



Proyecto 1

Introducción

Este proyecto consiste en diseñar un computador que simula Máquinas de Turing por medio de circuitos digitales, microprogramación y lenguaje de máquina. Dicho computador será implementado por medio de la versión italiana del software de simulación de circuitos Logisim.

“Una máquina de Turing puede considerarse como una cinta infinita dividida en casillas, cada una de las cuales contiene un símbolo. Sobre dicha cinta actúa un dispositivo que puede adoptar diversos estados y que, en cada instante, lee un símbolo de la casilla sobre la que está situado. En función del símbolo que ha leído y del estado en que se encuentra, realiza las tres acciones siguientes: pasa a un nuevo estado, escribe un símbolo en lugar del que acaba de leer y se desplaza a una posición hacia la izquierda, o hacia la derecha, o bien la máquina se para. (...) El concepto de máquina de Turing es tan general y potente, que es posible construir una máquina que sea capaz de simular el comportamiento de otra máquina de Turing cualquiera. Esto es lo que se llama máquina de Turing universal. Gracias a su existencia, podemos disponer de ordenadores electrónicos, que no son más que máquinas generalizadas capaces de realizar cualquier cálculo computable.” [1]

Note que el comportamiento de una máquina de Turing es similar al de un autómata de estados finitos estudiado en cursos anteriores. El computador que simula máquinas de Turing debe tener una arquitectura similar a la usada hoy en día en computadores comerciales, es decir, utilizando la arquitectura propuesta por el físico y matemático John von Neumann en 1945 [2]. Los circuitos digitales del computador deben ser implementados en software, usando el simulador de circuitos digitales Logisim (<http://logisim.altervista.org/>), según se describe en este enunciado.

Objetivos de aprendizaje

Los objetivos de este primer proyecto integrador son:

- Utilizar conceptos básicos de diseño de computadores, microprogramación y lenguaje de máquina para construir un simulador de máquinas de turing por medio de circuitos digitales.
- Aplicar el modelo de computadores propuesto por John von Neumann en el diseño de un computador que simula máquinas de turing, para comprender la arquitectura del CPU, memoria y dispositivos propuestos en dicho modelo.
- Definir un lenguaje de máquina para el computador que simula máquinas de Turing, e implementarlo por medio de microprogramas y microinstrucciones construidas con circuitos digitales, para comprender la relación de los circuitos digitales con el lenguaje de máquina.
- Utilizar técnicas básicas de validación de circuitos digitales para evaluar la implementación del computador que simula máquinas de Turing.
- Evaluar ejemplos de máquinas de Turing, cuya definición se almacenará en una memoria ROM, y que serán ejecutados por medio de instrucciones en lenguaje de máquina en el computador con arquitectura de von Neumann.

Características del sistema digital a construir

Una máquina de Turing se define formalmente, según [3] (también hay definiciones parecidas de autores diferentes), como una 7-tupla $(Q, \Sigma, \Gamma, \delta, q_0, q_{accept}, q_{reject})$, donde:

- Q : es un conjunto de estados (los estados del autómata representado por la función δ).
- Σ : es el alfabeto de entrada pero no incluye el símbolo blanco.
- Γ : es el alfabeto de la cinta, contiene el símbolo blanco y el alfabeto Σ .
- δ : es la función de transición que representa los cambios de estado, $\delta: Q \times \Gamma \rightarrow Q \times \Gamma \times \{L, R\}$, donde L y R representan el movimiento de la cabeza de la cinta hacia la izquierda (L) o a la derecha (R).
- $q_0, q_{accept}, q_{reject}$: representan el estado inicial, el estado de aceptación y el estado de rechazo de la máquina, respectivamente, donde $q_{accept} \neq q_{reject}$.

El computador a construir debe considerar los elementos usualmente incluidos en un sistema digital con arquitectura de von Neumann, a saber, un CPU (conteniendo registros, unidad de control, unidad aritmético/lógica, etc.), así como una memoria RAM, dispositivos periféricos según se requiera, así como cualquier otro componente apropiado para la solución del problema.

La función de transición, incluyendo las entradas, salidas y cambios de estado del autómata de cada máquina de Turing, serán representadas por medio de una memoria ROM, usando un formato binario, como una función representando sus entradas y salidas en una tabla de verdad. Como anexo a este enunciado le serán provistos uno o varios ejemplos de máquinas de Turing y su correspondiente tabla de verdad, los cuales debe utilizar para probar el computador, cambiando el contenido de la memoria ROM al del ejemplo correspondiente. La cinta infinita será representada por medio de una memoria RAM de un tamaño máximo predefinido, el puntero a la posición actual en la cinta será un registro dentro del CPU.

Para ejecutar el programa de una máquina de Turing, primero debe leer una hilera por medio de algún dispositivo de entrada en su computador y colocarla al inicio de la cinta (memoria RAM). Seguidamente debe iniciar la ejecución del autómata, comenzando por el estado inicial q_0 y tomando la entrada del siguiente carácter de la hilera recién colocada en la cinta. El estado inicial se encuentra definido en la memoria ROM que contiene la función de transición. Para determinar el siguiente estado debe buscarlo también en la memoria ROM, luego debe ejecutar las acciones que considere necesarias para representar el cambio de estado del autómata y de los demás componentes de la máquina de Turing, como son el contenido de la cinta, el registro que apunta a la posición actual, las salidas con el resultado de la operación, etc.

Luego del primer cambio de estado, en cada ciclo se lee el carácter que está bajo la cabeza lectora en ese momento (que típicamente finaliza al encontrar un blanco), buscando el siguiente estado en la ROM con base en la entrada y el estado actual, ejecutando el cambio de estado del autómata y modificando el estado de los otros elementos de la máquina de Turing como son el valor de la posición actual en la cinta y la posición de la cabeza (hacia la izquierda o derecha), hasta finalizar el procesamiento de la entrada o alcanzar alguno de los estados q_{accept} o q_{reject} .

El lenguaje de máquina para el computador debe considerar todas las acciones necesarias para ejecutar el programa de una máquina de Turing, y debe implementar cada instrucción en lenguaje de máquina por medio de microprogramas implementados con microinstrucciones. Algunas de las acciones que posiblemente desee considerar en su diseño, al definir las instrucciones en lenguaje de máquina, son:

- Poner en ejecución el programa en lenguaje de máquina.

- Colocar un valor (hilera) en la cinta.
- Leer un valor de la cinta.
- Escribir un valor en la cinta.
- Cambiar el valor del puntero a la cinta.
- Inicializar el estado del autómata.
- Buscar la transición al estado siguiente en la memoria ROM.
- Ejecutar la transición al estado siguiente.
- Mostrar apropiadamente los resultados generados por el programa en lenguaje de máquina.
- Determinar la finalización en la ejecución del programa en lenguaje de máquina.

Note que en los puntos anteriores se menciona el programa en lenguaje de máquina que ejecuta la máquina de Turing, pero no se ha indicado un espacio o forma de obtener dicho programa. Como parte de la solución, usted debe construir el programa en lenguaje de máquina que será ejecutado en el computador a ser construido como parte de este proyecto. Además, debe definir un componente de memoria apropiado (sea RAM, ROM o del tipo que considere conveniente) para almacenarlo y que puede ser accesado desde el CPU para ejecutarlo. Las memorias descritas previamente ya tienen propósitos específicos, a saber, una memoria RAM para representar la cinta infinita y una memoria ROM para contener los cambios de estado del autómata. Entonces, queda como parte de la tarea definir la memoria para colocar el programa en lenguaje de máquina que ejecuta las acciones indicadas por el autómata y que en última instancia permite ejecutar la máquina de Turing.

Por otra parte, en el nivel de los circuitos digitales, las características específicas del computador, como son la cantidad, nombre, tamaño y función de los registros del CPU, así como las acciones específicas de la unidad de control o la unidad aritmético lógica son propios de su diseño. Recuerde consultar referencias bibliográficas formales, en el tema de diseño de circuitos digitales, para que las decisiones en su diseño sean las más acertadas. En la definición de los microprogramas y microinstrucciones para las instrucciones en lenguaje de máquina, recuerde tener en cuenta consideraciones importantes del diseño de computadores, como son:

- Función y alcance apropiados para cada instrucción en lenguaje de máquina y para cada microinstrucción: este es un punto importante por cuanto es la forma en que la industria marca la diferencia entre los computadores de arquitectura RISC y CISC.
- Definición de los códigos binarios para cada instrucción: corresponde a la definición de códigos binarios para operadores y operandos de las instrucciones. Los códigos pueden ser de tamaño variables o fijo, en función de la arquitectura seleccionada. Típicamente los computadores en arquitectura RISC tienen códigos de operación de tamaño fijo, los de arquitectura CISC son de tamaño variable.
- Estados o ciclos del procesador: estados usuales en el diseño de computadores para la arquitectura en cuestión son a) obtener una instrucción de memoria (fetch), b) decodificar una instrucción a ejecutar (decode), c) ejecutar las acciones de una instrucción (execute), leer o escribir a memoria RAM (memory), etc.

Actividades a realizar

Usted debe seguir la siguiente lista de actividades, como plan de trabajo general, pero debe establecer un plan de trabajo específico para su grupo de trabajo y su proyecto, según se indica en la siguiente lista de actividades:

- Establecer y escribir el plan de trabajo específico para su proyecto.
- Definir y escribir el diseño de todos los componentes del computador.
- Construir y documentar el circuito digital. En su diseño debe considerar técnicas de construcción para circuitos modulares (análogo a la programación estructurada), técnicas de diseño de procesadores digitales y aprovechar las facilidades provistas por la herramienta de simulación para lograr un buen diseño.
- Evaluar ejemplos de máquinas de Turing en el computador resultante.
- Realizar informes de avance semanales durante el desarrollo del proyecto.

Metodología, cronograma y evaluación

Se trabajará en grupos de 2 ó 3 personas, según se establece en el programa del curso. Durante las lecciones el profesor dará guías generales sobre cómo construir el plan de trabajo, el diseño y la implementación del proyecto. Cada grupo acordará una estrategia de trabajo, establecerá un diseño, lo implementará, lo documentará y lo defenderá en clase.

Considere que antes de intentar un diseño, usted debe familiarizarse con el uso del simulador de circuitos. También debe realizar una investigación sobre la forma (técnicas, estrategias, algoritmos) de implementar el autómata y otros componentes de la máquina de Turing.

El proyecto se entregará por partes, como se indica a continuación (*primera fecha grupo 03, segunda fecha grupo 01*).

- **Inicio del proyecto:** entrega del enunciado. Fecha de inicio: *19 - 21 de marzo*.
- **Primer entregable:** un documento con el plan de trabajo para el proyecto. Fecha de entrega: *26 - 28 de marzo* (1 semana). Valor 20 %.
- **Segundo entregable:** un documento con el diseño detallado de la computadora. Fecha de entrega: *9 - 11 de abril* (2 semanas). Valor 20 %.
- **Tercer entregable:** implementación del computador, su documentación final y ejecución de los ejemplos de prueba. Opcionalmente, el grupo de estudiantes puede construir ejemplos de elaboración propia e introducir variantes al modelo de máquina de Turing, tales como suponer que la cinta de memoria se extiende infinitamente en ambas direcciones, que la cabeza puede tener 3 movimientos (derecha, izquierda, no moverse), que la lectura de un símbolo puede incluir un *wildcard* para que acepte leer cualquier valor no especificado explícitamente en otras instrucciones, etc. El profesor valorará la conveniencia, dificultad e implementación de las mejoras introducidas y otorgará hasta desde 0 hasta un máximo de 10 puntos adicionales en este rubro. Fecha de entrega *30 abril - 3 de mayo* (3 semanas). Valor 40 %.
- **Cuarto entregable:** Exposición y defensa del proyecto en clase y presentación de versión final de los ejemplos. Fecha: *14 - 16 de mayo* (1 semana). Valor 10 %.
- **Quinto entregable:** Informes de avance semanales. Fecha: cada día de clase. Valor 10 %.

Estructura del documento con el plan de trabajo

1. Introducción o descripción general: que describa las características principales del sistema a construir.
2. Marco teórico: una recopilación de los fundamentos teóricos que soportan el sistema a construir.
3. Diseño preliminar: contiene una primera versión, de alto nivel, con la especificación técnica del sistema a desarrollar. Incluye algunas decisiones y consideraciones para las siguientes etapas del diseño. También se recomienda incluir un índice preliminar del documento de diseño.
4. Cronograma de actividades: contiene la lista detallada de actividades a realizar, indicando una descripción suficientemente clara del alcance de cada una y las fechas propuestas de ejecución, considerando como marco de referencia las fechas de entrega del enunciado del proyecto. Planifique entregas semanales, con al menos un punto de revisión adicional a lo interno de su grupo de trabajo durante la semana.
5. Organización/distribución de tareas: describe la distribución de responsabilidades y trabajo entre los miembros del equipo y debe integrarse con el cronograma de actividades.
6. Referencias bibliográficas usadas para la elaboración del plan y que serán usadas inicialmente para la elaboración del proyecto.

Estructura de los informes de avance

1. Encabezado con la información de los miembros del equipo, la fecha del informe y el periodo de tiempo que se considera en el informe.
2. Una tabla con lista de actividades que estaban programadas para el periodo de tiempo reportado, indicando las que fueron ejecutadas, las que no fueron ejecutadas y un espacio para comentarios, principalmente para justificaciones en casos especiales.
3. Tareas realizadas por cada miembro del equipo.
4. Acuerdos relevantes, sean técnicos o logísticos, tomados durante la semana en relación con el desarrollo del proyecto.
5. Versión actualizada del cronograma restante.
6. Justificación a cambios en el cronograma de trabajo.

Referencias

1. Alfonseca M. (s.f.) *La máquina de Turing*. Las matemáticas del siglo XX. Tomado de: <http://www.sinewton.org/numeros/numeros/43-44/Articulo33.pdf>.
2. Von Neumann architecture. Wikipedia. https://en.wikipedia.org/wiki/Von_Neumann_architecture.
3. Sipser, M. Introduction to the Theory of Computation, 2nd Ed., Thomson Course Technology, 2006.