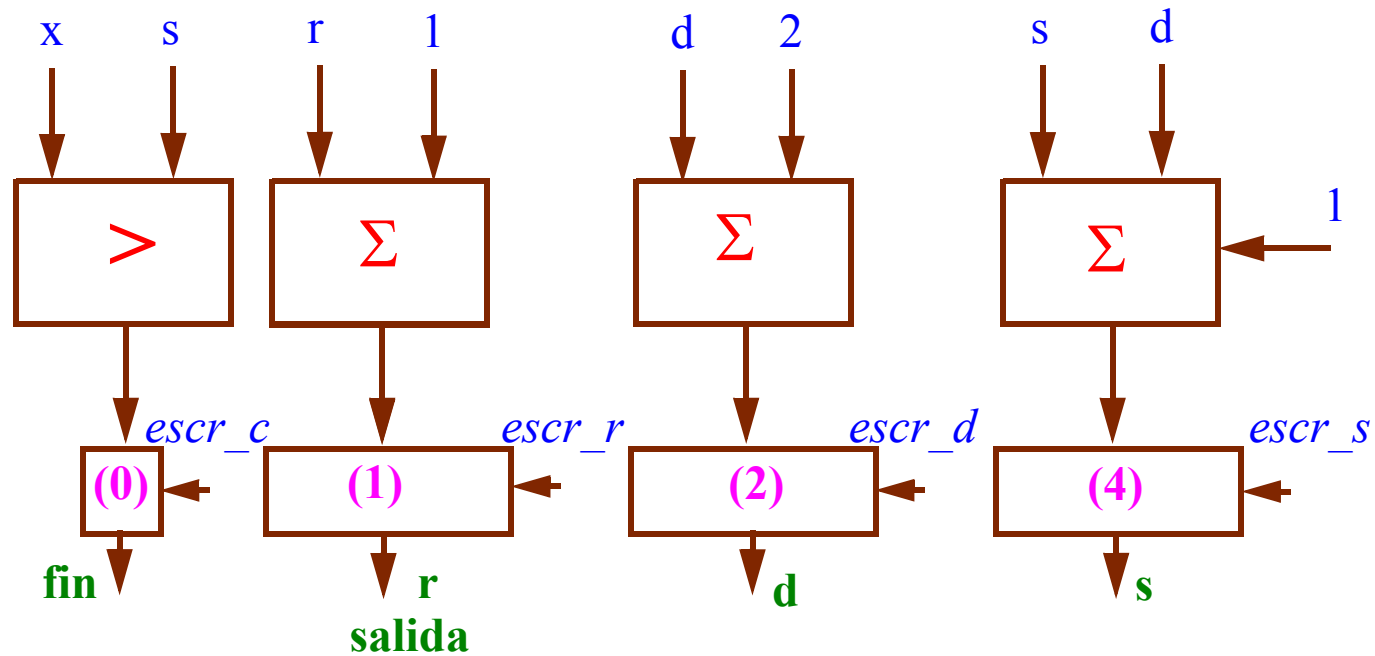
Multiplexores

Ejemplo de Descripción

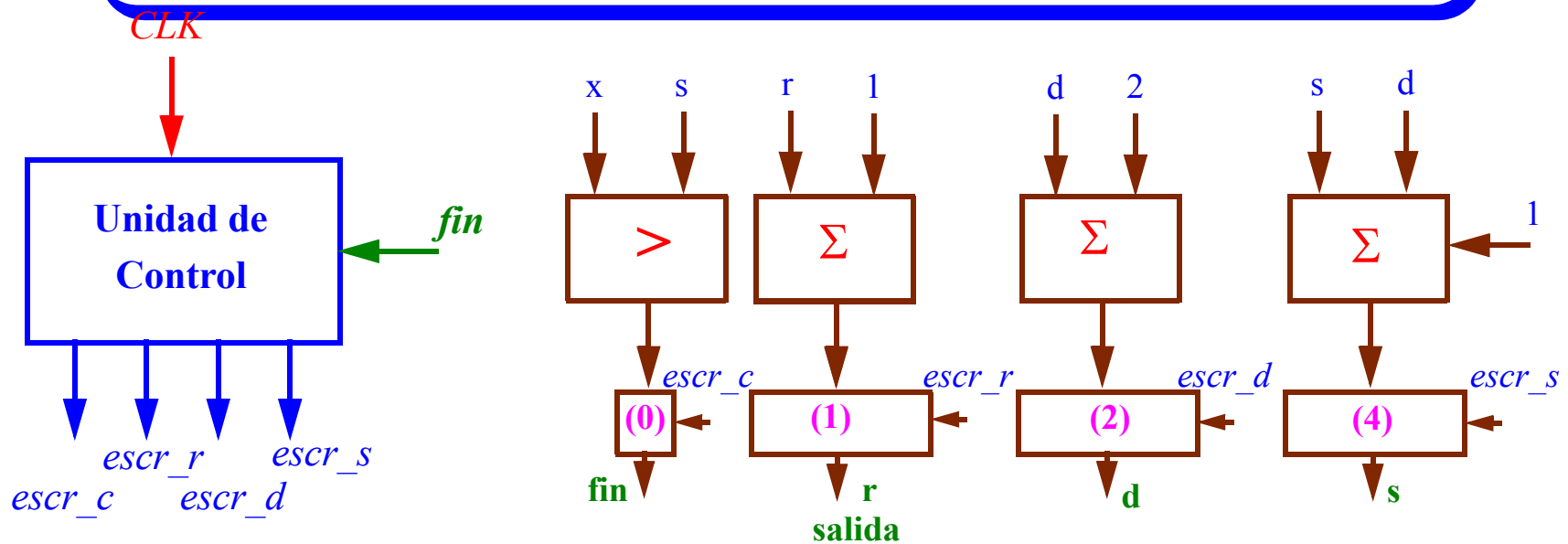


Ejemplo de Descripción

• Estructura #1:



Ejemplo de Descripción



● Proceso:

while fin = '0' loop

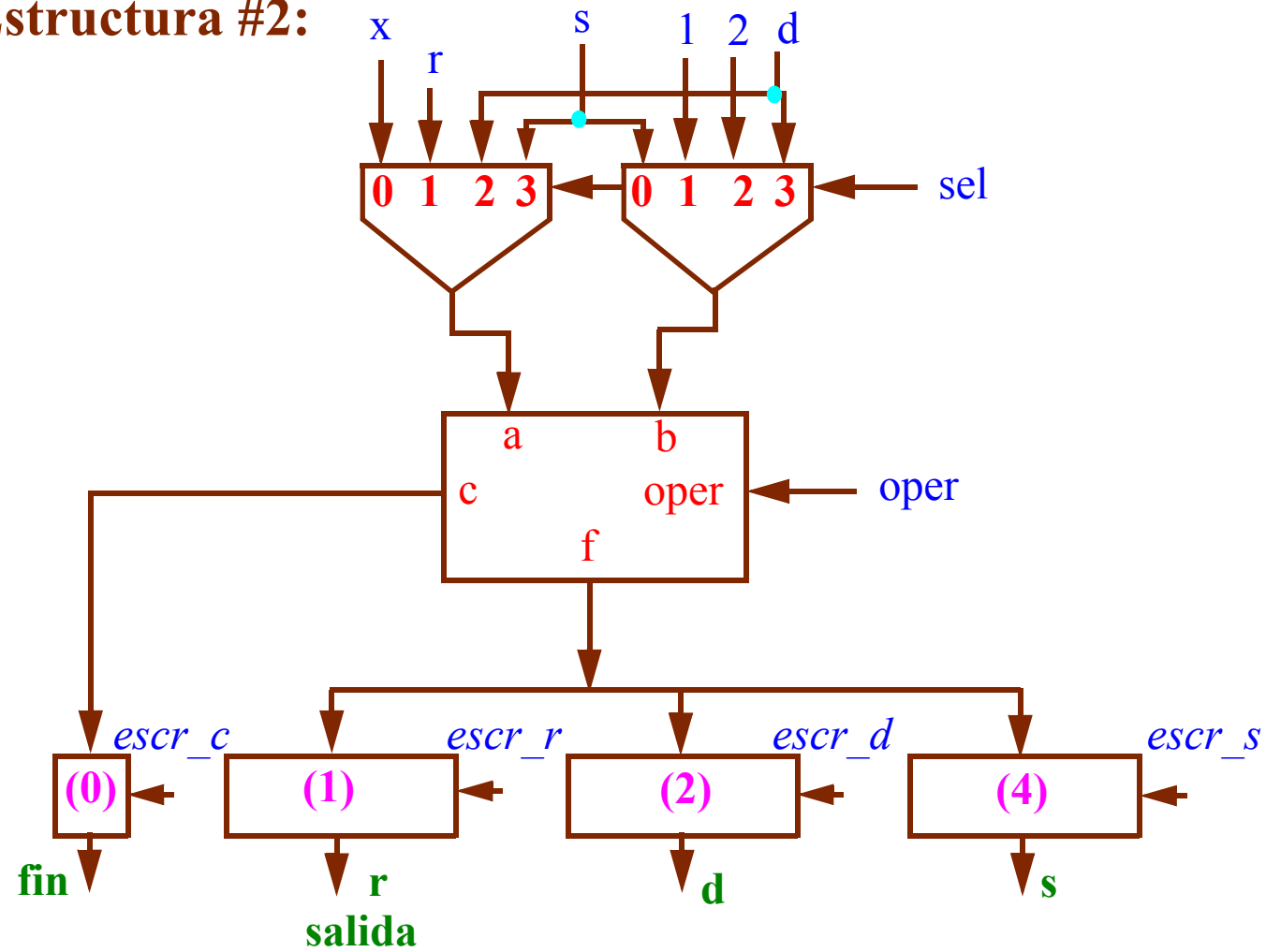
esc_c <= '0'; esc_r <= '1'; esc_d <= '0'; esc_s <= '0'; sync;

end loop;

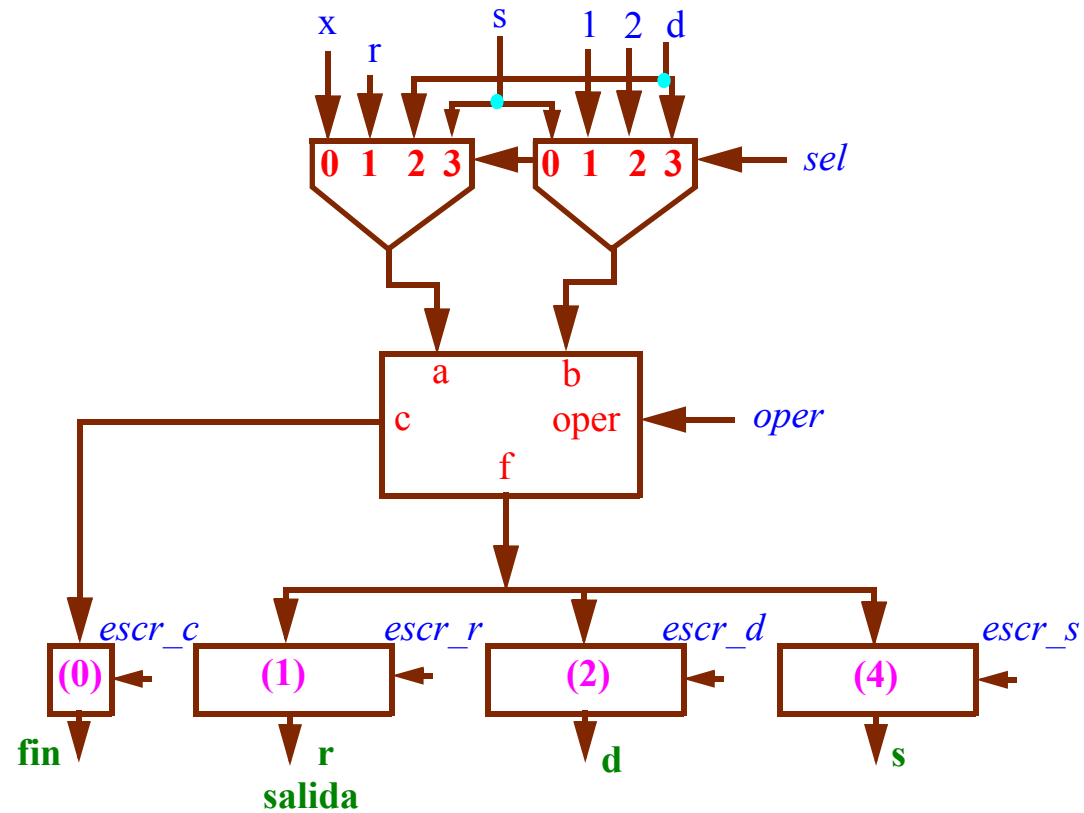
Estado	$fin = 0$	$fin = 1$
s₀	s₁/1111	s₁/0000
s₁	s₀/0000	s₁/0000

Ejemplo de Descripción

• Estructura #2:



Ejemplo de Descripción



<i>sel</i>	<i>a</i>	<i>b</i>
0	x	s
1	r	l
2	d	2
3	s	d

<i>oper</i>	<i>f</i>	<i>c</i>
0	$a + b$	---
1	$a + b + 1$	---
2	---	1 si $a < b$

Ejemplo de Descripción

<i>sel</i>	<i>oper</i>	<i>f</i>	<i>c</i>
0	2	---	1 si $x < s$
1	0	$r + 1$	---
2	0	$d + 2$	---
3	1	$s + d + 1$	---

Las señales de control deben ser generadas
Su temporización y valores dependen de
la estructura elegida

● Proceso:

while fin = '0' loop

sel <= '1'; *oper* <= '0'; *esc_c* <= '0'; *esc_r* <= '1'; *esc_d* <= '0'; *esc_s* <= '0'; *sync*;

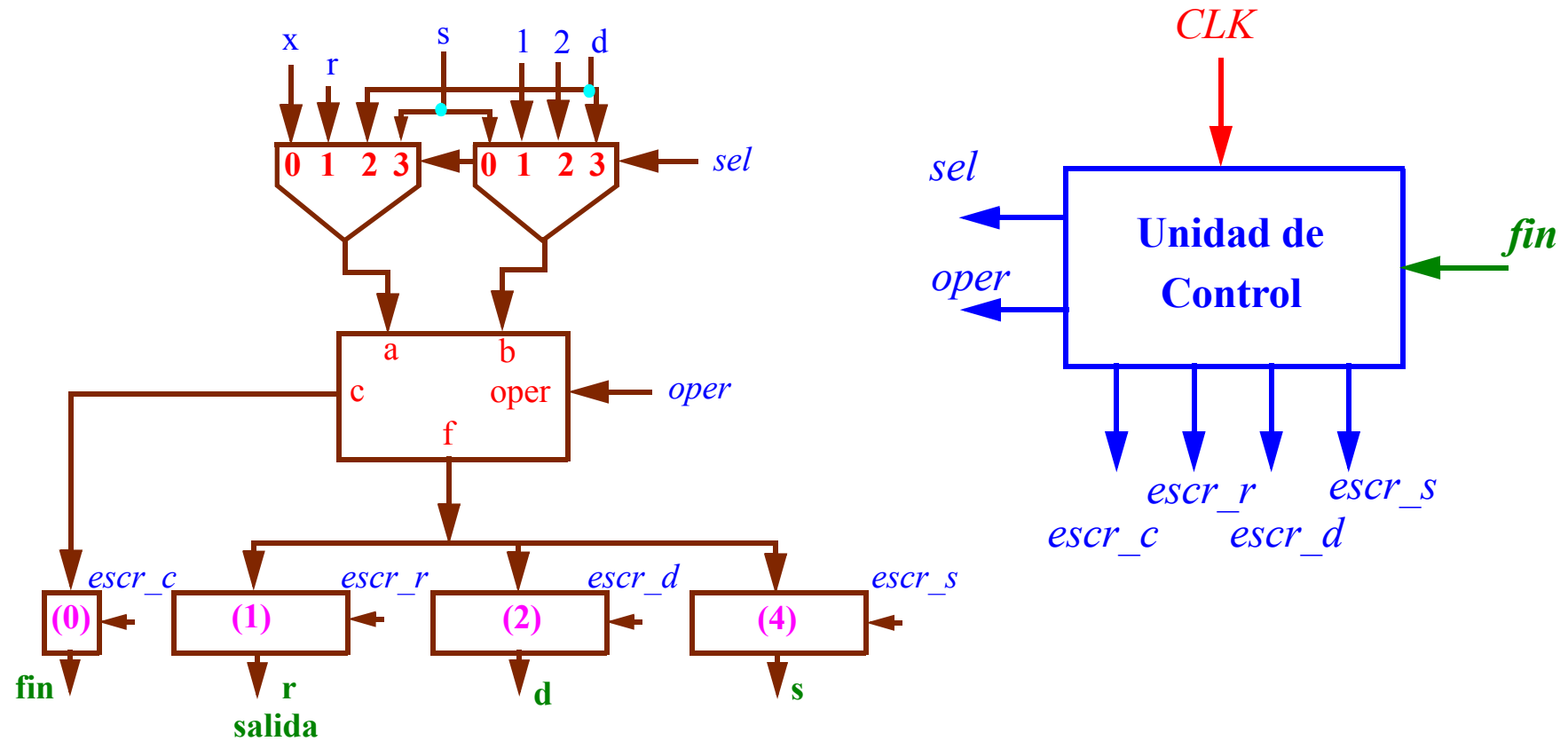
sel <= '2'; *oper* <= '0'; *esc_c* <= '0'; *esc_r* <= '0'; *esc_d* <= '1'; *esc_s* <= '0'; *sync*;

sel <= '3'; *oper* <= '1'; *esc_c* <= '0'; *esc_r* <= '0'; *esc_d* <= '0'; *esc_s* <= '1'; *sync*;

sel <= '0'; *oper* <= '2'; *esc_c* <= '1'; *esc_r* <= '0'; *esc_d* <= '0'; *esc_s* <= '0'; *sync*;

end loop;

Ejemplo de Descripción



Ejemplo de Descripción

while fin = '0' loop

S₀ *sel* <= '1'; *oper* <= '0'; *esc_c* <= '0'; *esc_r* <= '1'; *esc_d* <= '0'; *esc_s* <= '0'; *sync*;

S₁ *sel* <= '2'; *oper* <= '0'; *esc_c* <= '0'; *esc_r* <= '0'; *esc_d* <= '1'; *esc_s* <= '0'; *sync*;

S₂ *sel* <= '3'; *oper* <= '1'; *esc_c* <= '0'; *esc_r* <= '0'; *esc_d* <= '0'; *esc_s* <= '1'; *sync*;

S₃ *sel* <= '0'; *oper* <= '2'; *esc_c* <= '1'; *esc_r* <= '0'; *esc_d* <= '0'; *esc_s* <= '0'; *sync*;

S₄ *end loop*;

<i>Estado</i>	<i>fin</i> = 0	<i>fin</i> = 1
<i>s₀</i>	<i>s₁</i> /100100	<i>s₄</i> /--0000
<i>s₁</i>	<i>s₂</i> /200010	<i>s₂</i> /200010
<i>s₂</i>	<i>s₃</i> /310001	<i>s₃</i> /310001
<i>s₃</i>	<i>s₀</i> /021000	<i>s₀</i> /021000
<i>s₄</i>	<i>s₄</i> /--0000	<i>s₄</i> /--0000

Requisitos de un lenguaje de Descripción de Hardware

Capacidad de describir:

- ☐ *Bloques Funcionales, de los que se conoce su operación*
- ☐ *Bloques Estructurales, de los que se conoce su estructura interna*
- ☐ *Bibliotecas*
- ☐ *Conexiones entre bloques*

Capacidad de describir:

- ☐ *Algoritmos que relacionen E/S de los bloques funcionales*
- ☐ *Bibliotecas, que continen bloques funcionales*
- ☐ *Jerarquías, de manera que un bloque estructural pueda ser considerado un componente descrito a un nivel inferior*

Se puede asociar un Proceso a cada Bloque:

- ☐ *Si es Funcional, describe la relación entre entradas y salidas*
- ☐ *Si es Estructural, puede arrancar otros procesos de nivel inferior*
- ☐ *La comunicación entre Procesos se hace a través de Señales*
- ☐ *Las Señales entran o salen de los Bloques a través de Puertos*

Los Bloques se encapsulan en Entidades, caracterizadas por:

- ☐ *Sus Puertos (señales “visibles” desde fuera de la entidad)*
- ☐ *Su arquitectura o descripción interna*

Ejemplo de Descripción

- VHDL:

- ❑ Librerías

library nombre; use paquete;

- ❑ descripción del sistema

entity componente is
declarar variables (entrada/salida, #bits,...)
end componente;

- ❑ descripción de su operación

architecture flujo of componente is
declarar señales (nombre, tipo, #bits,...)
begin
process
programa de operación
end process
end flujo;

- ❑ Esta descripción puede ser Recurrente

Ejemplo 1: Tabla de Estados

X_1X_0	00	01	10	11
A	A/0	B/0	Q/1	--
B	P/1	B/0	--	C/0
C	--	S/1	D/0	C/0
D	A/0	--	D/0	R/1
P	P/1	B/0	Q/1	--
Q	A/0	--	Q/1	R/1
R	--	S/1	D/0	R/1
S	P/1	S/1	--	C/0

Ejemplo #1: Tabla de Estados

X_1X_0	00	01	10	11
AQ	AQ/0	BP/0	AQ/1	DR/1
BP	BP/1	BP/0	AQ/1	CS/0
CS	BP/1	CS/1	DR/0	CS/0
DR	AQ/0	CS/1	DR/0	DR/1

Equivalencias:

A-Q, B-P, C-S, D-R

Ejemplo #1: Descripción VHDL

```

library ieee; use ieee.std_logic_1164.all;
entity ejemplo#1 is
    port (x1, x0, clk, reset: in std_logic; z: out std_logic);
end ejemplo#1;

architecture funcional of ejemplo#1 is
    type estado is (AQ, BP, CS, DR);
    signal presente: estado := AQ;
begin
    process (reset, clk, presente)
    begin
        case presente is
            when AQ => z <= x1;
            when BP => z <= not(x0);
            when CS => z <= not(x1);
            when DR => z <= x0;
        end case;
    end process;
end funcional;

```

Diagram illustrating the state transitions for the VHDL entity `ejemplo#1`:

```

graph TD
    subgraph "if reset = '1' then presente <= AQ;"
        direction TB
        S1[if reset = '1' then presente <= AQ;]
    end
    subgraph "elsif clk'event and clk = '1' then"
        direction TB
        S2[elsif clk'event and clk = '1' then]
        S3[case presente is]
        S4[when AQ =>]
        S5[if x1 = '0' and x0 = '1' then presente <= BP;]
        S6[elsif x1 = '1' and x0 = '1' then presente <= DR; end if;]
        S7[when BP =>]
        S8[if x1 = '1' and x0 = '0' then presente <= AQ;]
        S9[elsif x1 = '1' and x0 = '1' then presente <= CS; end if;]
        S10[when CS =>]
        S11[if x1 = '0' and x0 = '0' then presente <= BP;]
        S12[elsif x1 = '1' and x0 = '0' then presente <= DR; end if;]
        S13[when DR =>]
        S14[if x1 = '0' and x0 = '0' then presente <= AQ;]
        S15[elsif x1 = '0' and x0 = '1' then presente <= CS; end if;]
        S16[end case;]
        S17[end if;]
        S18[end process;]
        S19[end funcional;]
    end
    S1 --> S2
    S2 --> S3
    S3 --> S4
    S4 --> S5
    S5 --> S6
    S6 --> S7
    S7 --> S8
    S8 --> S9
    S9 --> S10
    S10 --> S11
    S11 --> S12
    S12 --> S13
    S13 --> S14
    S14 --> S15
    S15 --> S16
    S16 --> S17
    S17 --> S18
    S18 --> S19

```


Ejemplo #1: Descripción VHDL de una Prueba

```
library ieee; use ieee.std_logic_1164.all;  
entity test_ejemplo#1 is end test_ejemplo#1;  
  
architecture test of test_ejemplo#1 is  
    component ejemplo#1  
        port (x1, x0, clk, reset: in std_logic; z: out std_logic);  
    end component;  
for all: ejemplo#1 use entity work.ejemplo#1(funcional);  
    signal x1, x0, reset, z: std_logic;  
    signal clk: std_logic := '0';  
begin  
        etiqueta: ejemplo#1 port map (x1, x0, clk, reset, z);  
        clk <= not (clk) after 50 ns;  
        reset <= '1', '0' after 100 ns;  
        x1 <= '0', '1' after 200 ns, '0' after 400 ns, '1' after 600 ns, '0' after 800 ns,  
            '1' after 1000 ns, '0' after 1300 ns, '1' after 1500 ns, '0' after 1700 ns, '1' after 1900 ns;  
        x0 <= '0', '1' after 300 ns, '0' after 500 ns, '1' after 700 ns, '0' after 900 ns,  
            '1' after 1100 ns, '0' after 1200 ns, '1' after 1400 ns, '0' after 1600 ns, '1' after 1800 ns;  
end test;
```

Ejemplo de Descripción: Datos

```
library ieee; use ieee.std_logic.all;;  
  
entity estructura_de_datos is  
    port (reset, clk, esc_c, esc_r, esc_d, esc_s: in std_logic;  
          sel, oper: in std_logic_vector(1 downto 0); x: in unsigned(7 downto 0);  
          resultado: out unsigned(7 downto 0); fin: out std_logic);  
end estructura_de_datos;  
  
architecture flujo of estructura_de_datos is  
    signal a, b, f, r, d, s: unsigned (7 downto 0);  
    signal c: std_logic;  
  
begin  
    r <= '00000001' when reset = '1' else f when clk = '1' and esc_r = '1';  
    d <= '00000010' when reset = '1' else f when clk = '1' and esc_d = '1';  
    s <= '00000100' when reset = '1' else f when clk = '1' and esc_s = '1';  
    fin <= '0' when reset = '1' else c when clk = '1' and esc_c = '1';  
    with sel select a <= x when '00', r when '01', d when '10', s when others;  
    with sel select b <= s when '00', '00000001' when '01', '00000010'  
        when '10', d when others;  
    with oper select f <= a+b when '00', a+b+1 when '01', a+b when others;  
    c <= '1' when a < b else '0';  
    resultado <= r;  
  
end flujo;
```

Ejemplo de Descripción: Control

```
library ieee; use ieee.std_logic.all;;  
entity unidad_de_control is  
    port (reset, clk: in std_logic; esc_c, esc_r, esc_d, esc_s: out std_logic;  
        sel, oper: out std_logic_vector(1 downto 0); fin: in std_logic;;  
        resultado: out unsigned(7 downto 0); fin: out std_logic);  
end unidad_de_control;  
architecture máquina_de_estados of unidad_de_control is  
    type estado is (s0, s1, s2, s3, s4);  
    signal presente: estado;  
  
    begin  
        .  
        .  
        .  
end máquina_de_estados;
```

(ver transparencia siguiente)

Ejemplo de Descripción: Control

```
library ieee; use ieee.std_logic.all;;  
entity unidad_de_control is  
    port (reset, clk: in std_logic; esc_c, esc_r, esc_d, esc_s: out std_logic;  
        sel, oper: out std_logic_vector(1 downto 0); fin: in std_logic;;  
        resultado: out unsigned(7 downto 0); fin: out std_logic);  
end unidad_de_control;  
architecture máquina_de_estados of unidad_de_control is  
    type estado is (s0, s1, s2, s3, s4);  
    signal presente: estado;  
  
    begin  
        .  
        .  
        .  
end máquina_de_estados;
```

(ver transparencias siguientes)

Ejemplo de Descripción: Control (parte 1)

begin

process (reset, clk, presente)

begin

case presente is

when S0 =>

*if fin = '0' then sel <= '01'; oper <= '00'; esc_c <= '0'; ; esc_r <= '1'; esc_d <= '0';
esc_s <= '0';*

else sel <= '00'; oper <= '00'; esc_c <= '0'; ; esc_r <= '0'; esc_d <= '0'; esc_s <= '0';

end if;

when S1 =>

sel <= '10'; oper <= '00'; esc_c <= '0'; ; esc_r <= '0'; esc_d <= '1'; esc_s <= '0';

when S2 =>

sel <= '11'; oper <= '01'; esc_c <= '0'; ; esc_r <= '0'; esc_d <= '0'; esc_s <= '1';

when S3 =>

sel <= '00'; oper <= '10'; esc_c <= '1'; ; esc_r <= '0'; esc_d <= '0'; esc_s <= '0';

when S4 =>

sel <= '00'; oper <= '00'; esc_c <= '0'; ; esc_r <= '0'; esc_d <= '0'; esc_s <= '0';

end case

(sigue en la siguiente transparencia)

Ejemplo de Descripción: Control (parte 2)

```
end case  
if reset= '1' then presente <= S0; elsif clk = '1'  
then  
  case presente is  
    when S0 => if fin = '0' then presente <= S1;  
    else presente <= S4; end if;  
    when S1 => presente <= S2;  
    when S2 => presente <= S3;  
    when S3 => presente <= S0;  
    when S4 => presente <= S4;  
  end case  
end if  
end process  
end máquina_de_estados;
```

Ejemplos

Contador de 20

Contador programable de 5 a 30

Contador bidireccional

Contador bidireccional y programable de 4 a 40

Ejemplo de Filtro

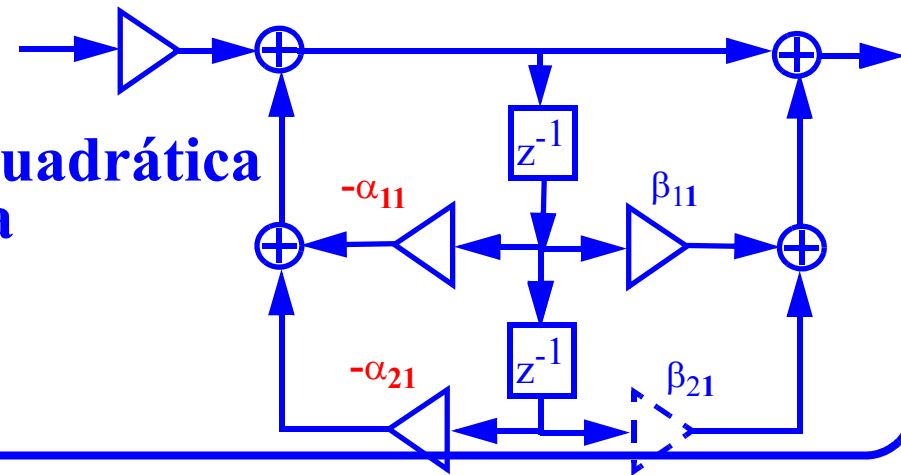
Problema:

Diseñar un filtro con función de transferencia

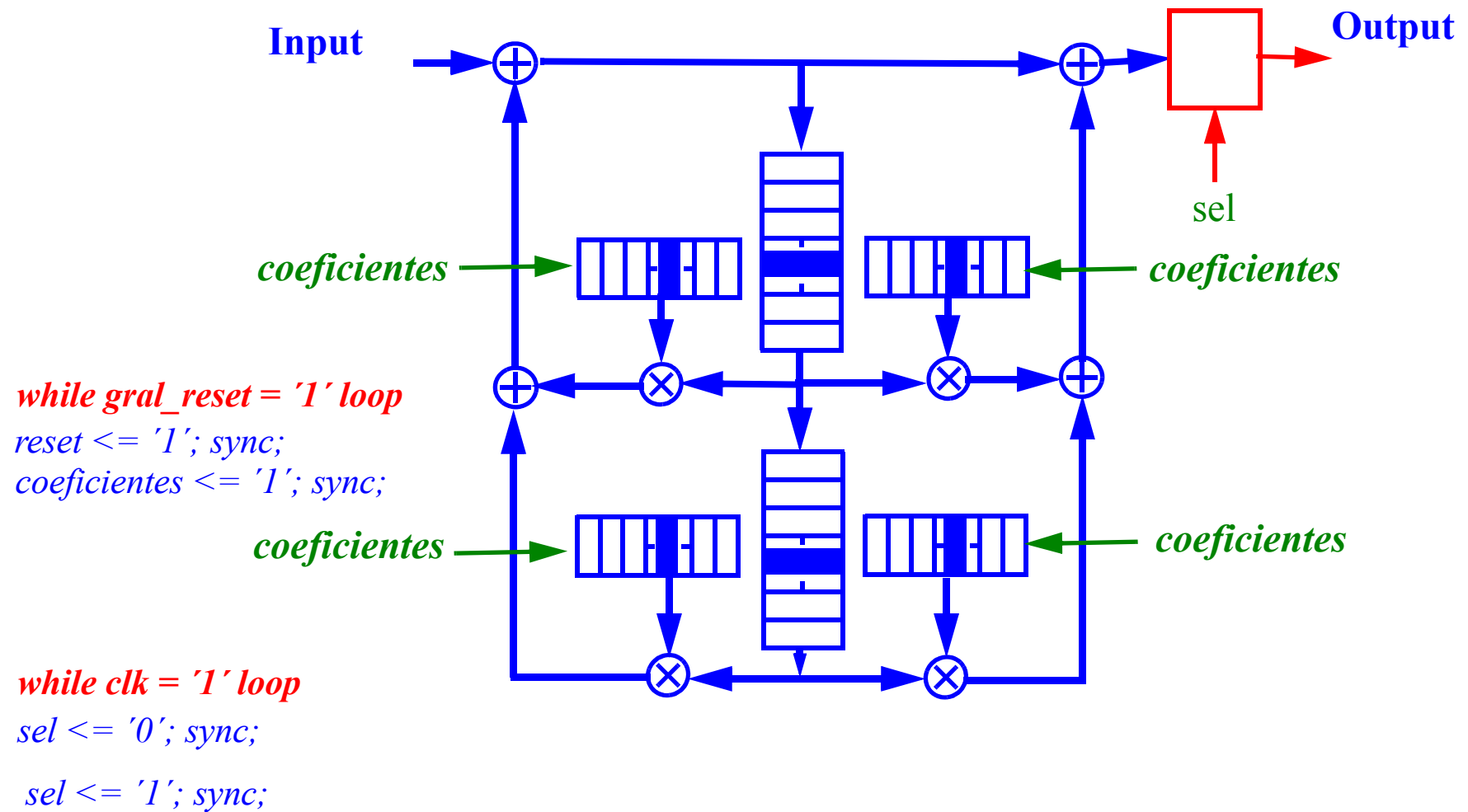
$$H(z) = \frac{(0.1z^{-1})(0.15)}{(0.15 + 0.1z^{-1} - 0.05z^{-2})(-1 + 0.75z^{-1} + 0.25z^{-2})}$$

$$y(n) = \sum_{m=0}^M b_m x(n-m) - \sum_{k=1}^N a_k y(n-k)$$

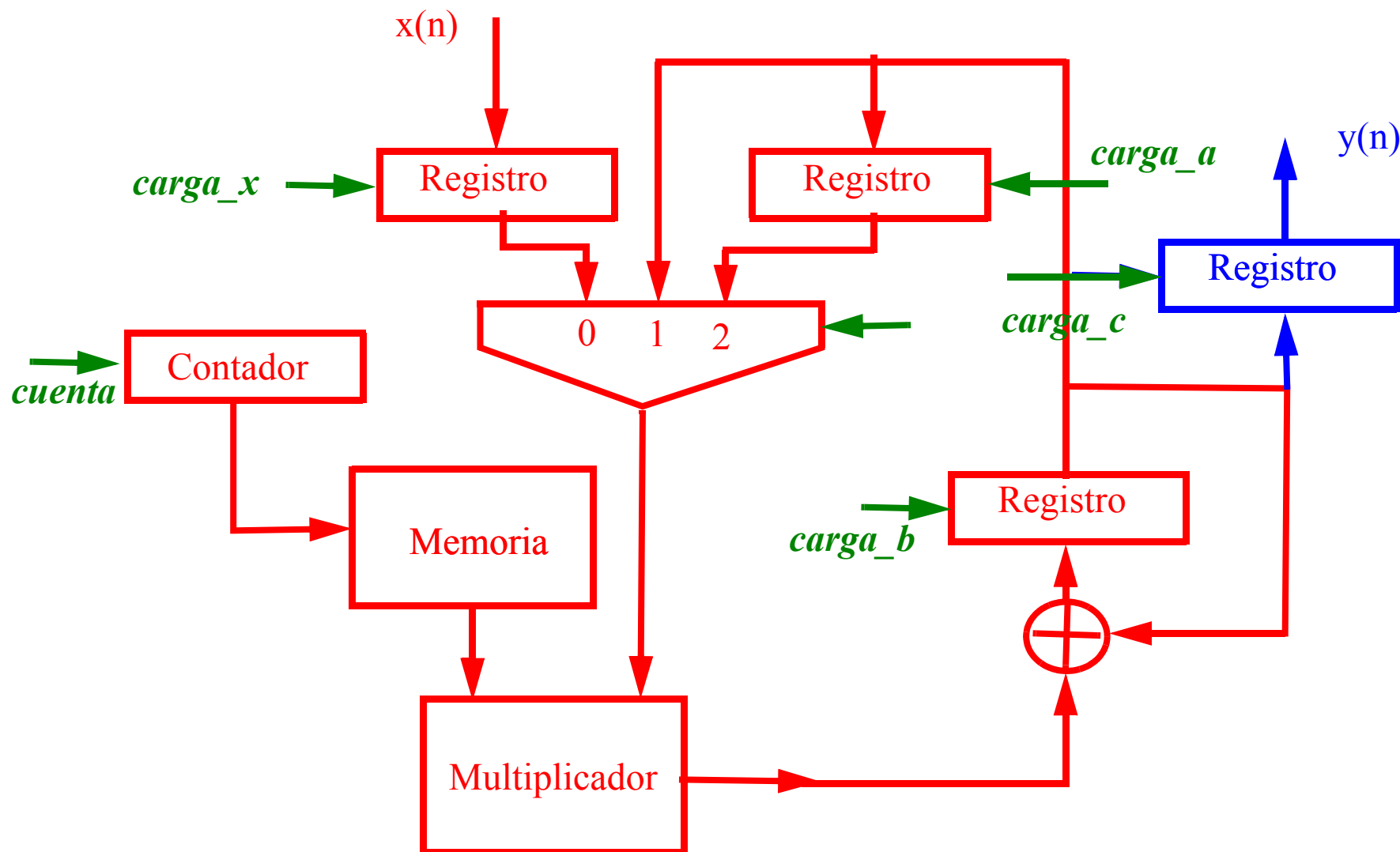
Multiplexando una estructura biquadrática usando codificación binaria pura y una precisión de 8 bits



Ejemplo de Filtro



Ejemplo de Filtro: Estructura de Datos



Ejemplo de Filtro: Estructura de Datos

$$H(z) = \frac{(0.1z^{-1})(0.15)}{(0.15 + 0.1z^{-1} - 0.05z^{-2})(-1 + 0.75z^{-1} + 0.25z^{-2})}$$

$$y_1(n) = 0.1x(n-1) - (0.1y_1(n-1)) + 0.05y_1(n-2)$$

$$y_2(n) = -0.15y_1(n) + 0.75y_2(n-1) + 0.25y_2(n-2)$$

