



INSTITUTO POLITECNICO NACIONAL
SECRETARIA ACADEMICA
DIRECCIÓN DE EDUCACIÓN SUPERIOR
ESCUELA SUPERIOR DE CÓMPUTO



Materia: Diseño de Sistemas Digitales	Turno: Vespertino
Semestre : segundo	Periodo: segundo parcial
Departamento o área: Ingeniería en Sistemas computacionales	Ciclo Escolar. 2020-2021 1
Fecha de Examen: 08/JUNIO/2021	Contenido a evaluar: Unidad III, IV
Horario del examen: 18:30 a 20:00, 20:00 a.21:30hrs.	Duración del examen 1:30 hrs.
EXAMEN TIPO : Único	Calificación.
Alumno <u>Silva Hernandez Noe Jasiel</u>	Boleta: <u>2016100938</u> Firma: _____
Profesor de esta asignatura en este grupo <u>Luis Octavio López Leyva</u>	Grupo <u>2CV15</u>

I) Relaciona las columnas: En cada paréntesis coloca la letra correspondiente a los conceptos que relaciona
(Valor 1 punto por cada paréntesis)

- | | |
|---|----------------------------|
| 1) Sintetiza un circuito automáticamente a partir de un diagrama de estado. | (10) Diagrama de estados |
| 2) Es simplemente un conjunto de palabras | |
| 3) Son abstracciones matemáticas que capturan solamente el aspecto referente a las secuencia de eventos que ocurren. | (8) Estado |
| 4) Es una situación en la que se permanece un cierto lapso de tiempo | (5) Estado final |
| 5) La secuencia de eventos que llevó hasta ahí puede considerarse como aceptables | (3) máquina |
| 6) Secuencia de eventos que representan concatenación de caracteres, se le llama? | (9) Función de transición |
| 7) Su funcionamiento consiste en ir pasando de un estado a otro, a medida que va recibiendo los caracteres de la palabra de entrada. | (6) Palabra |
| 8) Indica a qué estado se va a pasar sabiendo cuál es el estado actual y el símbolo que se está leyendo. | (4) Determinismo |
| 9) La característica que permite saber siempre cuál será el estado siguiente se llama? | (2) Lenguaje |
| 10) Está compuesto de círculos que representan estados y arcos para representar transiciones de un estado a otro e indica la salida que la máquina debe de generar. | (7) Autómata finito |
| | (1) VHDL |

II) Conteste las siguientes preguntas (valor 3 puntos cada item)

1.- ¿Cual es la diferencia entre un diagrama de estados y una tabla de asignación de estados?

Un diagrama de estados muestra los estados por los que pasa una máquina de estados finitos mientras que una tabla de asignación de estados muestra qué estado se moverá un autómata finito dado, basándose en el estado actual y otras entradas

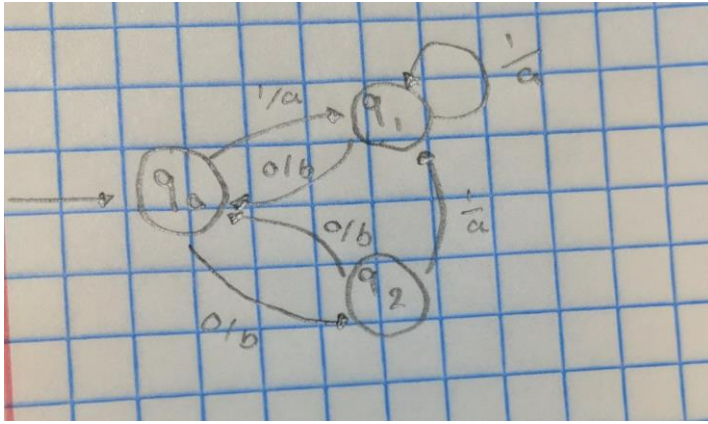
2.-¿Defina formalmente un Autómata de estados finitos

Es un modelo computacional que realiza cálculos en forma automática sobre una entrada para producir una salida.

3.-¿Ayudado de un diagrama defina qué es una máquina de Mealy?

R= * El estado siguiente depende de la entrada y del estado actual.

La salida depende de la entrada y del estado actual. • Esto implica que un autómata de Mealy, estando en un determinado estado, puede evolucionar hacia estados siguientes distintos y producir salidas distintas si se introduce una ó varias entradas binarias distintas.



4.- ¿A través de un diagrama defina qué es una máquina de Moore?

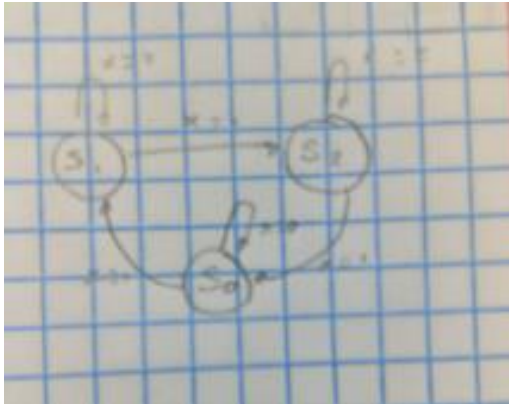
R= El estado siguiente depende de la entrada y del estado actual.

La salida depende exclusivamente del estado actual.

Esto implica que un autómata de Moore, estando en un determinado estado, produce siempre la misma salida, independientemente de cuál sea la entrada ó entradas de datos en ese estado.

Toda máquina de Moore es un caso particular de una máquina de Mealy.

Una máquina de Moore suele emplear mas estados internos y, consecuentemente mas Flip-Flops que uno de Mealy. Por el contrario, en una de Mealy el circuito combinacional que da las salidas del sistema suele ser mas complejo.



5.-¿Qué haría para evitar conflicto en el bus, al introducir datos a los registros R1, R2, y R3?

6.- ¿Por qué se considera un autómata al flip flop?

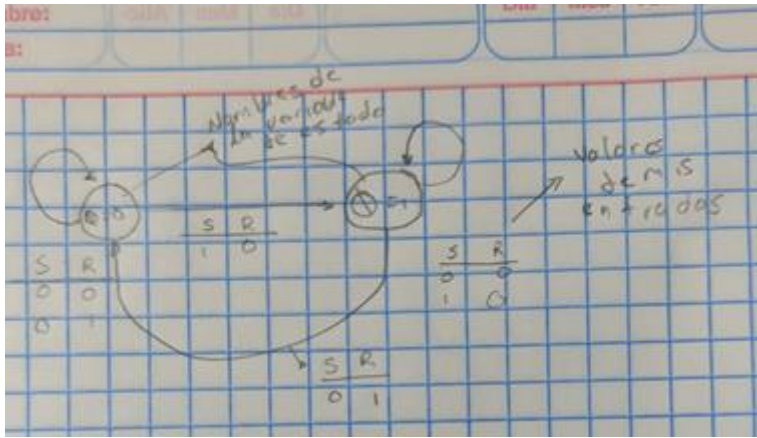
En un sistema secuencial además del valor de las entradas en un instante dado hay que tener en cuenta también el estado anterior por el que ha pasado el sistema, por lo que ha estos circuitos se les suele llamar sistemas con memoria. Para implementar esa memoria se suele utilizar biestables (Flip-Flop) o memorias ROM.

7.- ¿Describa detalladamente cómo trabaja la ALU 74LS 181?

La ALU puede programarse para que opere tanto en activa en alta como activa en baja.

puede realizar hasta 32 funciones diferentes (16 lógicas y 16 aritméticas), trabaja con números binarios de 4 bits, aunque se pueden conectar en cascada para aumentar el número de bits este opera por medio de una tabla por la cual ya vienen definidas sus operaciones

8.- ¿Haga el diagrama de estados de un flip flop SR?



9.- ¿Defina un circuito secuencial?

R= en el que la salida depende no solo de los valores actuales de las variables de entrada sino también de los valores pasados, o almacenados, de esas variables.

10.- ¿Realice el diagrama de una memoria RAM de 3X1.



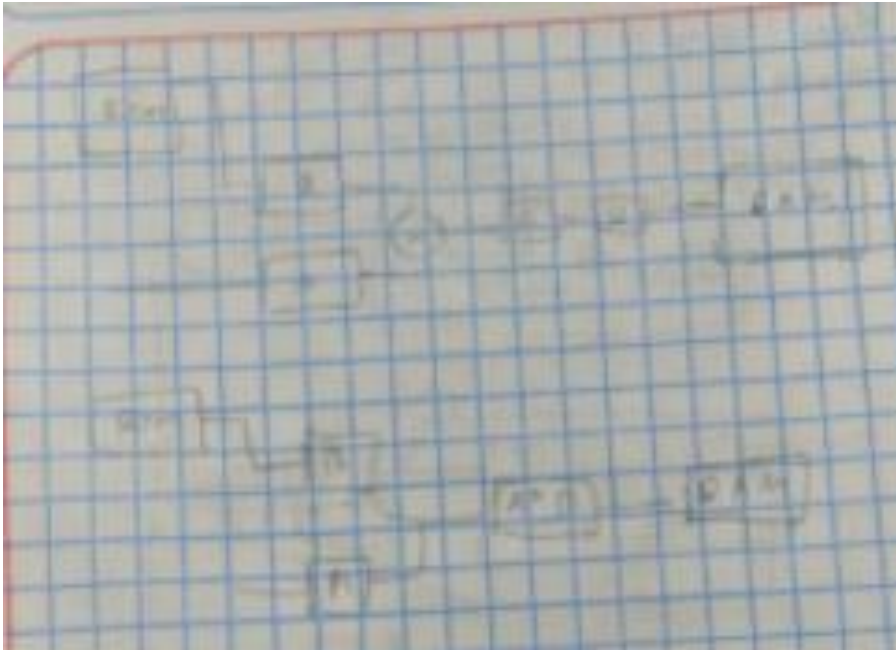
III) Realice los siguientes ejercicios.

(Valor 20 punto por cada ejercicio)

- 1.- Realice a diagrama de bloques el circuito que realice la multiplicación de $A \times B$ de 8 bits, y el resultado lo guarde en una memoria RAM, Cabe aclarar que los dato A debe tomarse del exterior y el dato B de una memoria ROM.
- 2.- Dado el siguiente diagrama, realizar el código VHDL correspondiente.
- 3.- Dado el diagrama de estado del circuito secuencial simple mostrado, este detecta si dos o más 1 consecutivos ocurren en su entrada W. Realice por el método de autómatas finitos deterministas (AFD) la implementación del circuito secuencial resultante.

III

1.-



```

2.- library IEEE;
use IEEE.STD_LOGIC_1164.ALL;

entity flipd_and_mux is
Port (in_put : in STD_LOGIC_VECTOR (3 downto 0);
      Sel : in STD_LOGIC_VECTOR (1 downto 0);
      clk : in STD_LOGIC;
      clearN : in STD_LOGIC;
      Outp : out STD_LOGIC);

end flipd_and_mux;

architecture ciruito of flipd_and_mux
    component Mux4_to_1
        Port ( w : in STD_LOGIC_VECTOR (3 downto 0)
              m : in STD_LOGIC_VECTOR (1 downto 0);
              f : out STD_LOGIC);
    end component;

    component flipd_flip_flop
        Port ( D : in STD_LOGIC;
              Clock : in STD_LOGIC;
              Reset_N : in STD_LOGIC;
              Q : out STD_LOGIC);
    end component;

    signal mux_d : STD_LOGIC;

begin
    mux: Mux4_to_1 PORT MAP ( w=>in_put, m=> Sel, f=> mux_d);
    flip_flop: d_flip_flop
        PORT MAP ( D=>mux_d, Q=>Outp, Clock=>clk, Reset_N => clearN);
End ciruito;

```