

Electrónica Digital 2

Maquinas de Estado algorítmico ASM

Ferney Alberto Beltrán Molina

`figs/logo-unal.jpg`

Agosto 2019

Contacto

Nombre: Ferney Alberto Beltrán Molina, Ing, MSc, PhD(c)
Email: fabeltranm@unal.edu.co
oficina:

Contenido

Introducción al Diseño Digital

Maquinas de Estado Algorítmico

Índice

Introducción al Diseño Digital

Maquinas de Estado Algorítmico

Dominios descriptivos

- ▶ **Representación funcional o de comportamiento**

Especifica el comportamiento o la función de un diseño sin información de aplicación.

La función realizada sin información sobre cómo se hace.

- ▶ **Representación estructural**

Especifica la implementación de un diseño en términos de componentes y sus interconexiones

Los bloques y las interconexiones (netlist o esquemas)

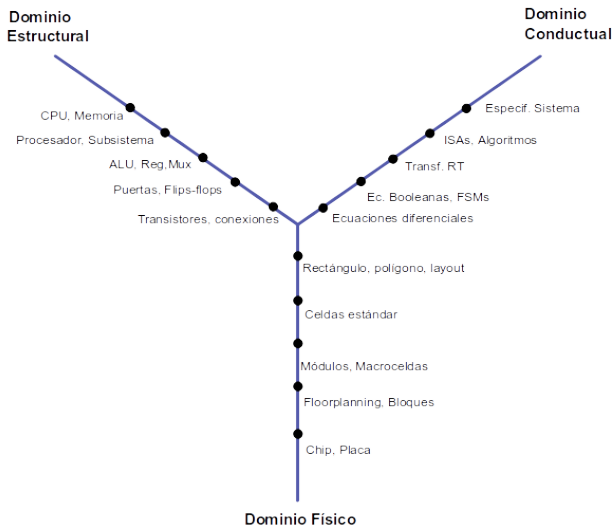
- ▶ **Representación física**

Especifica las características físicas del diseño

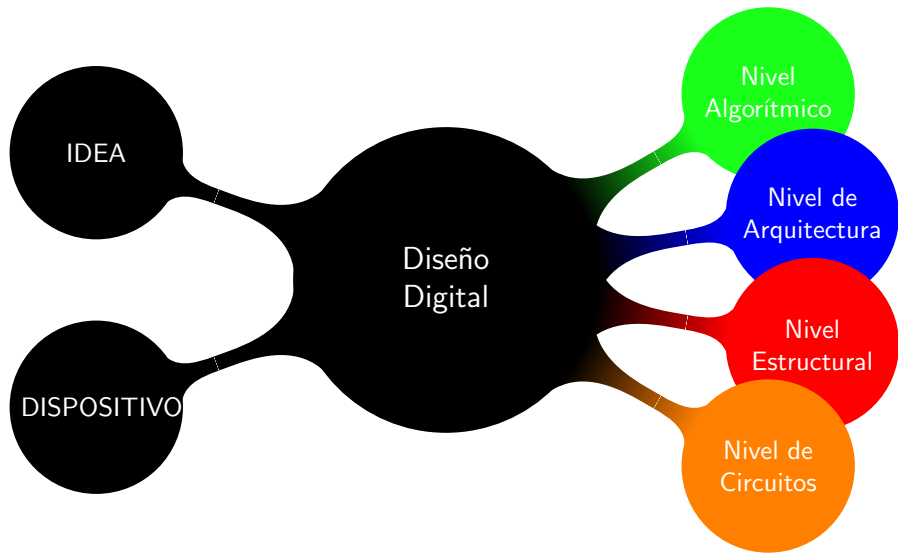
Localización y propiedades físicas reales

Dominios descriptivos

Diagrama Y de Gajsky-Khun

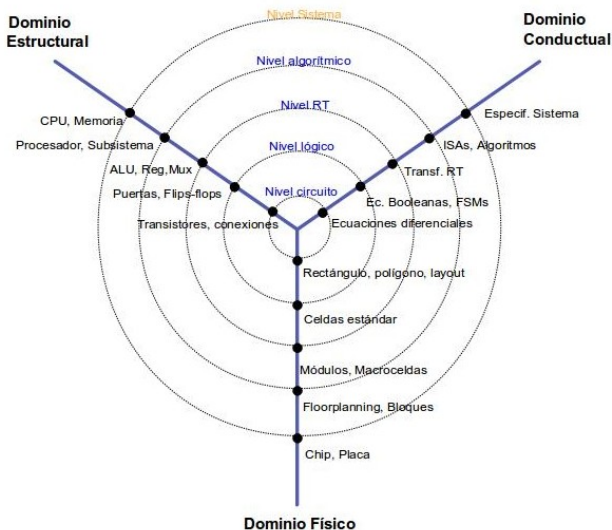


Abstracción



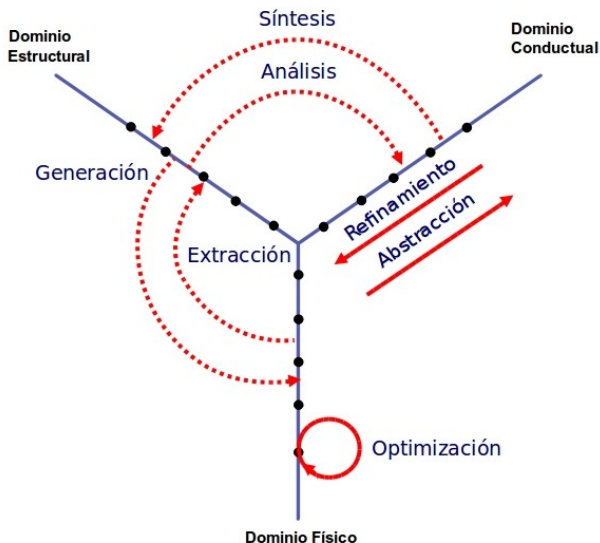
Dominios descriptivos / Nivel de abstracción

Diagrama Y de Gajsky-Khun



Dominios descriptivos / Transiciones

Diagrama Y de Gajsky-Khun

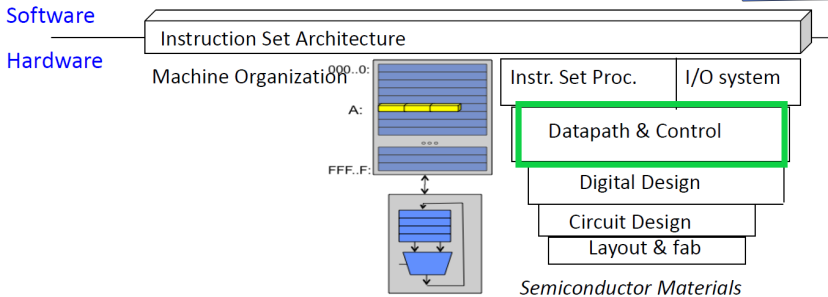
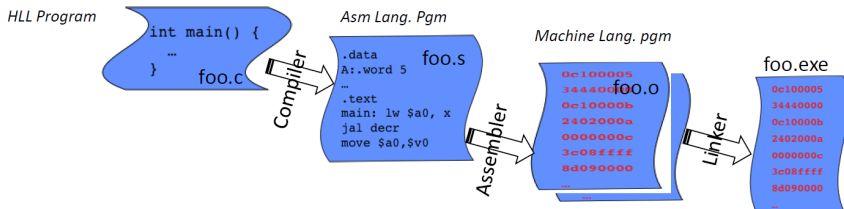


Índice

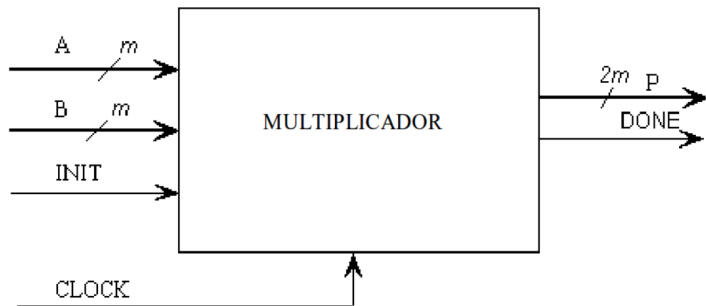
Introducción al Diseño Digital

Maquinas de Estado Algorítmico

Introducción al Datapath



Multiplicador NxM



El algoritmo de multiplicación que se implementa se basa en productos parciales (PP).

Multiplicador NxM

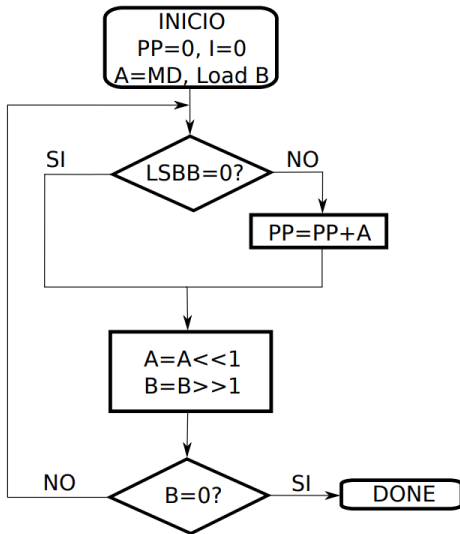
Se realiza la multiplicación iniciando con el bit menos significativo del multiplicador, el resultado de la multiplicación se suma al primer producto parcial y se obtiene el segundo producto parcial; si el bit del multiplicador es 0 no se afecta el contenido de PP, por lo que no se realiza la suma.

A continuación se realiza la multiplicación del siguiente bit (a la izquierda del LSB) y el resultado se suma al producto parcial pero corrido un bit a la izquierda. Este proceso continua hasta completar todos los bits del multiplicador y el último producto parcial es el resultado fina

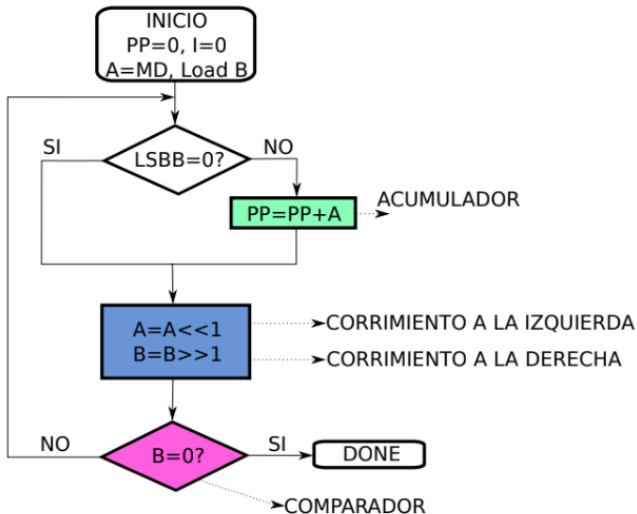
Multiplicador NxM

$$\begin{array}{r} 1010 \\ \times 0101 \\ \hline 0000 \leftarrow \text{Primer producto parcial} \\ 1010 \\ \hline 1010 \leftarrow \text{Segundo producto parcial} \\ 0000 \\ \hline 01010 \leftarrow \text{Tercer producto parcial} \\ 1010 \\ \hline 110010 \leftarrow \text{Cuarto producto parcial} \\ 0000 \\ \hline 110010 \leftarrow \text{Resultado} \end{array}$$

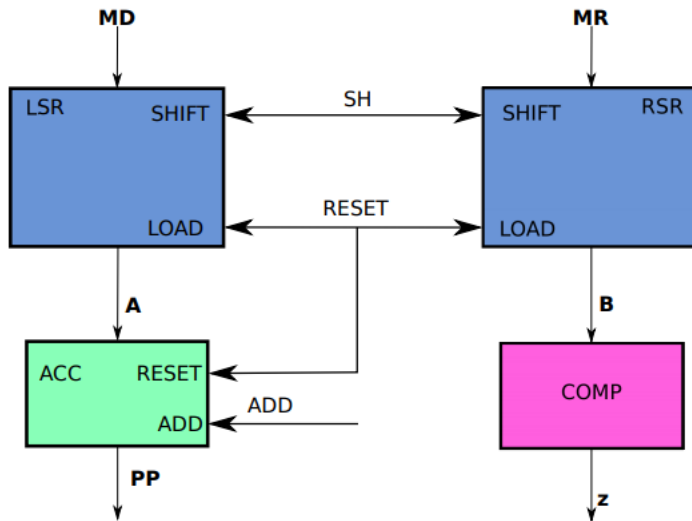
Multiplicador NxM: Descripción Funcional - diagrama de flujo



Multiplicador NxM: identificación de componentes



Multiplicador NxM: DATAPATH

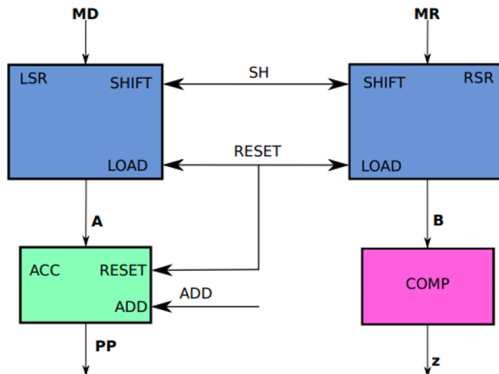


Multiplicador NxM: DATAPATH-verilog

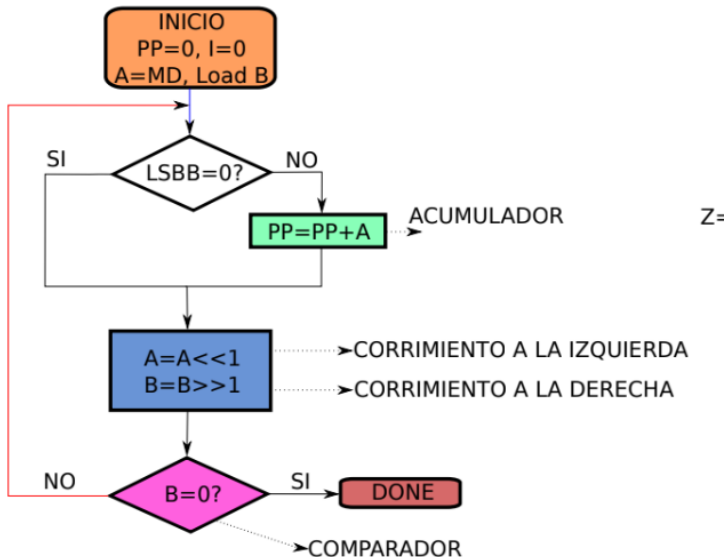
```
always @(posedge clk) begin
    if (rst) begin
        A = (3'b000,MD);
        B = MR;
    end
    else begin
        if (sh) begin
            A = A << 1;
            B = B >> 1;
        end
    end
end
```

```
always @(posedge clk) begin
    if (rst) begin
        pp = 0;
    end
    else begin
        if (add) begin
            pp = pp + A;
        end
    end
end
```

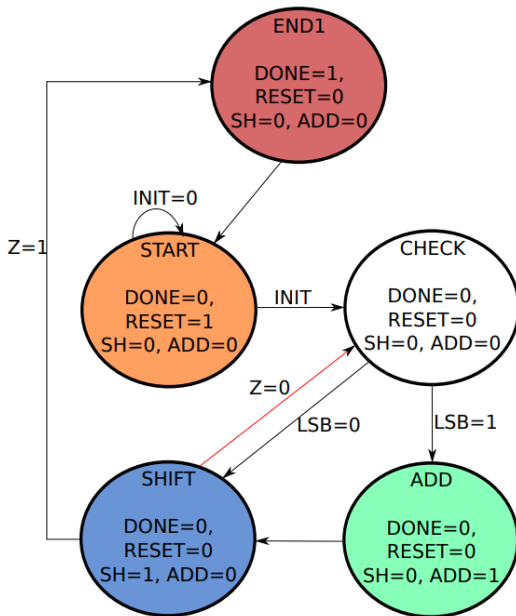
```
assign z = (B==0) ? 1:0;
```



Multiplicador: se identifica la unidad Control



Multiplicador: Máquina de estados Finitos (FSM) - Control



Multiplicador: FSM - Control-verilog

```
parameter START =0, CHECK =1, ADD =2, SHIFT =3, END1 =4;
```

```
always @(posedge clk) begin
```

```
  case (status)
```

```
    START: begin
```

```
      sh=0; add=0;
```

```
      if (init) begin ← INIT
```

```
        status=CHECK;
```

```
        done =0; rst=1;
```

```
      end
```

```
    end
```

```
    CHECK: begin
```

```
      done=0; rst=0; sh=0;
```

```
      add=0;
```

```
      if (B[0]==1) ← LSB
```

```
        status=ADD;
```

```
      else
```

```
        status=SHIFT;
```

```
      end
```

```
    ADD: begin
```

```
      done=0; rst=0;
```

```
      sh=0; add=1;
```

```
      status=SHIFT;
```

```
    end
```

```
    SHIFT: begin
```

```
      done=0; rst=0;
```

```
      sh=1; add=0;
```

```
      if (z==1) ← Z
```

```
        status=END1;
```

```
      else
```

```
        status=CHECK;
```

```
      end
```

```
    END1: begin
```

```
      done =1; rst =0;
```

```
      sh =0; add =0;
```

```
      status =START;
```

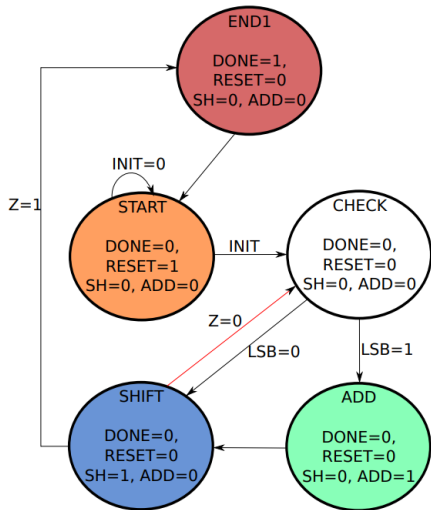
```
    end
```

```
  default:
```

```
    status =START;
```

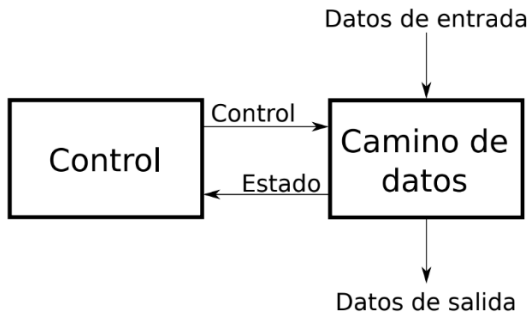
```
endcase
```

```
end
```



Proceso Realizado

- ▶ 1. Se elabora un diagrama de flujo que describa la funcionalidad deseada ya sea a nivel gráfico o en texto.
- ▶ 2. Se identifica los componentes del DataPath.
- ▶ 3. Se identifican las señales necesarias para controlar el Datapath y la interconexión.
- ▶ 4. Se especifica de la unidad de control (FSM) utilizando diagramas de estado.
- ▶ 5. En laboratorio:
 - ▶ Se implementan los componentes del DataPath y de la unidad de control utilizando HDL.
 - ▶ Simulación y pruebas.



ASM - Multiplicación

