

Universidad de Costa Rica
Escuela de Ciencias de la Computación e Informática
CI-0118 Proyecto Integrador de Arquitectura y Ensamblador
Prof. Juan José Vargas.
Segundo entregable: Diseño detallado
I ciclo de 2019

Integrantes del grupo

- B70771 Daniel Artavia Cordero.
- B75133 Gloriana Mora Villalta.
- B65580 Luis Carlos Quesada Rodríguez.

1. Composición del circuito

1.1. Dispositivos de acceso general

La máquina de Turing a implementar será construida con múltiples subcircuitos, sin embargo, estos se deben montar en un ambiente común. Muchos de los componentes de la máquina deben ser compartidos por dos o más de los subcircuitos, por tanto, se montan como dispositivos de acceso general y son los siguientes:

- *Instruction Buffer Register (IBR)*: 17 *bits*.
- Puntero a memoria de símbolos (*RAM* principal): 10 *bits*.
- Memoria de símbolos (*RAM* principal): Aproximadamente 1 KB.
- Memoria de programa ordenado (*RAM* secundaria): 68 KB.
- Memoria de programa (*ROM*): 131 KB.
- Bandera de encendido (Flip Flop activador): 1 *bit*.
- Registro de símbolo: 7 *bits*.
- Registro de estado: 8 *bits*.

Las relaciones de estos dispositivos con los demás módulos e incluso entre ellos mismos se muestra de forma detallada en la sección [2](#).

1.2. Módulo ordenador

El módulo ordenador tiene como función copiar las instrucciones de la memoria *ROM* a una memoria *RAM* de manera ordenada para que se pueda acceder a ellas más rápidamente. Para realizar esta función, se tomará la combinación del estado actual de la máquina y el símbolo leído como llave única para usarse como dirección de memoria. Tomando en cuenta que los bits de direccionamiento no pueden ser 00, entonces todas las filas que estén conformadas solamente por ceros no serán escritas.

El módulo consta de un contador de 15 *bits*, un comparador de 32 bits, un registro que contiene 32 ceros y un *splitter* para separar los datos leídos de la memoria *ROM* en dos partes.

El funcionamiento del módulo se puede describir de la siguiente manera:

- El usuario presiona un botón para iniciar la simulación que envía una señal de activación.
- La señal de activación llega a un contador de 15 bits que se incrementa en uno y envía el número almacenado como dirección a ser leída en la memoria *ROM*.
- Los datos que entran son comparados con 32 ceros. Si son iguales se envía una señal para desactivar la memoria *RAM* para que no sean escritos.
- Los datos son divididos en dos partes:
 - Los bits del 0 al 14 saldrán como bits de dirección a la memoria *RAM*
 - Los bits del 15 al 31 saldrán como bits de datos a ser escritos en la memoria *RAM*
- Cuando el contador alcanza su máximo, envía un bit avisando que ha terminado.

Lo anterior se puede dividir en tres secciones:

1.2.1. Sección del Contador

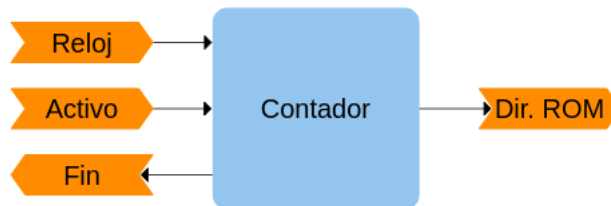


Figura 1: Sección del Contador

1.2.2. Sección del Comparador



Figura 2: Sección del Comparador

1.2.3. Sección del Splitter



Figura 3: Sección del Splitter

1.3. Módulo de direccionamiento

El módulo de direccionamiento se encarga de ejecutar la acción de mover el puntero de símbolos en memoria *RAM* hacia la izquierda, derecha o permanecer estático. Se decidió utilizar 2 *bits* de dirección, de forma que las combinaciones 01, 10 y 11 representen los movimientos mencionados respectivamente, y se reserva 00 para indicar que esa porción de memoria no fue escrita por el usuario, por lo que la máquina lo considera como estado de rechazo y finalización del programa.

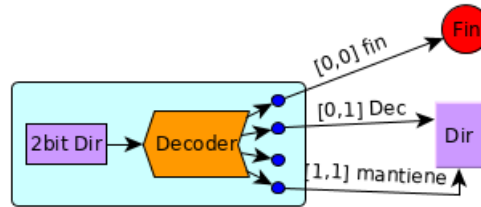


Figura 4: Diseño de la unidad de dirección

La entrada pasa a un decodificador que emitirá una señal a uno de los 3 pines de salida de 1 bit, funcionando como *triggers* para la acción de movimiento en el contador que funciona como puntero de símbolos.

Con los 2 bits en la posición de dirección, se pueden llevar a cabo las siguientes acciones.

- Entrada [00]: Se considera que la instrucción en el IBR es una porción de memoria que no fue escrita por el usuario, por lo que se considera un estado de rechazo y se procede a dar inicio a la secuencia de finalización de programa.
- Entrada [01]: Se considera que el puntero se mueve a la izquierda (se le resta una unidad).
- Entrada [10]: Se considera que el puntero se mueve a la derecha, no se realiza una acción pues el contador lo hará de forma automática.
- Entrada [11]: Se mantiene la misma posición del puntero, para esto se desactiva el contador por un pulso del reloj.

Este módulo tiene 3 pines de salida para el control del contador que funciona como puntero a la memoria de símbolos, pues la entrada [10] no es necesario que tenga una salida, ya que se ejecutará como el paso *default*, esto se debe a que al utilizar un contador en lugar de un registro, la única acción necesaria para hacer un incremento es una entrada de reloj.

1.4. Módulo de control

El módulo de control es el encargado de orquestar las etapas de ejecución del programa que, como se ilustra en la figura 5, son las siguientes: (0) leer un símbolo de la memoria *RAM*, (1) buscar una instrucción, (2) escribir en la memoria *RAM* y cambiar de estado, y (3) desplazarse en la memoria *RAM*.

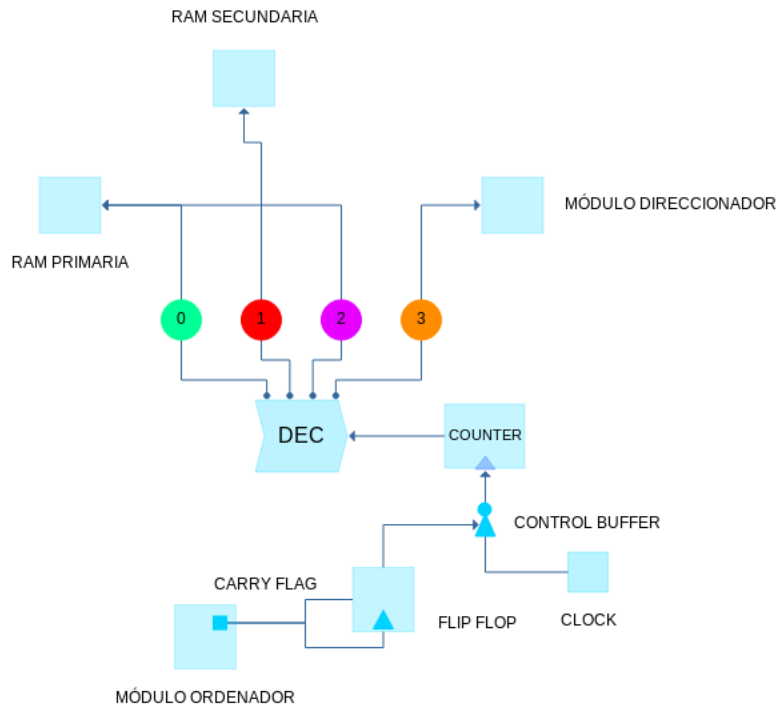


Figura 5: Diseño de la unidad de control

Esta unidad consta de un decodificador cuya entrada es un contador de 2 *bits* que incrementará cuando finalice cada una de las etapas de ejecución y se deba dar paso a la siguiente. Dicho contador se encuentra conectado al reloj, sin embargo, se regula el paso de flujos mediante un *control buffer* que se encuentra conectado a un *flip flop*. Este último será un indicador de inicio y fin de arraque, ya que la máquina requiere una etapa previa de ordenamiento de instrucciones que es ejecutada por el módulo ordenador. La finalización de esta etapa de ordenamiento implica el recorrido completo de la memoria *RAM* secundaria, por tanto, se activa un *carry flag* que el *flip flop* recibe como señal para permitir el paso de los flujos de reloj en todas las etapas siguientes. Esto activará de forma sucesiva cada uno de los pines de salida del decodificador y enviará señales de activación a los respectivos módulos y dispositivos de acceso general.

1.5. Dispositivos de entrada y salida

Para la entrada de datos se utiliza una memoria *RAM* en la que se cargarán los símbolos que en una máquina de Turing tradicional se almacenarían en la cinta, existe una segunda memoria de tipo *ROM* donde se debe cargar las instrucciones a ejecutar. En el módulo de entrada también habrán 3 botones, uno para mover el puntero de símbolos a la derecha, otro para moverlo a la izquierda y finalmente un botón para iniciar la máquina, de esta forma el usuario puede decidir en que posición de la memoria de símbolos desea iniciar la ejecución del programa. Para los dispositivos de salida se utiliza una matriz de 7 *leds* que representará el símbolo que haya en la celda de memoria *RAM* a la que se le apunte al finalizar la ejecución del programa, de la misma forma se activará un *led* que indica el fin de la ejecución.

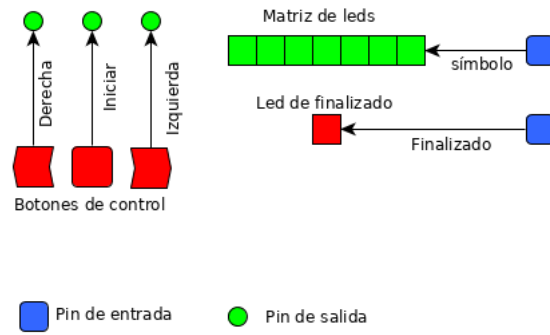


Figura 6: Diseño de dispositivos entrada y salida

2. Flujo de ejecución

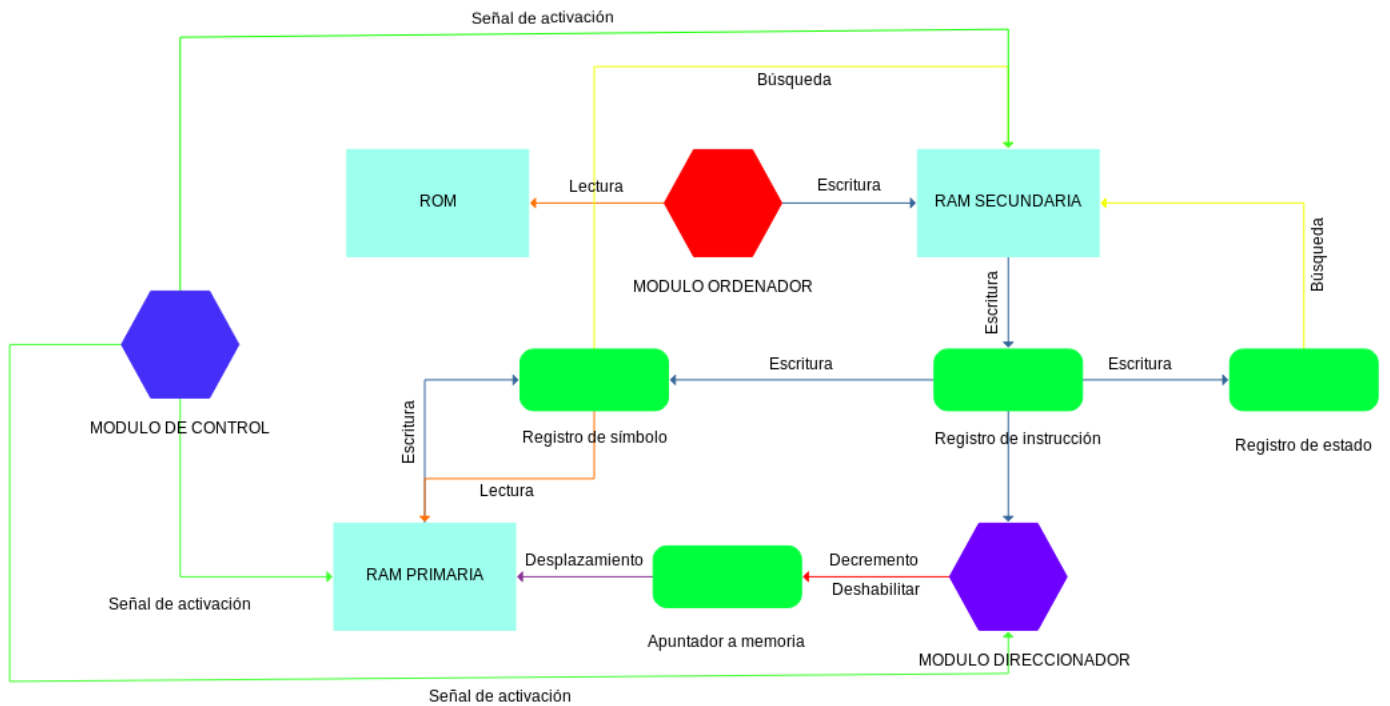


Figura 7: Relaciones entre los componentes