



CENTRO UNIVERSITARIO DE CIENCIAS EXACTAS E INGENIERÍAS

INGENIERIA EN COMUNICACIONES Y ELECTRONICA

Circuitos Digitales 1

Tarea 08

Cronometro 59:59

Nombre: Aguilar Rodríguez Carlos Adolfo

Código: 215860049

Fecha: 19 de Abril del 2018

Profesor: Chávez Martínez Ehecatl Joel

Contenido

Marco Teórico	(3)
Máquina de 3 estados (Cronometro)	(4)
-Diagrama de flujo	(4)
-Código vhdl	(5)
-Código pat	(7)
-Código ioc	(7)
-Diagrama Esquemático	(9)
-Diagrama caja negra	(10)
-Simulación	(10)
-Plano real	(11)
-Tabla de resultados	(12)

- **Marco Teórico**

Diseño de un cronometro de 00:00 a 59:59 en forma de máquina de estados de 3 entradas y 4 salidas

Las entradas están compuestas por:

1- Reloj

2-Reset

3-Control

Todas ellas de un bit, su función es la siguiente:

Reloj, será encargado de generar pulsos de onda cuadrada

Reset, pone en 0 el contador y las memorias

Control, determina el inicio y la pausa de la cuenta.

Siempre al inicio de un sistema se debe poner los contadores y memorias a 0, por ello cuando se detecta un pulso de reloj y reset está en 1 todos los valores serán llevados a 0.

De igual forma si se tiene una cuenta registrada en el sistema y se detecta un reset junto con el pulso de reloj se dará un reinicio a los valores.

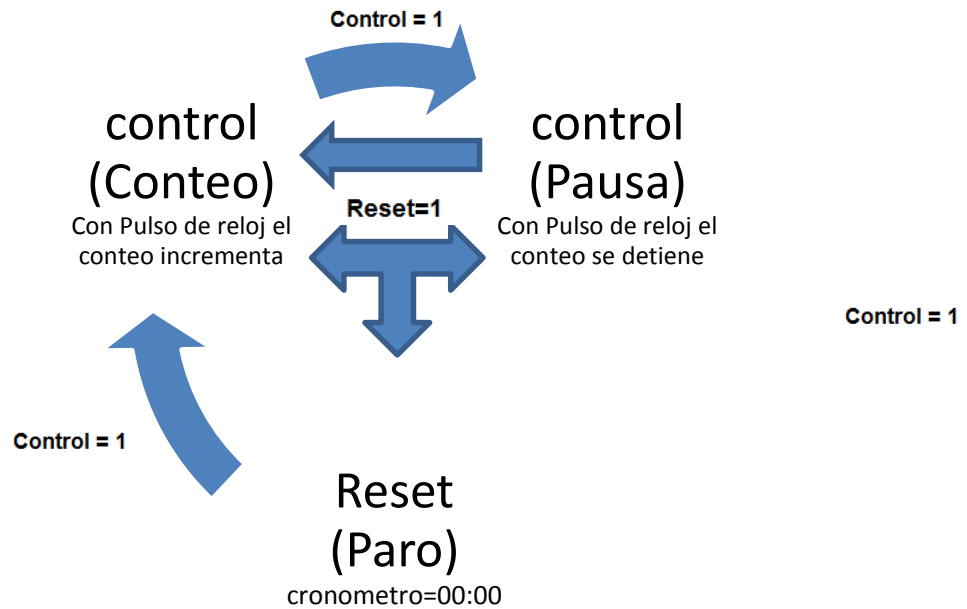
Para iniciar la cuenta es suficiente con detectar un pulso de reloj con el control en 1, se puede llevar a control a 0 y la cuenta seguirá avanzando de la misma manera para pausar.

Las salidas son asignadas para el cronometro;

Para las unidades de segundo y minuto se usa una salida de 4 bits, ya que el numero máximo que se representa con 3 Bits en binario es el 7 en decimal y las unidades máximas de segundo y minuto son 9 .

Para las decenas de segundo y minuto se usa una salida de 3 bits ya que el numero máximo que se representa con 2 Bits en binario es el 3 en decimal y las decenas máximas de segundo y minuto son 5 .

- Diagrama de flujo cronometro



Contador

Control	reset	reloj	Unidades seg	Decenas seg	Unidades min	Decenas min
0	1	No hay pulso	0	0	0	0
0	1	Hay pulso	0	0	0	0
0	0	No hay pulso	0	0	0	0
0	0	Hay 4 Pulsos	0	0	0	0
1	0	Hay 6 Pulsos	6	0	0	0
0	0	Hay 2 Pulsos	8	0	0	0
1	0	Hay pulso	8	0	0	0
0	0	Hay pulso	8	0	0	0
0	0	Hay pulso	8	0	0	0
1	0	Hay 10 pulsos	1	8	0	0
0	1	Hay pulso	0	0	0	0

Un reset aplicado en cualquier estado será llevado a el estado paro y reiniciara la cuenta del cronometro

- Código VHDL Cronometro 59:59

```
--Aguilar Rodriguez Carlos Adolfo
--Circuitos Digitales
--Cronometro 0 - 59:59
library ieee;
use ieee.std_logic_1164.all;
use ieee.std_logic_arith.all;
use ieee.std_logic_unsigned.all;
-- Entidad
entity cronometro is
  port(  rlj,rst,ctrl          :in std_logic;
        dseg :out std_logic_vector(2 downto 0);
        useg :out std_logic_vector(3 downto 0);
        dmin :out std_logic_vector(2 downto 0);
        umin :out std_logic_vector(3 downto 0));
  end cronometro;
--Arquitectura
architecture arql of cronometro is
  type estados is (paro,conteo,pausa);
  signal maquina : estados;
  signal memuseg : std_logic_vector(3 downto 0);
  signal memdseg : std_logic_vector(2 downto 0);
  signal memumin : std_logic_vector(3 downto 0);
  signal memdmin : std_logic_vector(2 downto 0);
begin
  unidad_segundos:process (rlj,rst,ctrl)
  begin

    if (rlj'event and rlj='1') then
      if (rst='1') then
        maquina<=paro;
        memuseg<= 0;

      else
        case maquina is
          when paro =>
            --memoria
            memuseg <= 0;
            --siguiente estado
            if(ctrl='1')then
              maquina<=conteo;
            end if;
          when conteo =>
            --memoria
            if (memuseg=9) then
              memuseg<=0;
            else
              memuseg <= memuseg+'1';
            end if;
            --siguiente estado
            if(ctrl='1')then
              maquina<=pausa;
            end if;
          when pausa =>
            --memoria
            memuseg <= memuseg;
            --siguiente estado
            if(ctrl='1')then
              maquina<=conteo;
            end if;

        end case;
      end if;
    end if;
  end process;

  decenas_segundos: process(rlj,rst,memuseg)
  begin
    if(rlj'event and rlj='1')then
      if (rst='1') then
        memdseg <= 0;
      else
        if (memuseg = 9) then
          if (memdseg = 5) then
            memdseg <= 0;
          else
            memdseg <= memdseg+'1';
          end if;
        end if;
      end if;
    end if;
  end process;
end architecture arql;
```

Continuación de Código VHDL Cronometro 59:59

```
        end if;
        end if;
        end if;
        end if;
    end process;

    unidades_minutos: process (rlj,rst,memdseg)
    begin
        if(rlj'event and rlj='1')then
            if (rst='1') then
                memumin <= 0;
            else
                if ((memdseg = 5) AND (memuseg = 9))then
                    if (memumin = 9) then
                        memumin <= 0;
                    else
                        memumin <= memumin+'1';
                    end if;
                end if;
            end if;
        end if;
    end process;

    decenas_minutos: process (rlj,rst,memumin)
    begin
        if(rlj'event and rlj='1')then
            if (rst='1') then
                memdmin <= 0;
            else
                if ((memumin = 9) AND (memdseg = 5) AND (memuseg = 9)) then
                    if (memdmin = 5) then
                        memdmin <= 0;
                    else
                        memdmin <= memdmin+'1';
                    end if;
                end if;
            end if;
        end if;
    end process;

    useg <= memuseg;
    dseg <= memdseg;
    umin <= memumin;
    dmin <= memdmin;
end arq1;
```

- Código ioc

```

TOP(
    (IOPIN dseg(0).0);
    (IOPIN dseg(1).0);
    (IOPIN dseg(2).0);
    (IOPIN useg(0).0);
    (IOPIN useg(1).0);
    (IOPIN useg(2).0);
    (IOPIN useg(3).0);
    (IOPIN rlj.0);
    (IOPIN ctrl.0);
)

LEFT (
)

RIGHT(
)

BOTTOM(
    (IOPIN dmin(0).0);
    (IOPIN dmin(1).0);
    (IOPIN dmin(2).0);
    (IOPIN umin(0).0);
    (IOPIN umin(1).0);
    (IOPIN umin(2).0);
    (IOPIN umin(3).0);
    (IOPIN rst.0);
)

```

- Código pat

```

-- declaracion de puertos
in rlj      B;
in rst      B;
in ctrl     B;
out useg(3 downto 0)X;
out dseg(2 downto 0)X;
out umin(3 downto 0)X;
out dmin(2 downto 0)X;

-- fin de puertos |
-- fuente de alimentacion
in vdd B;
in vss B;

begin

--
      rlj    rst    ctrl    useg    dseg    umin    dmin    vdd    vss

<0ns>:      0      1      0      ?*      ?*      ?*      ?*      1      0;
<+50ns>:    1      1      0      ?*      ?*      ?*      ?*      1      0;
<+50ns>:    0      0      0      ?*      ?*      ?*      ?*      1      0;
-- inicio de la cuenta
<+50ns>:    1      0      1      ?*      ?*      ?*      ?*      1      0;
<+50ns>:    0      0      1      ?*      ?*      ?*      ?*      1      0;
<+50ns>:    1      0      0      ?*      ?*      ?*      ?*      1      0;
<+50ns>:    0      0      0      ?*      ?*      ?*      ?*      1      0;
<+50ns>:    1      0      0      ?*      ?*      ?*      ?*      1      0;
<+50ns>:    0      0      0      ?*      ?*      ?*      ?*      1      0;
<+50ns>:    1      0      0      ?*      ?*      ?*      ?*      1      0;
<+50ns>:    0      0      0      ?*      ?*      ?*      ?*      1      0;
-- pausa
<+50ns>:    1      0      1      ?*      ?*      ?*      ?*      1      0;
<+50ns>:    0      0      1      ?*      ?*      ?*      ?*      1      0;
<+50ns>:    0      0      0      ?*      ?*      ?*      ?*      1      0;
<+50ns>:    1      0      0      ?*      ?*      ?*      ?*      1      0;
<+50ns>:    0      0      0      ?*      ?*      ?*      ?*      1      0;
<+50ns>:    1      0      0      ?*      ?*      ?*      ?*      1      0;
<+50ns>:    0      0      0      ?*      ?*      ?*      ?*      1      0;
--continucion de la cuenta
<+50ns>:    1      0      1      ?*      ?*      ?*      ?*      1      0;
<+50ns>:    0      0      1      ?*      ?*      ?*      ?*      1      0;
<+50ns>:    1      0      0      ?*      ?*      ?*      ?*      1      0;
<+50ns>:    0      0      0      ?*      ?*      ?*      ?*      1      0;
<+50ns>:    1      0      0      ?*      ?*      ?*      ?*      1      0;
<+50ns>:    0      0      0      ?*      ?*      ?*      ?*      1      0;
<+50ns>:    1      0      0      ?*      ?*      ?*      ?*      1      0;
<+50ns>:    0      0      0      ?*      ?*      ?*      ?*      1      0;

```

Código pat

```
--Aplicacion de reset
<+50ns>:      1      1      0      ?*      ?*      ?*      ?*      1      0;
<+50ns>:      0      1      0      ?*      ?*      ?*      ?*      1      0;

-- inicio de la cuenta
<+50ns>:      1      0      1      ?*      ?*      ?*      ?*      1      0;
<+50ns>:      0      0      1      ?*      ?*      ?*      ?*      1      0;
<+50ns>:      1      0      0      ?*      ?*      ?*      ?*      1      0;
<+50ns>:      0      0      0      ?*      ?*      ?*      ?*      1      0;
<+50ns>:      1      0      0      ?*      ?*      ?*      ?*      1      0;
<+50ns>:      0      0      0      ?*      ?*      ?*      ?*      1      0;
<+50ns>:      1      0      0      ?*      ?*      ?*      ?*      1      0;
<+50ns>:      0      0      0      ?*      ?*      ?*      ?*      1      0;
<+50ns>:      1      0      0      ?*      ?*      ?*      ?*      1      0;
<+50ns>:      0      0      0      ?*      ?*      ?*      ?*      1      0;
<+50ns>:      1      0      0      ?*      ?*      ?*      ?*      1      0;
<+50ns>:      0      0      0      ?*      ?*      ?*      ?*      1      0;
<+50ns>:      1      0      0      ?*      ?*      ?*      ?*      1      0;
<+50ns>:      0      0      0      ?*      ?*      ?*      ?*      1      0;
<+50ns>:      1      0      0      ?*      ?*      ?*      ?*      1      0;
<+50ns>:      0      0      0      ?*      ?*      ?*      ?*      1      0;
<+50ns>:      1      0      0      ?*      ?*      ?*      ?*      1      0;
<+50ns>:      0      0      0      ?*      ?*      ?*      ?*      1      0;
<+50ns>:      1      0      0      ?*      ?*      ?*      ?*      1      0;
<+50ns>:      0      0      0      ?*      ?*      ?*      ?*      1      0;
<+50ns>:      1      0      0      ?*      ?*      ?*      ?*      1      0;
<+50ns>:      0      0      0      ?*      ?*      ?*      ?*      1      0;
<+50ns>:      1      0      0      ?*      ?*      ?*      ?*      1      0;
<+50ns>:      0      0      0      ?*      ?*      ?*      ?*      1      0;
<+50ns>:      1      0      0      ?*      ?*      ?*      ?*      1      0;
<+50ns>:      0      0      0      ?*      ?*      ?*      ?*      1      0;
<+50ns>:      1      0      0      ?*      ?*      ?*      ?*      1      0;
<+50ns>:      0      0      0      ?*      ?*      ?*      ?*      1      0;
<+50ns>:      1      0      0      ?*      ?*      ?*      ?*      1      0;
<+50ns>:      0      0      0      ?*      ?*      ?*      ?*      1      0;

end;
```


DIAGRAMA ESQUEMATICO

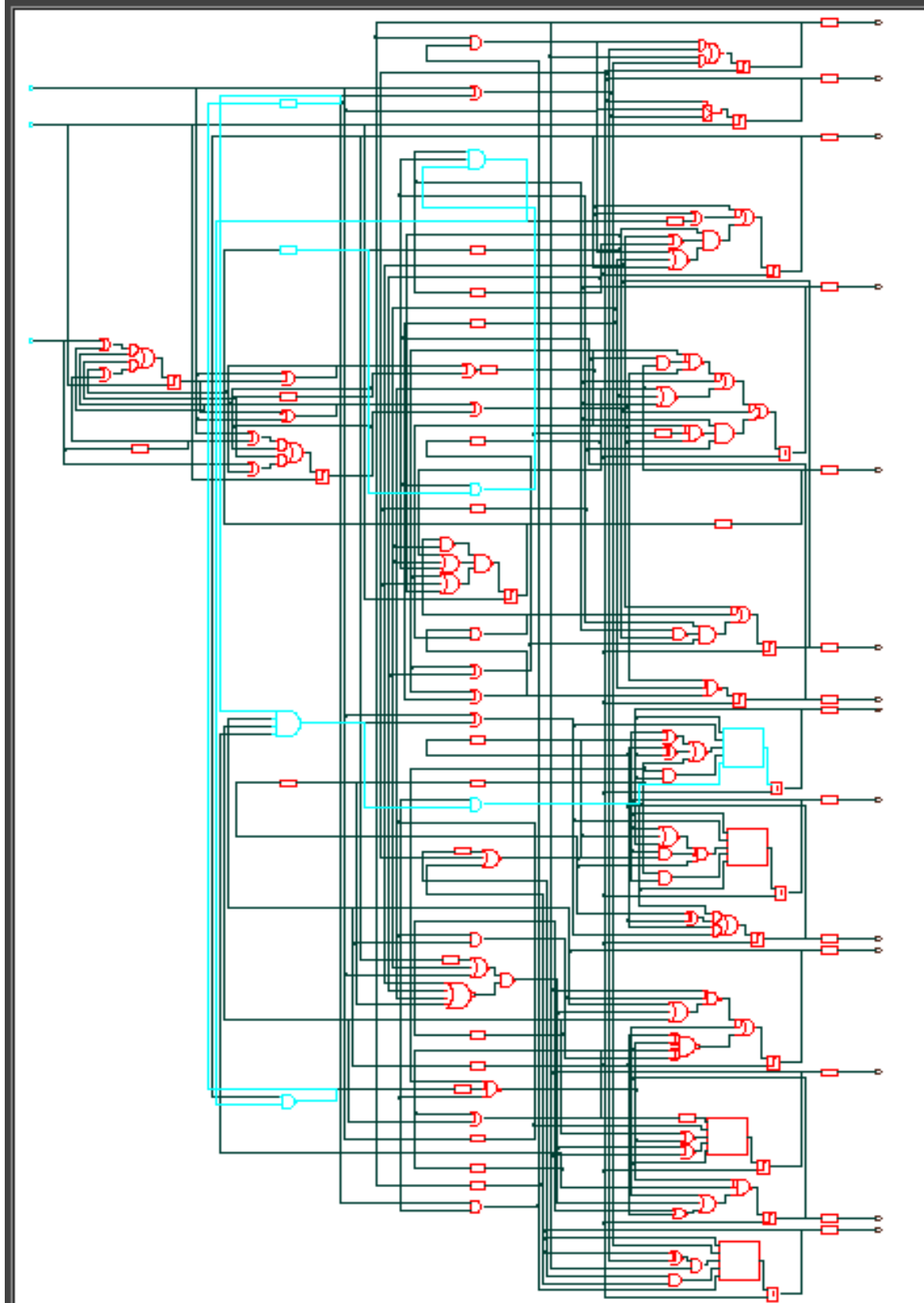
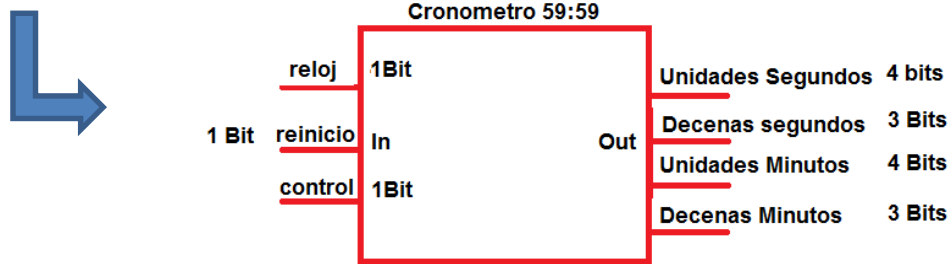
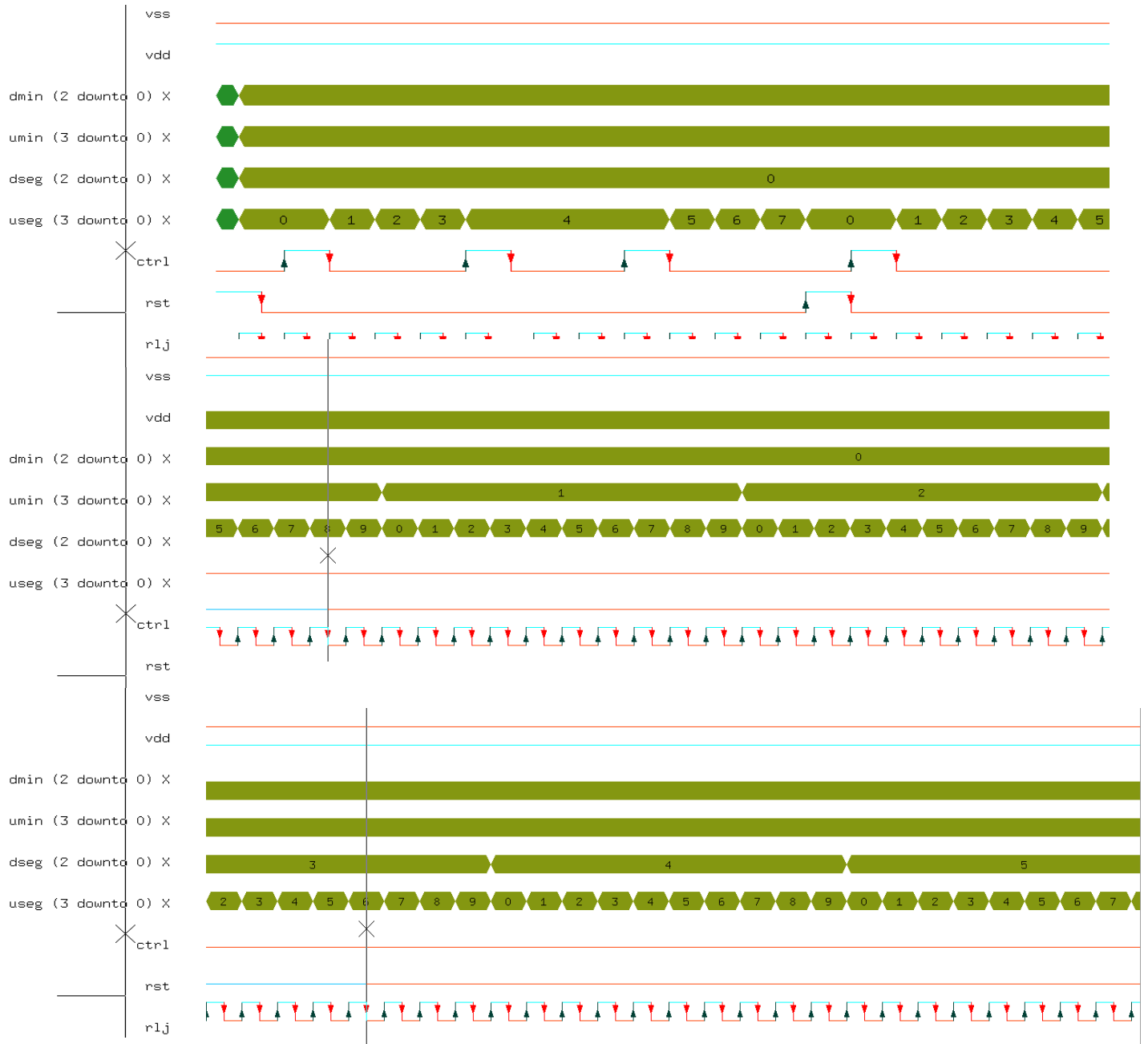


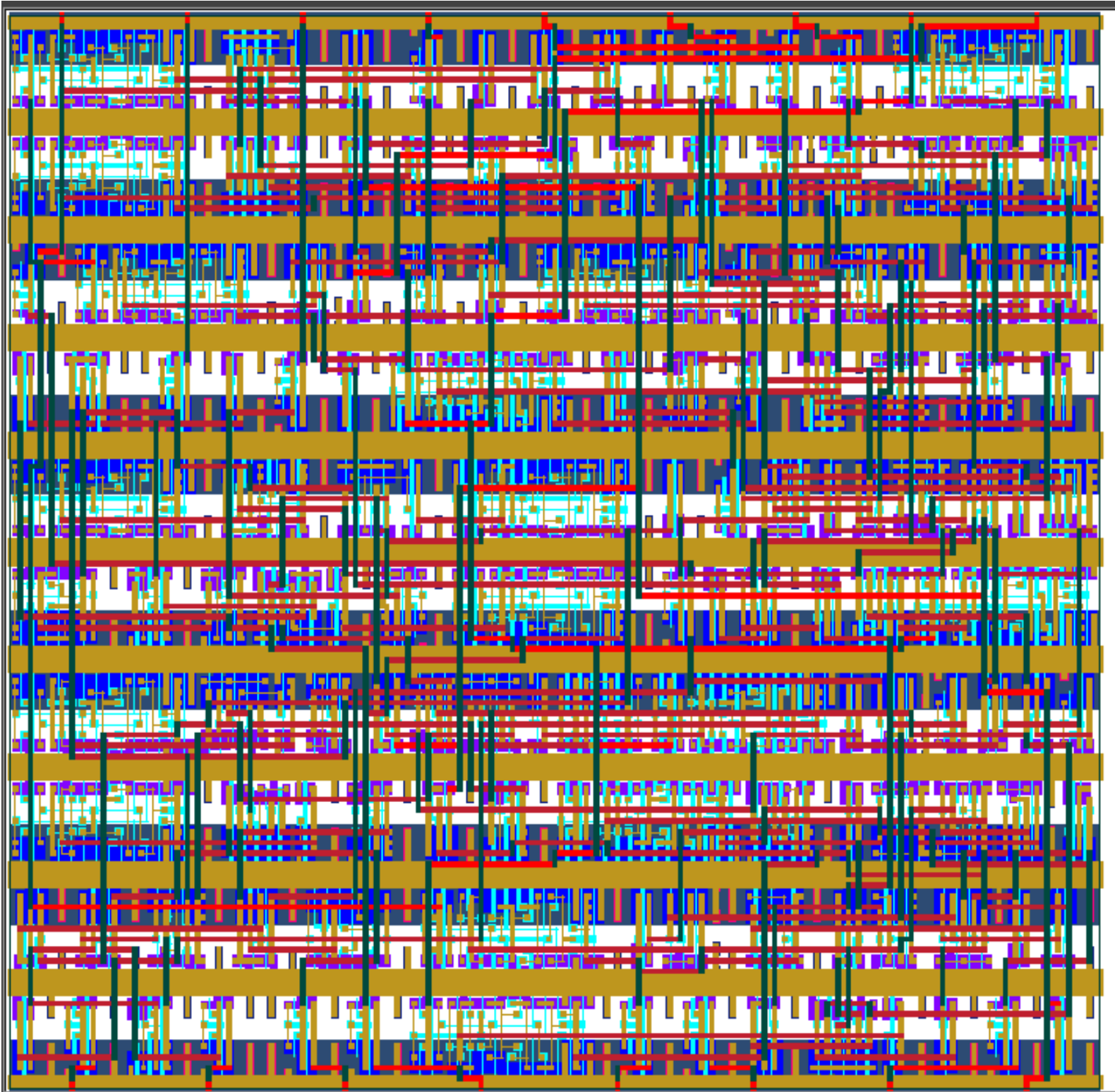
Diagrama caja negra



• Simulación



- Plano real



- **Tabla de resultados**

```

--L00N-----
Area on file 'cronometro_boog.vst'...210750 lamda² (with over-cell routing)
Area on file 'cronometro.vst'...217000 lamda² (with over-cell routing)
Critical path (no warranty)...3244 ps from 'memuseg 2' to 'memdmin_2_ins'
-----

--COUGAR-----
---> Figure size   : (   -100,   -100 )
                      ( 52100,  50100 )
---> Build transistors
<--- 1058
-----

```

Critical path 3244 pico segundos
 Area $A = 65100 \text{ micrometros}^2$
 Transistores = 1058
 Figura = 2,610,210,000