

Implementación cableada y microprogramada de la Unidad de Control

Kevin Andrés Hernández Rostrán

Escuela de Computación, Instituto Tecnológico de Costa Rica

Cartago, Costa Rica

kevinah95@gmail.com

Resumen

En este artículo se analiza cómo es la arquitectura de un procesador, el funcionamiento de sus unidades de control y de procesamiento, y además de dos tipos de implementaciones la cableada y microprogramada para que se estudie en una asignatura de primeros cursos de una Ingeniería Informática, ya sea técnica o superior. Se estudian también las características de un computador en su arquitectura, desde la apreciación de un programador. Finalmente, en cada una de las implementaciones se discuten sus ventajas e inconvenientes.

Introducción

está demostrado que el corazón del computador es el procesador, la importancia de su estudio trasciende en entender las dimensiones en las que un ingeniero piensa y analiza a la hora de crear un ordenador, que a la verdad, va más allá de la lógica y la aritmética, ello

porque dependiendo de su organización y su forma de implementación se pueden lograr resultados extraordinarios o simplemente un fracaso.

El objetivo primordial de la investigación consta en el manejo de términos, seguimiento de secuencias estructuradas, comprensión de las señales de control demandadas y registradas por la Unidad Central de Procesamiento (CPU). Se desea así un profundo análisis de cómo el procesador logra buscar datos, crear punteros, crear interrupción, de hecho, el procesador funciona con meras interrupciones y una cantidad exuberante de unos y ceros, cada dato en memoria es solamente eso un dato, por lo que de esto trata la arquitectura y diseño de un procesador, en decodificar y codificar instrucciones, mantener una secuencia, ser tolerante a fallos, manejo de banderas que en la lógica del computador puede significar lo que el diseñador moldeó en su arquitectura.

Arquitectura de un computador

Amdahl definió en 1964 la arquitectura de un computador como "la apariencia funcional que presenta a sus usuarios inmediatos". Es decir, los atributos o características de un sistema, visibles al programador.[1]

Todo dispositivo computable compuesto de transistores, circuitos integrados y componentes electrónicos mantiene una arquitectura en su diseño, para que funcione de forma lógica y aritmética. Un elemento fundamental del computador es el **procesador**, cuya arquitectura lógica es dada por un juego de instrucciones, denominadas instrucciones máquina, que constan de múltiples operaciones que se ejecutan, ayudándose de la sintaxis en un lenguaje denominado ensamblador, el cual describe la operación que realiza, los tipos de datos u operandos que puede utilizar, los códigos de condición que establece y el formato de la instrucción en su representación binaria. Este procesador debe tener una Unidad de Proceso y una Unidad de Control.

Como el computador solamente habla un lenguaje, todos sus compuestos deben poder comunicarse de la misma forma, por lo que el diseño de su estructura se basa ya sea en 32 o 64 bits. Esto quiere decir que para ejecutar estas instrucciones, es preciso que la Unidad de Proceso disponga de un banco de registros de tamaño adecuado: 32 ó 64 bits. También debe poseer un registro contador de programa de tamaño acorde con la memoria. Es necesario también un registro de instrucciones, cuyo tamaño dependerá del juego de instrucciones del procesador.

Registros del procesador

Se dividen generalmente en tres grupos: enteros, coma flotante y otros. En todos los procesadores de propósito general, sólo los registros enteros pueden almacenar punteros (una dirección de algún dato en memoria). Los registros que no son de enteros no se pueden utilizar

para almacenar punteros para leer o escribir memoria y por tanto no se pueden utilizar para evitar cualesquiera restricciones impuestas por el tamaño de los registros enteros. Es conveniente que el tamaño de instrucción coincida con el tamaño de los datos, ya que esto evitará problemas de alineamiento y facilitará el posterior diseño de la Unidad de Control y la Unidad de Procesamiento. [2]

Los registros del procesador se cargan con datos de la memoria o de dispositivos de entrada y salida, procesar dichos datos y devolver el resultado a la memoria o dispositivo de entrada y salida, si bien es cierto que a veces se puede actuar directamente sobre la memoria sin usar registros si la operación es sencilla. [3]

Los registros básicos son:

- Registros de Direcciones de Memoria (*Memory Address Register*, MAR)
 - Conectado al bus de direcciones.
 - Especifica la dirección en la que se lee o escribe un operando.
- Registro Separador de Memoria (*Memory Buffer Register*, MBR)
 - Conectado al bus de datos.
 - Almacena el dato a escribir o el último dato leído.
- Contador de Programa (*Program Counter*, PC)
 - Almacena la dirección de la próxima dirección a ser ejecutada.
- Registro de Instrucción (*Instruction Register*, IR)
 - Almacena la última instrucción captada. [4]

Unidad de Procesamiento

La Unidad de Procesamiento debe contener una unidad aritmético-lógica capaz de realizar las operaciones requeridas por las instrucciones, y un registro de estado con tantos bits de condición (flags) como requieran las

instrucciones de salto. Puede ser suficiente con disponer únicamente del bit de cero (Z) o se pueden incluir además el bit de signo (N), el bit de desbordamiento (V) y el bit de acarreo (C). También puede optarse por diseñar una máquina sin flags, en la que las instrucciones de salto consulten el valor de un registro previamente actualizado por una instrucción de comparación.

Unidad de Control

Detecta señales de estado procedentes de las distintas partes del ordenador y genera señales de control dirigidas a todas las unidades para controlar el funcionamiento de la máquina. Algunas de sus funciones también son:

- Captar de la memoria principal las instrucciones del programa que ejecuta el ordenador, las decodifica y las ejecuta una a una.
- Contiene un reloj que sincroniza todas las operaciones elementales involucradas en la ejecución de una instrucción.
- La frecuencia del reloj (en Megahertzios MHZ o Gigahertzios GHz) determina, en parte, la velocidad de funcionamiento del ordenador. [5]

Temporización de las señales de control

El funcionamiento de un computador es síncrono, esto quiere decir que el transmisor debe coordinarse con el receptor antes del envío de datos dependiendo de los pulsos del reloj y para ello se ejecuta una instrucción que tiene diferentes fases: Fetch (búsqueda de una instrucción), decodificación y búsqueda de operandos, operación y almacenamiento del resultado. El número de ciclos de reloj varía según las instrucciones, depende del modo de direccionamiento y el tipo de operación. [6]

La secuencia de búsqueda (Fetch), realiza una secuencia: La dirección de la instrucción está en el PC. El valor de la dirección (MAR) se coloca en el bus de direcciones. La Unidad de Control genera un comando READ. El resultado (un dato desde la memoria) aparece en el bus de datos. El dato se copia al MBR desde el bus de datos. El PC se incrementa en 1 (en paralelo con la búsqueda del dato desde la memoria). El dato (instrucción) se mueve del MBR al IR. El MBR queda libre para nuevas búsquedas de datos. [4]

Cada fase puede descomponerse en una serie de micro-operaciones elementales, por ejemplo, para los ciclos de búsqueda, indirecto y de interrupción, para el ciclo de ejecución (hay una secuencia de micro-operaciones para cada código de operación). Se unen las secuencias de micro-operaciones y con un nuevo registro de 2 bits necesario para almacenar el Código de Ciclo de Instrucción (Instruction Cycle Code, ICC), que determina el estado del procesador en términos de que parte del ciclo se encuentra:

- 00: Búsqueda
- 01: Indirecto
- 10: Ejecución
- 11: Interrupción

Bus interno del procesador

Dada la complejidad de las conexiones, lo usual es utilizar un bus interno por el que circulan direcciones y datos. Las compuertas (AND, XOR, etc.) controlan el movimiento de datos desde y hacia el bus interno. Las señales de control manejan la transferencia de datos desde y hacia un bus externo. Es necesario agregar registros temporales (de entrada y salida) para asegurar una operación apropiada de la Unidad de Procesamiento. [4]

Unidad de Control mediante lógica cableada

La unidad de control es una máquina secuencial, en la que las salidas dependen del estado en el que se encuentra y no de las entradas únicamente. El número de estados depende del número de operaciones elementales a realizar, lo que dará el número de biestables para realizarlos. Inicialmente la unidad de control está en estado de espera y necesita una señal externa para cambiar d estado. Su control está basado en una arquitectura fija, es decir, que requiere cambios en el cableado si el conjunto de instrucciones es modificado o cambiado. Esta arquitectura es preferida en los computadores RISC pues consiste en un conjunto de instrucciones más pequeño. Las unidades de control cableado son implementadas por medio de unidades de lógica secuencial ofreciendo un número finito de puertas que pueden actuar como generadores de resultados específicos, basada en las instrucciones que fueron usadas para invocar esas respuestas. Estas instrucciones son evidentes en el diseño de la arquitectura, pero también pueden ser representadas de otras maneras. [5],[7],[8],[9]

La implementación de este circuito puede hacerse mediante:

- Método de la Tabla de Estados
- Método de las Células de Retardo
- Método del Contador Secuenciador

Método de la Tabla de Estados

El control pasa por una serie de estados que dependiendo de las entradas determinan el estado siguiente y qué señales de control se activan. Se basa en una máquina de estados finitos. Una máquina de estados finitos consta de:

- Memoria interna que contiene el estado. Cada estado corresponde a un ciclo de reloj y contiene las operaciones a realizar en ese ciclo.
- Función de estado siguiente. Función combinacional que a partir de las entradas y el

estado actual determina el estado siguiente, es decir, dice a qué estado pasa, dependiendo del estado en el que se encuentra.

- Función de salida. Produce el conjunto de señales de control a partir de sus entradas y el estado actual. [8]

Camino de datos en lo que se supone un repertorio RISC, siendo las instrucciones del tipo registro-registro. El repertorio de instrucciones será:

- Instrucciones de memoria:
 - LD R, Memoria (R <-- Contenido de la Memoria)
 - ST Memoria, R (Contenido de la Memoria <-- R)
- Instrucciones de proceso:
 - ADDI / SUBI R1, R2, Inmediato (R1 = R2 +/- Dato inmediato)
 - ADD / SUB R1, R2, R3 (R1 = R2 +/- R3)
- Instrucciones de control:
 - JZ dirección de salto (CP <-- CP + Desplazamiento)
 - JMP Dirección de salto (CP <-- Dirección de salto)

Los modos de direccionamiento permitidos son el inmediato, para las instrucciones de proceso; el relativo a registro, para las instrucciones de acceso a memoria y la de salto condicional; y el directo a memoria, para la instrucción de salto incondicional.

El formato de todas las instrucciones es de 32 bits, en las que la máquina de estados generará 15 estados enumerando cada estado con 4 bits, por ejemplo, el estado 0 (0000), el estado 1 (0001), el estado 2 (0010), etc. Generando ecuaciones de estado y finalizando con las señales de control. [5],[7],[8],[9]

Células de retardo

El diseño se basa en implementar el diagrama de estados mediante unos elementos que únicamente proporcionan un retardo llamados células de retardo. Las líneas de control se conectan a las salidas de las células de retardo que representan el estado en el que se deben activar. La función de las células de retardo es la de sincronizar la secuencia de las señales de control. Una vez realizadas las conexiones se introduce un único pulso por el circuito. El pulso circula por el circuito activando las señales adecuadas en cada momento. [5],[7],[8],[9]

Contador secuenciador

Emplea un contador que marca los instantes en que se deben activar las señales. La unidad de control debe conocer además, la instrucción que se está ejecutando y la información del estado de la máquina, con esa información las instrucciones recibirán, para ejecutarse, las señales apropiadas en el momento preciso. El secuenciador AMD 2910, por ejemplo, tiene un contador de microinstrucciones con autoincremento, una micropila de cinco niveles, un contador con detector de cero para la realización de microbuses, un multiplexor para determinar la dirección de la siguiente microinstrucción y señales para recoger la salida de un comparador de condición de bifurcación para producir microbifurcaciones condicionales. [5],[7],[8],[9]

Ventajas y desventajas

La ventaja es su circuito que es más rápido que el de lógica almacenada.

Los inconvenientes son que requiere de cambios en el cableado si se modifica el juego de instrucciones y la laboriosa tarea de diseño. [10]

Unidad de Control Microprogramada

La implementación microprogramada se basa en la codificación de las micro operaciones en un lenguaje simbólico, lo que es conocido como lenguaje de microprogramación. Cada línea describe un conjunto de micro operaciones que ocurren simultáneamente, y se denomina microinstrucción. Una secuencia de microinstrucciones se denomina microprograma o 'firmware'. Para cada micro operación la unidad de control debe generar señales de control determinadas, por lo que pueden codificarse en una palabra de control en la que hay un bit por cada línea de control.

No obstante, utilizar una palabra de control de este tipo podría dar problemas cuando la secuencia de microoperaciones no coincide entre instrucciones diferentes (por ejemplo, a veces se necesita un ciclo indirecto y a veces no). Podemos aproximarnos a la solución almacenando las palabras de control en una memoria en la que cada palabra tenga una única dirección. De esta manera, se puede añadir un campo de dirección a cada palabra de control que indique dónde se almacena la siguiente palabra de control a ejecutar si se cumplen determinadas condiciones y algunos bits para especificar la condición.

El resultado de esto se conoce como microinstrucción horizontal. Tiene un bit para cada línea de control interna de la CPU y un bit para cada línea del bus de control externo. Hay un campo de condición que indica la condición para la que se producirá el salto, y otro campo con la dirección de la siguiente instrucción a ejecutar en este caso. La microinstrucción se interpretará de la siguiente forma:

- Para ejecutarla, poner a estado alto todas las líneas de control cuyo bit representativo esté a '1', y a bajo las que estén a '0'. Las señales resultantes pueden realizar una o más microoperaciones.
- Si la condición indicada no se cumple, ejecutar la siguiente microinstrucción.

- Si la condición se cumple, ejecutar la microinstrucción indicada por el campo de direcciones.

En microinstrucciones verticales, se utiliza un código para cada acción a realizar, y un decodificador convierte este código en señales individuales de control. La ventaja de este tipo es que es más compacto que la microinstrucción horizontal a expensas de un pequeño incremento de lógica y retraso temporal.

Las microinstrucciones de cada rutina se ejecutan secuencialmente. Cada rutina termina con un salto que indica por dónde se debe continuar. El ciclo de ejecución representa las distintas rutinas que implementan las instrucciones máquina a ejecutar dependiendo del código de operación actual. [11]

Técnicas de secuenciamiento

Las basadas en la microinstrucción actual, en las banderas de condición, en los contenidos del IR o la dirección de la memoria de control al generarse.

También se basan en el formato de la información de la dirección:

- Dos campos de dirección
- Un campo de dirección
- Formato variable

Secuenciamiento Explícito

Cada microinstrucción incluye la dirección de la siguiente. Los códigos de operación apuntan al comienzo del microprograma. Cada microinstrucción incluye: un campo de dirección, señales de control y un bit que indica si es la última. Su ventaja es la repetición de microinstrucciones.

Secuenciamiento Implícito

Consiste en tener ordenadas secuencialmente todas las microinstrucciones de cada microprograma. Se necesita

de un contador de microprograma que apunte a la microinstrucción en curso. La primera microinstrucción de cada microprograma en la ROM que cuenta con los códigos de operación y las posiciones de la primera microinstrucción.

Codificación de las microinstrucciones

Al codificar las microinstrucciones se puede ahorrar memoria de control. La microprogramación horizontal no usa codificación. Son formatos largos, pero rápidos. La microprogramación vertical es de alta codificación para un menor tamaño de memoria, pero mayor lentitud para decodificación.

Microprocesadores actuales

Cuentan con muchas instrucciones asociadas al nivel de registro del hardware. Múltiples puntos de control a ser manipulados, esto resulta en memoria de control que contiene un número grande de palabras, correspondiente al número de instrucciones a ser ejecutadas. Tiene un tamaño amplio de palabra debido al gran número de puntos de control a ser manipulados.

Ventajas e inconvenientes

Las ventajas son: la fácil depuración de las microinstrucciones al estar en ROM, para las instrucciones complejas basta con reservar más memoria para ejecutarlas, simplifica el diseño de la unidad de control, es más económica, menos propensa a errores, basta con cambiar los microprogramas para tener diferente juego instrucciones y permite la emulación de máquinas.

Por otro lado el inconveniente es que todas sus rutinas y subrutinas se vuelven más lentas por el proceso de depuración.

Conclusiones

El trato de objetivos del informe es muy superficial, pero lo bastante conciso como para entender el funcionamiento del CPU.

Se segmenta cada parte del procesador, ahora es sencillo entender la estructura y control del mismo, resaltando la idea del uso de memoria, la coordinación de algoritmos, el secuenciamiento de instrucciones. La lógica computable hace ver que este universo puede ser sencillo y sin complicación, pero con déficits en la localización de errores o complejo y bien depurado, pero un poco lento. De ahí la elección del diseñador rapidez o precisión.

Referencias

[1] F. Aylagas Romero, Notas y transparencias de arquitectura de computadores, 1st ed. [Madrid]: Departamento de Publicaciones de la Escuela Universitaria de Informática de la Universidad Politécnica de Madrid, 2006.

[2] F. Sánchez, Características deseables en un procesador pedagógico para la enseñanza básica de Arquitectura de Computadores, Universidad Politécnica de Catalunya, Barcelona, 2001.

[3] F. Oterino Echávarri, ESTRUCTURA DE COMPUTADORES I, Universidad del país Vasco, España, 2002.

[4] W. Stallings, Computer organization and architecture, 5th ed. Upper Saddle River, N.J.: Prentice Hall, 2000.

[5] A. Prieto Espinosa, A. Lloris Ruíz and J. Torres Cantero, Introducción a la informática, 3rd ed. Madrid: McGraw-Hill, 2002.

[6] J. M. López, Unidad de Control, Universidad de Alcalá, España.

[7] J. Angulo Usategui, J. García Zubía and I. Angulo Martínez, Fundamentos y estructura de computadores, 1st ed. Madrid: International Thomson, 2003.

[8] D. Patterson and J. Hennessy, Estructura y diseño de computadores, 1st ed. Barcelona [etc.]: Reverté, 2011.

[9] J. Frutos Redondo and R. Rico López, Arquitectura de computadores, 1st ed. [Alcalá de Henares, Madrid]: Universidad de Alcalá de Henares, Depart. de automática, 1995.

[10] K. Leal Algara, Unidad de control, Universidad Rey Juan Carlos, Madrid, 2011.

[11] J. F. Guerrero Martínez and J. V. Francés Villora, Introducción a los Microprocesadores, Universidad de Valencia, España, 2010.