ESTRUCTURA DE COMPUTADORES. GII-IC **Grupo 3. Primer control. Curso 2014-2015**

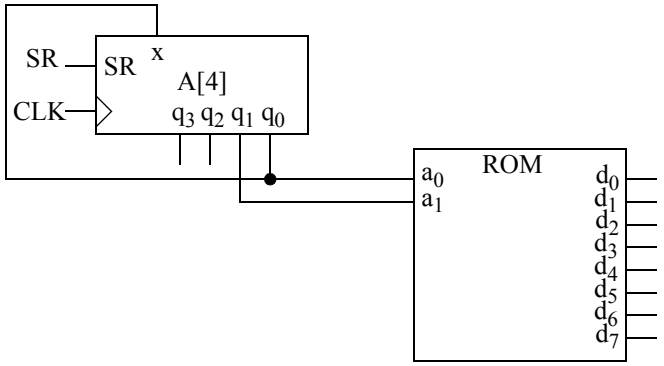
Apellidos:.....**SOLUCIÓN**.....

Nombre:.....

1	2	3

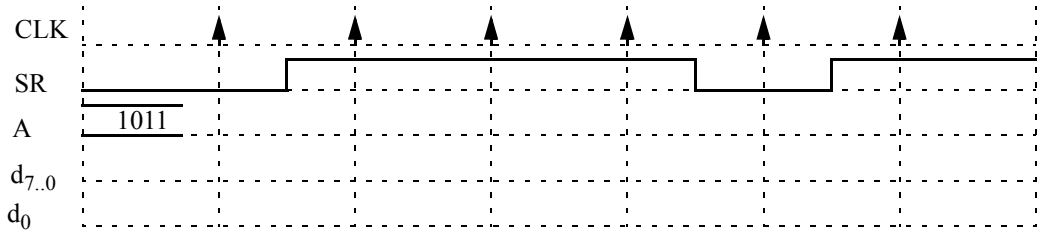
Duración 2:00 h

1.- **[3 puntos]** Inicialmente el registro A del circuito de la figura contiene el dato "1011". Complete el cronograma dando [A], $d_{7:0}$ en hexadecimal y dibujando la forma de onda de la salida d_0.

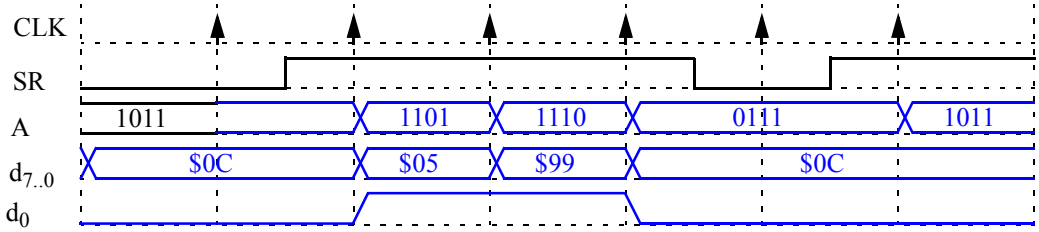


Contenido de ROM

$a_{1..0}$	$d_{7..0}$
00	$34
01	$05
10	$99
11	$0C



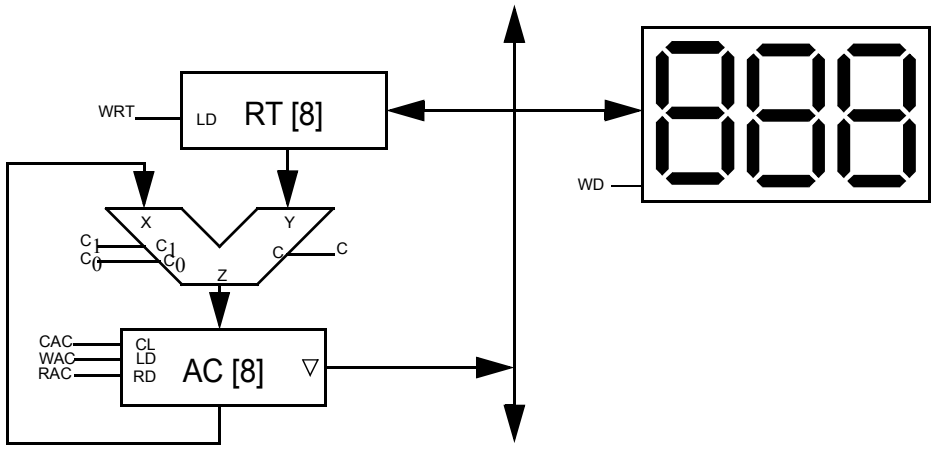
SOLUCIÓN



2.- **[4 puntos]** Se desea realizar un sistema digital a nivel RT que genere y muestre en un display los números de Fibonacci 1, 1, 2, 3, 5, 8, 13... Para ello se propone la Unidad de Datos de la figura: todos los dispositivos (registros, buses) son de 8 bits. RT tiene capacidad de carga en paralelo; AC puede ser puesto a cero, cargado en paralelo y dispone de dos buses de salida, uno con control de lectura. El display de siete segmentos puede ser escrito con WD y no cambia hasta que sea cargado con un nuevo dato. La ALU se describe en la tabla. Una vez mostrados todos los números posibles (menores que 256), el sistema se para activando la

señal FIN y queda a la espera de una nueva señal Xs.

c_1c_0	Op.
00	X
01	X+1
10	X+Y
11	X-Y



a) Describa, a nivel RT, los componentes de esta unidad de datos.
b) Dé la carta ASM de las unidades de datos y control.

SOLUCIÓN

a) Descripción RT de la unidad de datos:

RT[8]:

LD	$RT \leftarrow$	$OUT_{7..0}$=
0	RT	[RT]
1	IN	[RT]

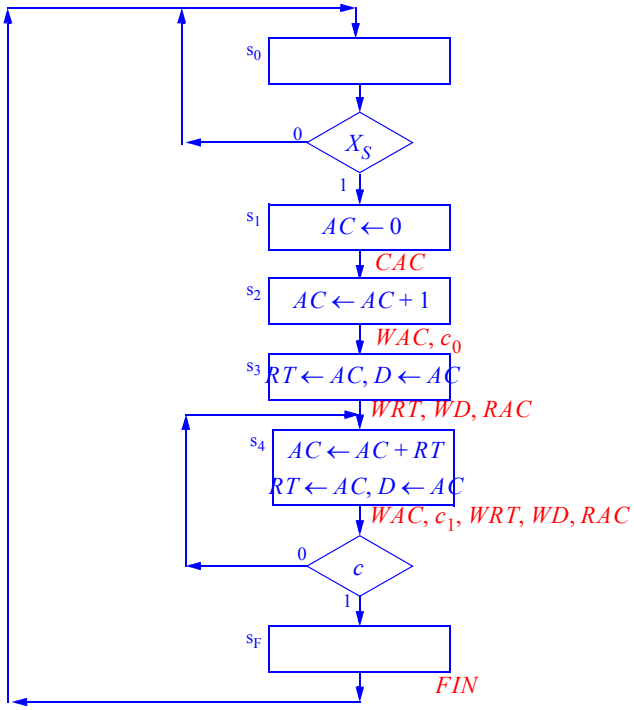
AC[8]:

CL LD RD	$AC \leftarrow$	$OUT_{7..0}$=	$BUS_{7..0}$=
000	AC	[AC]	HiZ
001	AC	[AC]	[AC]
010	IN	[AC]	HiZ
011	IN	[AC]	[AC]
100	0	[AC]	HiZ
101	0	[AC]	[AC]
110	0	[AC]	HiZ
111	0	[AC]	[AC]

ALU:

c_1c_0	$z_{7..0}$=	c=
00	x	0
01	x+1	1 si x=$FF, 0 otro caso
10	x+y	1 si x+y>$FF, 0 en otro caso
11	x-y	1 si x<y, 0 en otro caso

b) Carta ASM de las unidades de datos y control:



El algoritmo comienza generando la constante 1 en AC en dos pasos (primero lo pone a 0 y después le suma 1 vía la ALU). A continuación, se manda dicha constante a RT y display. Finalmente se entra en un bucle en el que, en un ciclo de reloj, se suman AC y RT que tienen las dos últimas constantes (para generar la siguiente), y se transfiere AC a RT y display para desplazar las mismas. La condición de salida del bucle es que se active el bit de carry. Esto indica que el número generado ya no cabe en 8 bits. La secuencia generada es 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, 233. Al realizar la última suma el resultado es 121 con carry activado, es decir, 377.

3.- **[3 puntos]** Describa con Verilog la Unidad de Datos del problema anterior.

SOLUCIÓN

```
// ALU
module ALUt (input [1:0] c, input [7:0] x, y, output cy, [7:0] z);
  reg [8:0] res;

  always @(*)
    case c:
      'b00: res = x;
      'b01: res = x+1;
      'b10: res = x+y;
      'b11: res = x-y;
    endcase
  assign {cy,z} = res;
endmodule

// Registro tipo RT
module RTt (input clk, ld, [7:0] in, output [7:0] out);
  reg [7:0] q;

  always @(posedge clk)
    if (ld)
      q <= in;

  assign out = q;
endmodule
```

```
// Registro tipo AC
module ACt (input clk, cl, ld, rd, [7:0] in, output [7:0] out, bus);
  reg [7:0] q;

  always @(posedge clk)
    if (cl)
      q <= 0;
    else if (ld)
      q <= in;

  assign out = q;
  assign bus = rd ? q : 'bZ;
endmodule

// Registro tipo Display
module DSPt (input clk, ld, [7:0] in);
  reg [7:0] q;

  always @(posedge clk)
    if (ld)
      q <= in;
endmodule

//Unidad de datos
module UDatos (input clk, wrt, wd, cac, rac, rac, [1:0]c, output cy);
  wire [7:0] bus, rtout, aluout, acout;

  ALUt ALU (c, acout, rtout, cy, aluout);
  RTt RT (clk, wrt, bus, rtout);
  ACt AC (clk, cac, lac, rac, aluout, acout, bus);
  DSPt Display (clk, wd, bus);
endmodule
```