

Diseño definitivo



Figura 1: Simbología del diagrama de bloques

Memorias de mayor capacidad: Representan la memoria RAM y ROM en el circuito.

Registros: Son los encargados de guardar la información en una posición en memoria.

Cajas Negras: Representan un sector del circuito del que aún no se ha decidido los componentes que lo forman, pero se tiene la idea general de la función de ese sector.

Funciones de Logisim: Simbolizan todos los componentes que se encuentran en el programa Logisim, es decir, compuertas AND, OR, XOR, buffers, etc.

En general el programa será controlado por un sistema compuesto de un solo reloj, varios contadores y decodificadores. Esto se debe a que el programa no puede trabajar al mismo tiempo todas las etapas, debe completar algunas antes de seguir con la siguiente. Los componentes anteriores permiten que se logren desprender partes del circuito mientras se ejecutan otras, por ejemplo desprender el sistema de escritura en la memoria RAM mientras se hace la búsqueda de la siguiente instrucción.

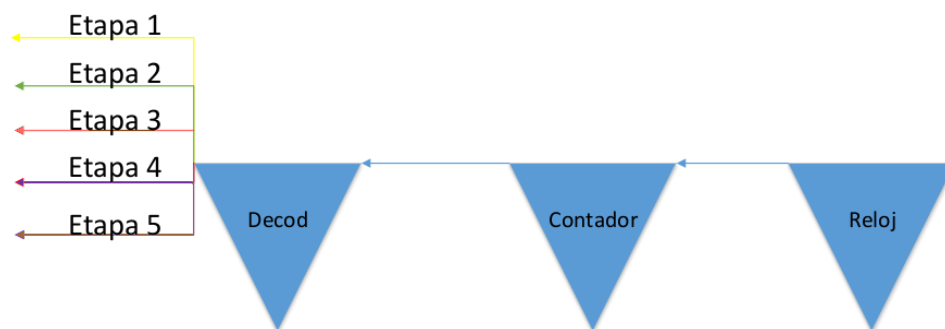


Figura 2: Controlador de etapas

El diseño constará de diferentes etapas y no necesariamente las establecidas por la teoría de la máquina de Turing. Para mayor facilidad en la representación y lectura del esquema del circuito se separa esta sección del circuito cuya función básica es habilitar o deshabilitar el paso de cada una de las etapas del circuito general(ver figura 3).

Reloj: envía pulsos a través del circuito, lo que provoca que las entradas de los diversos componentes se actualicen.

Contador: este componente del circuito aumenta en uno con cada pulso del reloj. El número guardado en el contador permite acceder a una dirección de la memoria ROM o a una de las salidas del decodificador (ver Figura 3).

Decodificador: Es el componente encargado de hacer la transición de etapas. Por medio de buffers de control habilita o deshabilita el paso del reloj a secciones del circuito.

Colores: En las figuras 2 y 3 se utiliza cableado con distintos colores para identificar cuáles componentes están implicados en cada una de las etapas.

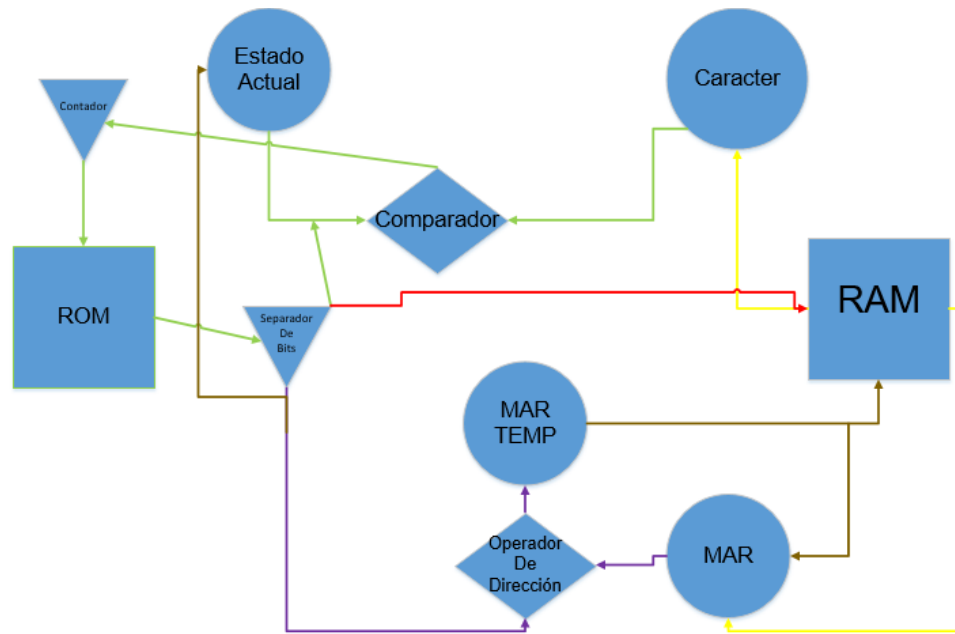


Figura 3: Vista General

Primera etapa: Se carga en primera instancia el carácter inicial a partir de la dirección de memoria, ingresada al inicio como parámetro, desde la memoria RAM para poder buscar posteriormente la instrucción que se debe ejecutar.

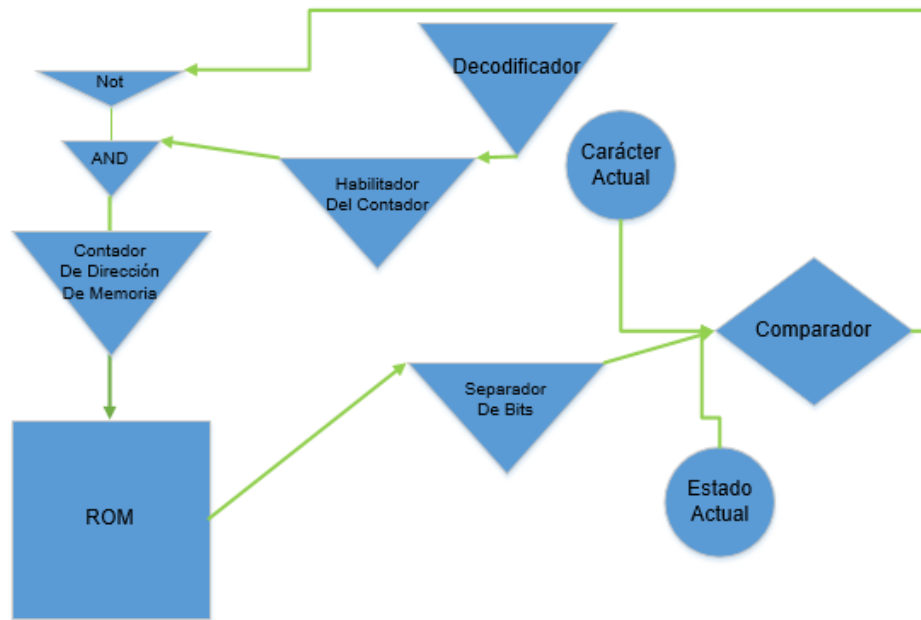


Figura 4: Etapa 2

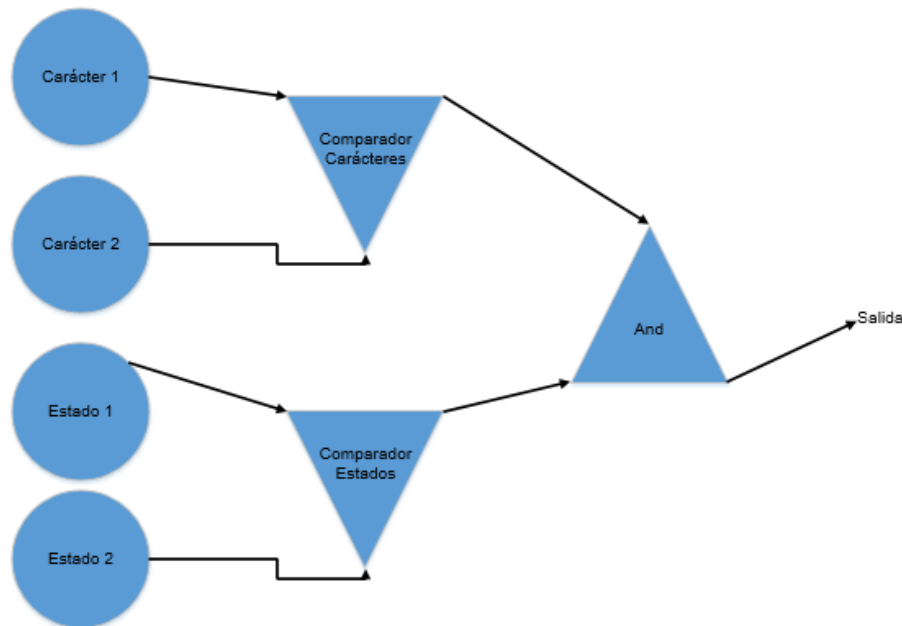


Figura 5: Comparador

Segunda etapa: consiste en comparar el estado inicial y carácter actual de la posición de memoria con cada uno de las instrucciones que se encuentra en la memoria ROM. Este proceso se repite hasta encontrar una instrucción que coincida.

La memoria ROM tiene una entrada que es la posición de memoria a leer. Para recorrer todas las posiciones de la memoria ROM es necesario un contador que actualice en cada pulso del reloj la posición a acceder. Para que este contador actualice, debe recibir una señal *true* del Habilitador Del Contador (*buffer* de control) y a su vez, el Habilitador Del Contador debe recibir una señal *true* de la salida *Etapa 2* del decodificador. La memoria ROM envía los datos del nuevo estado y nuevo carácter al separador de bits.

El comparador, que se puede observar en la figura 5, se encarga de comparar el carácter y estado actual con los provenientes de la ROM, los cuales serían el carácter y estado futuro. Si estos son iguales, el operador tendrá como salida una señal *true* y esta bloqueará las señales del reloj que le ingresan a la RAM, por otro lado, si esta salida es una señal *false*, estas no se bloquearán y se ingresará de nuevo al ciclo buscando los que sean iguales al estado y carácter actual. En caso de que se recorra toda la memoria ROM, el **carry** del contador deshabilita el reloj que controla el circuito.

Una vez comprobado que la información contenida en la instrucción se corresponde con el carácter y estado actuales, la etapa finaliza y el decodificador pasa a *Etapa 3*. En este proceso se usaran la memoria ROM, dos registros de los que se leerá el estado y carácter actual, el comparador, un NOT, el separador de bits que proporciona el carácter y el estado leídos desde la ROM y una compuerta AND que compara la información proporcionada por la ROM y el registro.



Figura 6: Etapa 3

Tercera etapa: Se debe sobre-escribir sobre el espacio de memoria en el que esta posicionado el cursor de la memoria RAM el carácter siguiente procedente de la memoria ROM. Para esto se utiliza un separador de bits que toma solamente la información del carácter a escribir. Esta información será escrita en la memoria RAM si la salida *Etapa 3* del decodificador envía una señal *true* al Habilitador de Paso (*buffer* de control).

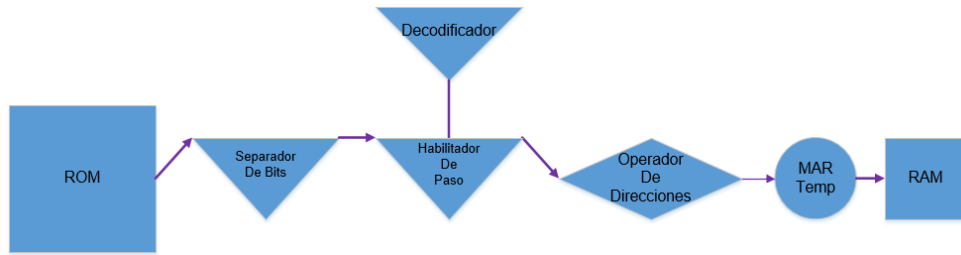


Figura 7: Etapa 4

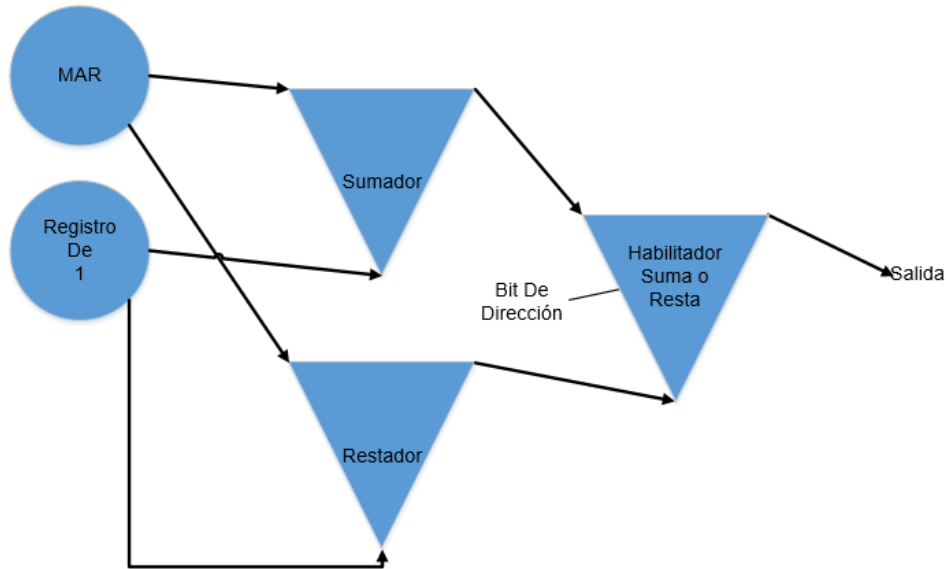


Figura 8: Operador de dirección

Cuarta etapa: Se mueve el puntero de la RAM según lo que indique la instrucción de la memoria ROM. Por medio del Operador de Dirección, figura 8, y la dirección guardada en el registro MAR se suma o resta una posición.

El operador de dirección recibe un bit de dirección, la dirección de memoria que se está leyendo en la cinta (desde el registro MAR), y cuenta con un registro que contiene un uno constante. Ambos registros son enviados al sumador y restador (las dos operaciones son ejecutadas). El habilitador de suma y resta cuenta con dos *buffers* de control (uno de ellos negado) que habilitan el paso de la información proveniente del sumador o del restador.

Esta nueva posición se guarda en el registro MAR Temp, ya que si se guardara en el registro MAR directamente esto crearía un error al alimentar la entrada con la salida en un mismo ciclo de reloj, y este se lo enviará a la RAM para que esta reconozca hacia donde mover el puntero.

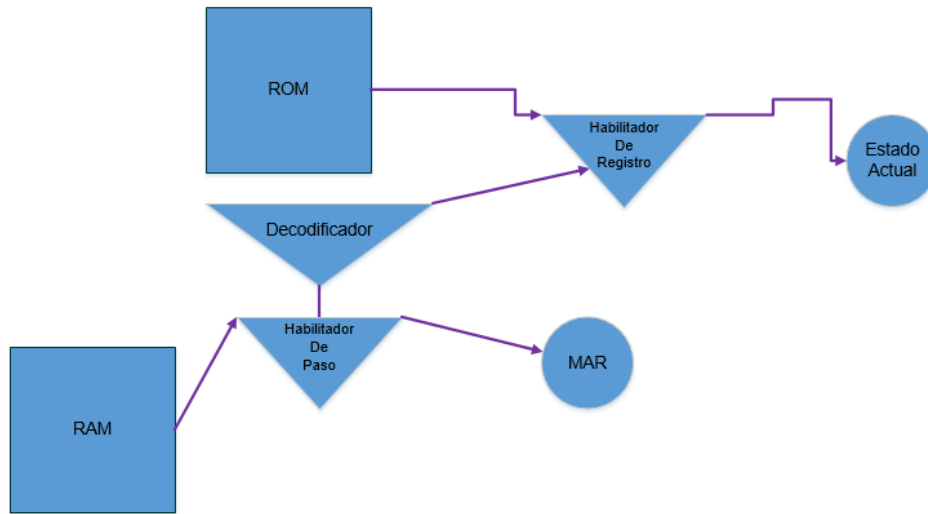


Figura 9: Etapa 5

Quinta etapa: Se envía una señal desde la memoria RAM hacia el registro Estado Actual la cual le indicará a este registro si habilitarse o no con las señales del reloj, para dejar pasar el estado futuro proveniente desde la ROM. Al mismo tiempo, la memoria RAM le envía la dirección en la que se encuentren actualmente la memoria ROM al registro MAR para que se almacene. Por último, se reinicia el contador que redirecciona el puntero de la ROM.