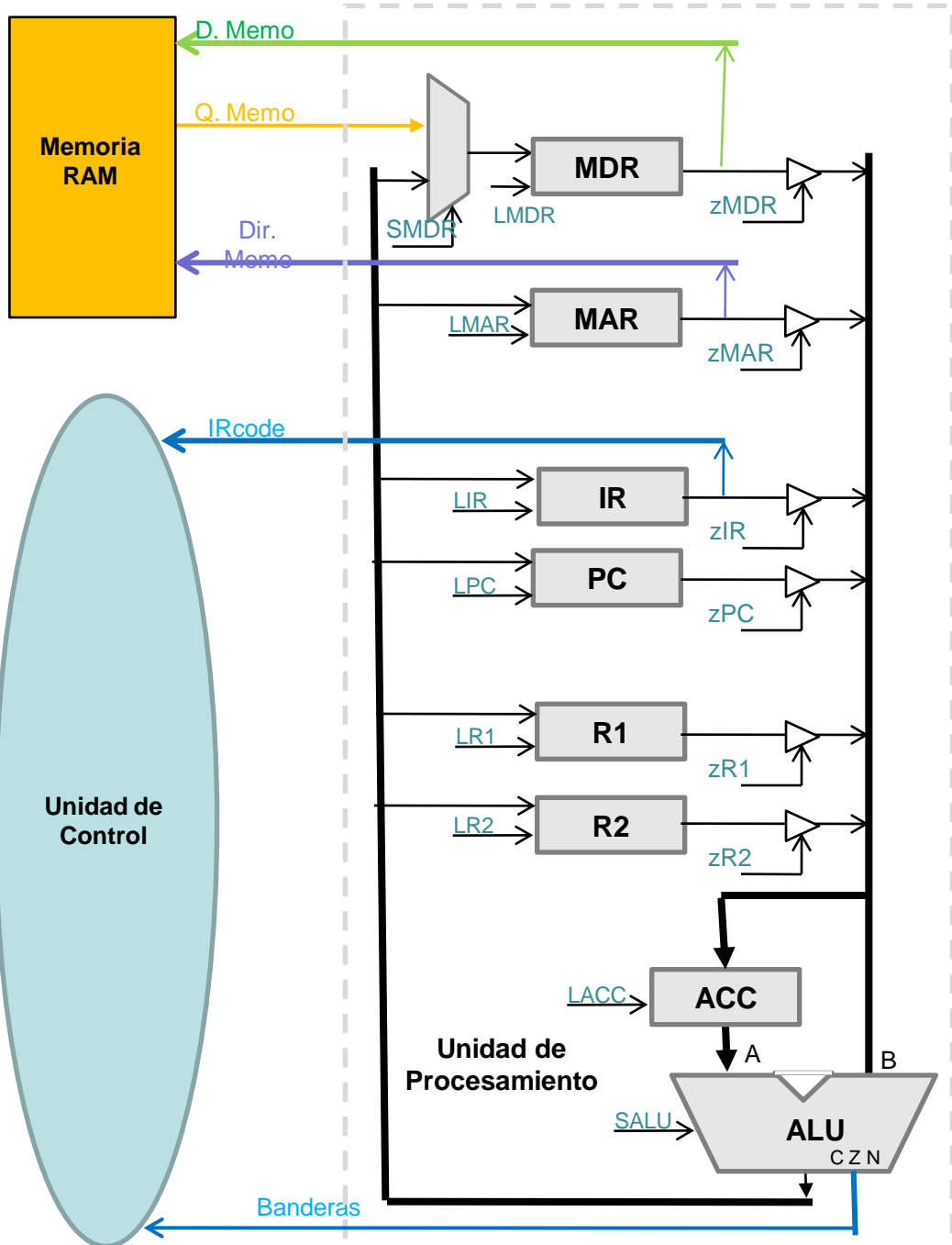


SISTEMAS DIGITALES VHDL

Fredy Campiño

Arquitectura del Micro-procesador



Arquitectura

Hay dos tipos de descripción para la arquitectura:

- **Funcional (o de comportamiento):** La descripción se basa en la tarea o funcionamiento del dispositivo. Se utilizan **procedimientos y**
- **Estructural :** la descripción se basa en la interconexión de los elementos que componen el dispositivo. Se utilizan **componentes y/o paquetes**

Componente

- Los componentes en VHDL proporcionan una forma de especificar un diseño en forma jerárquica.
- La declaración de un componente proporciona información acerca de la interface externa de un componente que incluye entradas y salidas y parámetros genéricos.
- La declaración es similar a la declaración de una entidad y se realiza en el mismo espacio donde se declaran las señales.
- Los componentes también se declaran en los paquetes.

Declaración componente

```
ARCHITECTURE a OF __entity_name IS  
    -- Definición de señales
```

```
    component <component_name>  
    generic (  
        generic_declaration;  
        generic_declaration;  
        ...);  
    port (  
        port-declaration;  
        port-declaration;  
        ....);
```

```
BEGIN  
    -- Definición de la arquitectura
```

```
END a;
```

Instanciación componente

```
ARCHITECTURE a OF __entity_name IS
```

```
-- Definición de señales
```

```
-- Definición de componentes
```

```
.....
```

```
BEGIN
```

```
Instance_label: <component-name>
```

```
  generic map(
```

```
    generic-association;
```

```
    generic -association
```

```
  )
```

```
  port map(
```

```
    port-association;
```

```
    port-association
```

```
  )
```

```
END a;
```

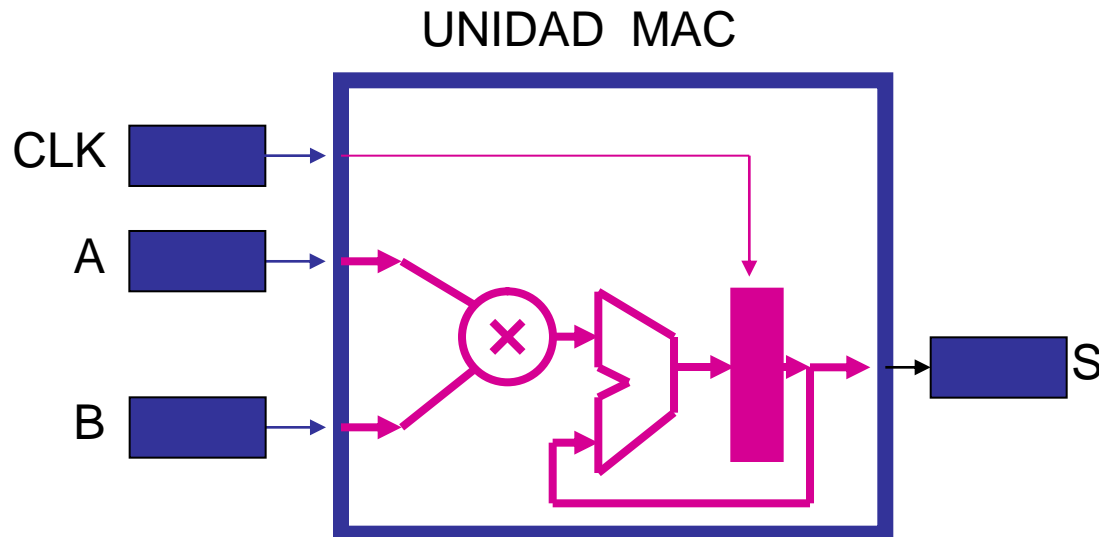
Arquitectura Estructural

Unidad MAC: Estructural

EJEMPLO:

Implemente la arquitectura en VHDL para una unidad MAC:

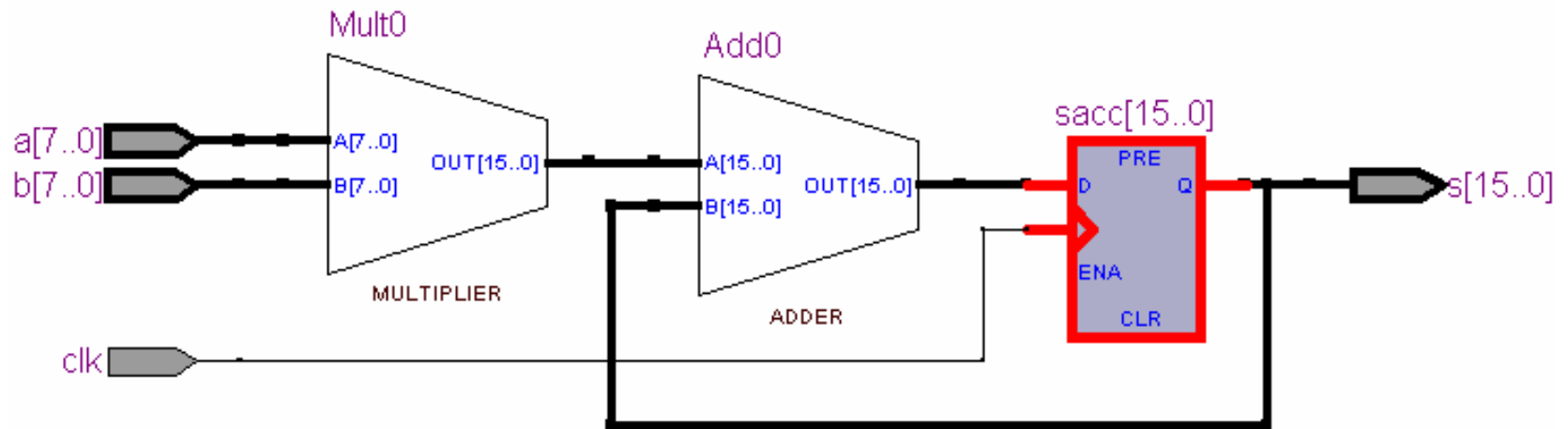
1. *Enfoque estructural* (proyecto mac2)



Arquitectura funcional unidad MAC1-VHDL

```
23 ARCHITECTURE funcional OF mac1 IS
24 -- CONSTANTES, SEÑALES
25 SIGNAL sacc,sadd: STD_LOGIC_VECTOR(ancho_bits*2-1 downto 0);
26
27 BEGIN
28 unidad_mac: PROCESS (clk,a,b,sacc,sadd)
29 -- VARIABLES
30
31 BEGIN
32     if clk'event and clk='1' then
33         sacc<= std_logic_vector(signed(a)*signed(b) + signed(sacc));
34     end if;
35     s<=sacc;
36 END PROCESS unidad_mac;
37
38 END funcional;
```

Arquitectura funcional unidad MAC1-RTL síntesis

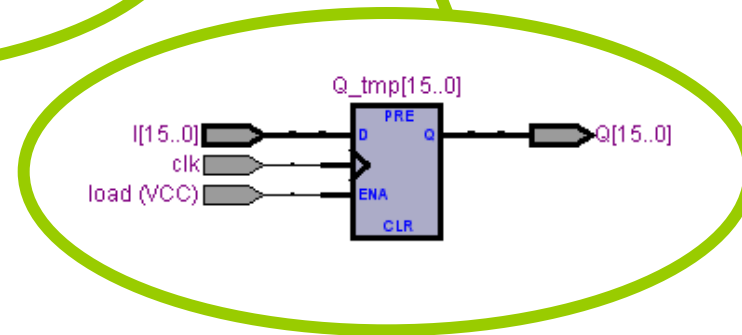
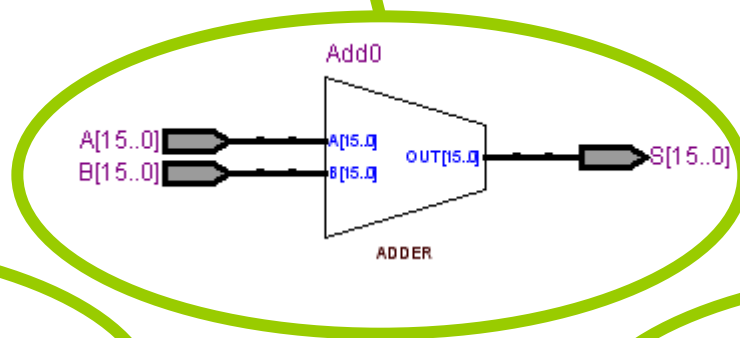
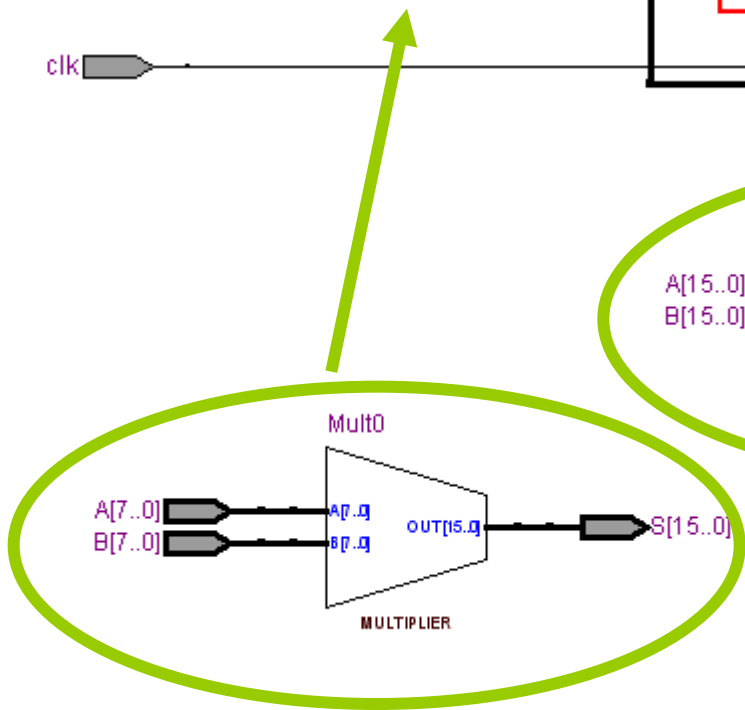
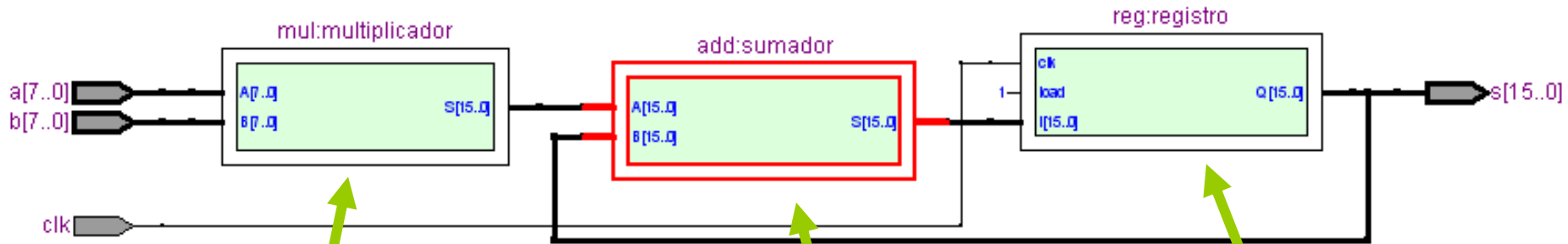


Arquitectura estructural unidad MAC2-VHDL

Instancia del
componente **mul**

```
24 ARCHITECTURE estructural OF mac2 IS
25 -- CONSTANTES, SEÑALES
26 SIGNAL smul,sadd, sreg: STD_LOGIC_VECTOR(ancho_bits*2-1 downto 0);
27 BEGIN
28
29 -- CONEXION MULTIPLICADOR
30 multiplicador: mul
31 GENERIC MAP(width_dataa =>ancho_bits,width_datab =>ancho_bits)
32
33 PORT MAP (A=>a, B=>b, S=>smul);
34
35 -- CONEXION SUMADOR
36 sumador: add
37 GENERIC MAP(width_dataa =>ancho_bits*2,width_datab =>ancho_bits*2)
38
39 PORT MAP (A=>smul, B=>sreg, S=>sadd);
40
41 --CONEXION REGISTRO ACUMULADOR
42 registro: reg
43 GENERIC MAP(n=>ancho_bits*2)
44 PORT MAP (I=>sadd,clk=>clk,load=>'1',Q=>sreg);
45
46 s<=sreg;
47
48 END estructural;
```

Arquitectura estructural unidad MAC2-VHDL



Simulaciones MAC1 y MAC2

