Pontificia Universidad Católica de Chile
Escuela de Ingeniería
Departamento de Ingeniería Eléctrica
IEE2713 - Sistemas Digitales

Ayudantía 10

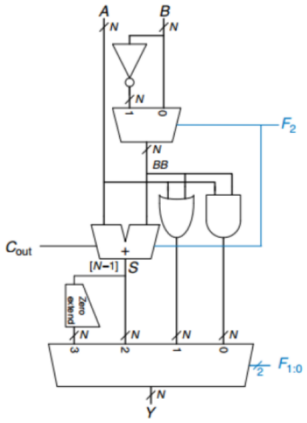
Mathías Lambert(mglambert@uc.cl)

14 de Noviembre de 2018

Problema 1

Observe la Unidad Lógica Aritmética que se presenta a continuación. Considere que la palabra F[2 : 0] es la instrucción sobre cómo operar A y B.

¿Qué operaciones puede realizar esta ALU?



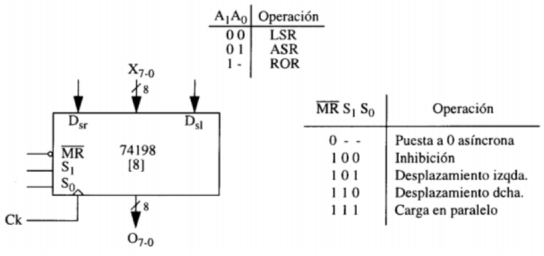
Considere una ALU implementada en un CI que acepta entradas de 4 bits, muestre como implementar una ALU de 16 bits.

Problema 2

Se definen las siguientes operaciones de desplazamiento a la derecha:

- LSR(Logic Shift Right): El bit de entrada serie es 0.
- ASR (Aritmetic Shift Right): El bit de entrada serie es el bit de signo.
- ROR (Rotation Right): El bit de entrada serie es el bit de salida serie.

Realice estas 3 operaciones utilizando el circuito 74198. Luego diseñe un circuito que permita realizar estas 3 operaciones.



A_1A_0	Operación
0 0	LSR
0 1	ASR
1 -	ROR

$\overline{MR}$ S_1 S_0	Operación
0 - -	Puesta a 0 asíncrona
1 0 0	Inhibición
1 0 1	Desplazamiento izqda.
1 1 0	Desplazamiento dcha.
1 1 1	Carga en paralelo

Problema 3

Utilizando el CI anterior implemente un contador de 8 bits.

Problema 4

Para el código adjunto:

- Identifique el tipo de maquina.
- Dibuje el diagrama de bloques.
- Reescriba el código reduciendo la cantidad de estados.

Problema 5

Implemente en Verilog un contador de la secuencia de *Fibonacci*.

Código problema 4

```
module maquina_desconocida(input clk,
                           input reset,
                           input in,
                           output out);
    parameter z = 2'b00;
    parameter o = 2'b01;
    parameter t = 2'b10;

    reg out;
    reg [2:1] s, ns;

    always @(posedge clk) begin
        if (reset) s = z;
        else s = ns;
    end

    always @(in or s) begin
        case (state)
            z:
            begin
                if (in) ns = o;
                else ns = z;
            end
            o:
            begin
                if (in) ns = t;
                else ns = z;
            end
            t:
            begin
                if (in) ns = t;
                else ns = z;
            end
        endcase
    end

    always @(s) begin
        case (s)
            z: out = 0;
            o: out = 0;
            t: out = 1;
        endcase
    end
endmodule
```