

Resumen ORGA I

Diaz Julian Lautaro

22 de julio de 2018

Índice

1. Introducción	3
2. Sistemas de representación	3
2.1. Representación de enteros	3
2.2. Representación de reales	4
3. Lógica Digital	4
3.1. Clock	4
3.2. Flip-Flops	4
4. Arquitectura	4
4.1. Memoria	5
4.2. Unidad de Control	5
4.3. Ciclo de instrucción	5
4.4. Diseño ISA	6
4.5. Registros	8
4.6. Unidad Aritmético Lógica	8
4.7. Buses	8
4.8. Clocks	9
5. E/S	9
5.1. Interrupciones	9
5.2. Polling	10
5.3. DMA	10
6. Microarquitectura	10
7. Memorias	12
7.1. Jerarquía de memoria	13

8. Performance	14
9. Bibliografía	14

1. Introducción

- Arquitectura del computador: atributos visibles al programador(interfaz).
 - Set de registros internos.
 - Set de instrucciones.
 - Bits utilizados para representar los datos.
 - Mecanismos de direccionamiento de memoria.
 - Acceso a dispositivos de entrada y salida.
 - ...
- Organización del computador: cómo se implementa la arquitectura(implementación).
 - Señales de control.
 - Tecnologías de la memoria.
 - ...

2. Sistemas de representación

Conjunto de símbolos y reglas que permiten representar los números.

2.1. Representación de enteros

- Sin signo: todos los bits representan la magnitud y se pueden representar los números en el rango $[0, r^d - 1]$, con r la base y d la cantidad de dígitos.
- Signo+Magnitud: el primer bit se utiliza para indicar el signo (1 negativo, 0 positivo) y el resto para la magnitud. N bits pueden representar los números en el rango $[-2^{N-1} - 1, 2^{N-1} - 1]$.
- Complemento a 1: dado un número n en base r con d dígitos, el complemento a 1 se define como $(r^d - 1) - n$. Se calcula negando todos los bits. N bits pueden representar los números en el rango $[-2^N - 1, 2^N - 1]$.
- Complemento a 2: dado un número n en base r con d dígitos, el complemento a 2 se define como $r^d - n$. Se calcula negando todos los bits y sumando 1. N bits pueden representar los números en el rango $[-2^N, 2^N - 1]$.
- Exceso a m : dado un número n se representa como $n + m$.

2.2. Representación de reales

3. Lógica Digital

- Compuerta: dispositivo electrónico que produce un resultado en base a un conjunto de valores de entrada.
- Circuitos: combinación de compuertas que implementan una función booleana. Se dividen en:
 - Combinatorios: dependen únicamente de la entrada.
 - Secuenciales: dependen de la entrada y del estado actual.
- Conjunto Universal: un conjunto universal es aquel conjunto de compuerta/s que puede/n construir cualquier circuito utilizando únicamente a éstas.

3.1. Clock

Los circuitos secuenciales pueden ser asincrónicos o sincrónicos. Los asincrónicos pueden cambiar su estado en cualquier momento en el que se modifique su entrada, los asincrónicos utilizan un reloj para ordenar los eventos. Un clock es un circuito que emite una serie de pulsos con un ancho y un intervalo entre pulsos consecutivos determinado. Este intervalo se llama ciclo de clock y está dividido en flanco ascendente y descendente. El clock es utilizado por un circuito secuencial sincrónico para saber cuando los valores de entrada van a modificar su estado.

3.2. Flip-Flops

Un flip-flop es un circuito secuencial retroalimentado, lo que permite almacenar bits. Se puede describir el comportamiento de un flip-flop a través de una tabla característica, que muestra cual va a ser el siguiente estado basado en las entradas y en el estado actual. Ejemplos de flip-flops son el flip-flop D, SR y JK. Este último es universal, ya que puede utilizarse para armar los otros dos.

4. Arquitectura

Todas las computadoras tienen un CPU (Central Processing Unit). Esta unidad se divide en dos partes. La primera es el datapath que es una red

de unidades de almacenamiento (registros) y unidades lógicas y aritméticas (para realizar operaciones sobre los datos) conectadas a través de buses (cables que permiten enviar información) donde la sincronización es controlada por clocks. La segunda parte es la UC (Control Unit), un módulo responsable de ejecutar operaciones y asegurarse de que los datos sean correctos y estén donde tengan que estar y en el tiempo correcto. Juntos, estos componentes realizan las tareas del CPU: obtener las instrucciones, decodificarlas y finalmente ejecutar la secuencia indicada de operaciones.

4.1. Memoria

La memoria puede pensarse como una matriz de bits. Cada fila, implementada por un registro, tiene un cierto tamaño. Cada registro (posición en memoria) tiene una única dirección. Las direcciones suelen representarse con enteros sin signo. Se le llama palabra a la cantidad de bits que pueden moverse simultáneamente dentro de la CPU (unidad de transferencia). Al tamaño del registro en cada fila se le llama unidad direccionable.

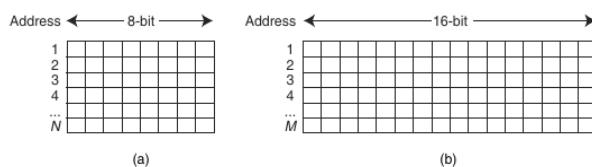


Figura 1: (a) Memoria de tamaño $8 * N - \text{bit}$ y unidad direccionable de 8-bit
(b) Memoria de tamaño $16 * N - \text{bit}$ y unidad direccionable de 16-bit

4.2. Unidad de Control

La unidad de control es la que gestiona el tráfico del CPU. Ésta monitorea la ejecución de todas las instrucciones y transfiere toda la información, extrayendo la información de la memoria y decodificando las instrucciones, asegurándose de que los datos estén en el lugar y momento correcto. La UC usa un registro especial llamado PC (Program Counter) para encontrar la próxima instrucción y, en caso de ser necesario, utiliza registros de estado (flags) para saber en que estado se encuentra el programa.

4.3. Ciclo de instrucción

El ciclo de instrucción representa los pasos que una computadora sigue para correr un programa. Este ciclo tiene 3 etapas:

1. Fetch:
 - La UC obtiene la próxima instrucción yendo a buscarla a la memoria principal.
 - La UC incrementa el PC.
2. Decode: la UC determina el código de operación y carga todos los datos necesarios para llevar a cabo la instrucción.
3. Execute: la UC realiza la operación indicada por la instrucción.

4.4. Diseño ISA

El Instruction Set Architecture (ISA) es el conjunto de instrucciones de una arquitectura. Estas instrucciones se diferencian por varios aspectos:

- Métricas.
 - Cantidad total de instrucciones disponibles.
 - Complejidad del conjunto de instrucciones.
 - Reduced Instruction Set Computer (RISC): se simplifican las instrucciones para que se puedan ejecutar más rápido. Cada instrucción realiza una única operación, son todas de tamaño fijo y los operandos son únicamente registros.
 - Complex Instruction Set Computer (CISC): hay una gran cantidad de instrucciones, de longitud variable y una única instrucción puede ejecutar múltiples operaciones.
 - Longitud de las instrucciones.
 - Cantidad de memoria que un programa requiere.
- Tipos de datos soportados.
 - Enteros 16/32-bits
 - Puntos flotante 16/32-bits.
 - ...
- Almacenamiento de los datos.
 - ¿Cómo se almacenan los bytes?
 - Big Endian: se guardan los bits más significativos primeros.

Address →	00	01	10	11
Big Endian	12	34	56	78
Little Endian	78	56	34	12

Figura 2: El valor hexadecimal 12345678 almacenado en Big y Little Endian

- Little Endian: se guardan los bits menos significativos primeros.
- ¿Dónde se almacenan?
 - Arquitectura de stack: esta arquitectura utiliza una pila para ejecutar instrucciones y los operandos se encuentran al tope de ésta.
 - Arquitectura de acumulador: se tiene un registro especial (AC) que se utiliza como operando implícito.
 - Arquitectura con registros de propósito general.
 - ◇ Memoria-Memoria: puede tener muchos operandos en memoria sin recurrir a registros.
 - ◇ Registro-Memoria: operandos en memoria y en registros.
 - ◇ Cargar-Almacenar: operandos únicamente en registros.
- ¿Cómo se acceden?
 - Inmediato: el valor a ser referenciado sigue inmediatamente al código de operación.
 - Directo: el valor a ser referenciado se obtiene especificando su posición en memoria.
 - Indirecto: el valor a ser referenciado se obtiene en la posición de memoria contenida en la posición en memoria especificada.
 - Registro: el valor a ser referenciado se obtiene en el registro especificado.
 - Indexado: el valor a ser referenciado se obtiene en la posición de memoria resultante de sumar una constante con la posición de memoria en el registro especificado.
- ¿Qué operaciones puede ejecutar?
 - Movimiento de datos.
 - Aritméticas.
 - Lógicas.
 - E/S.

- Transferencia de control.
- Específicas.
- ¿Cómo se codifican las operaciones?
 - Código de operación: representa la operación a utilizar.
 - Fuente/s: información para obtener operandos de origen.
 - Destino/s: información para obtener la ubicación del resultado de la operación.
 - Referencia a próxima instrucción: información para determinar el próximo valor del PC.

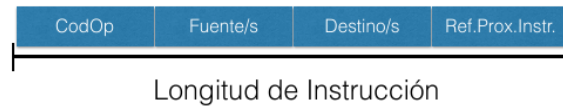


Figura 3: Modelo de instrucción

4.5. Registros

Los registros son utilizados para almacenar una amplia variedad de datos como direcciones, contadores o datos necesarios para la ejecución del programa y son controlados por la UC. Los registros están en el procesador, eso hace que la información pueda ser accedida rápidamente. Pueden ser implementados usando flip-flops D en paralelo.

4.6. Unidad Aritmético Lógica

La Unidad Aritmético Lógica (ALU) lleva a cabo las operaciones lógicas (como comparaciones) y aritméticas (como la suma, resta o multiplicación) requeridas durante la ejecución del programa. Ésta es controlada, a través de señales que indican que operación realizar, por la UC.

4.7. Buses

La CPU se comunica con los otros componentes a través del bus. Un bus es un conjunto de cables que actúa como una ruta para conectar múltiples subsistemas dentro del sistema. Éstos pueden conectar únicamente dos dispositivos específicos (dedicado) o muchos a la vez (multiplexado). En general,

los dispositivos se dividen en master y slave, donde el maestro es aquel que empieza la acción y el esclavo el que responde a ella. Hablamos de escritura cuando el master quiere enviar datos y de lectura cuando quiere recibirlos. Para evitar escrituras simultáneas es necesario un protocolo de bus y para esto se utilizan líneas de control que indican que dispositivo tiene permiso para usar el bus y para que propósito. Los protocolos pueden ser sincrónicos o asincrónicos. Los buses que conectan el procesador con la memoria suelen ser cortos y rápidos para maximizar el ancho de banda. En cambio, los buses de Entrada/Salida (E/S) son más largos y permiten ser utilizados por distintos tipos de arquitecturas.

4.8. Clocks

Todas las computadoras tienen un clock interno que regula que tan rápido se pueden ejecutar las instrucciones. El clock regula el ritmo de todo lo que pasa en el sistema. La CPU requiere un determinado número de ciclos de clock para ejecutar cada instrucción. Algunos buses también tienen su propio clock y éste suele ser más lento que el del CPU.

5. E/S

Los dispositivos de E/S nos permiten comunicarnos con la computadora. E/S es la transferencia de datos entre la memoria primaria y varios periféricos de E/S. Hay una interfaz que maneja la transferencia de datos. Esta interfaz convierte las señales del bus en un formato reconocible por el dispositivo. La CPU se comunica con estos dispositivos externos a través de registros específicos de E/S. La interfaz puede mapear los registros a la memoria principal o el dispositivo puede tener registros propios y la CPU utiliza instrucciones especiales para poder comunicarse. A continuación vemos dos maneras para controlar la transferencia de información entre CPU y dispositivos de E/S.

5.1. Interrupciones

Las interrupciones son eventos que alteran la normal ejecución del programa. El dispositivo de E/S interrumpe a la CPU cuando quiere comunicarse. En este momento la CPU guarda el estado actual en el que se encontraba antes de la interrupción y ejecuta la Rutina de Atención de Interrupciones (RAI), un programa que se carga en alguna posición de memoria. Al finalizar la RAI se vuelve al estado anterior del programa. En la Figura 4 se puede ver como puede ser implementado un sistema para manejar interrupciones.

Cuando el dispositivo interrumpe el controlador de interrupciones activa la señal Interrupt (INT). Cuando la CPU está lista para procesar la interrupción activa la señal Interrupt Acknowledge (INTA). Luego el controlador puede bajar la señal INT.

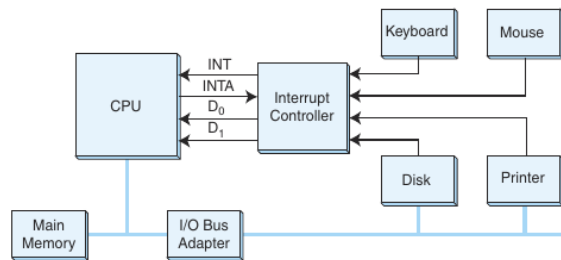


Figura 4: Modelo de E/S con interrupciones

5.2. Polling

En este caso, en cada ciclo antes de realizar el fetch, la CPU, a través de un flag, se fija si el dispositivo de E/S solicitó comunicarse y de ser así interrumpe la ejecución del programa realizando lo mismo que con interrupciones.

5.3. DMA

Con polling e interrupciones la CPU mueve datos hacia y desde los dispositivos de E/S. Para alivianar el trabajo de la CPU se implementa un chip dedicado, llamado Direct Memory Access (DMA), para poder acceder a la memoria sin tener que recurrir a la CPU. La CPU le envía una señal a la DMA y procede con la siguiente tarea, mientras la DMA se ocupa del dispositivo de E/S. Cuando finaliza, la DMA envía otra interrupción a la CPU.

6. Microarquitectura

Cada instrucción está compuesta por microinstrucciones. Estas microinstrucciones realizan pasos muy simples, por ejemplo poner cosas en registros, activar circuitos o prender/apagar líneas. La UC es la encargada de ejecutar estas microinstrucciones, a través de señales de control. Hay dos maneras de implementar la UC y ambas tienen señales de tiempo para saber en qué paso están. Estas señales se generan con un simple contador. Por ejemplo,

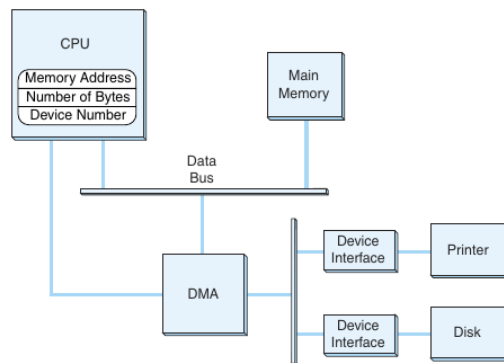


Figura 5: Modelo de DMA

una arquitectura podría incluir las señales de tiempo T_1, \dots, T_8 , el fetch de una instrucción podría ocurrir cuando se active T_1 y el fetch de un operando cuando se active T_4 , etcétera.

- UC Cableada (Hardwired): esta UC se implementa utilizando hardware (circuitos). Los circuitos van a tener como entradas el código de operación, los flags, señales del bus y señales del clock. Esta UC es muy rápida, pero difícil de diseñar y modificar.

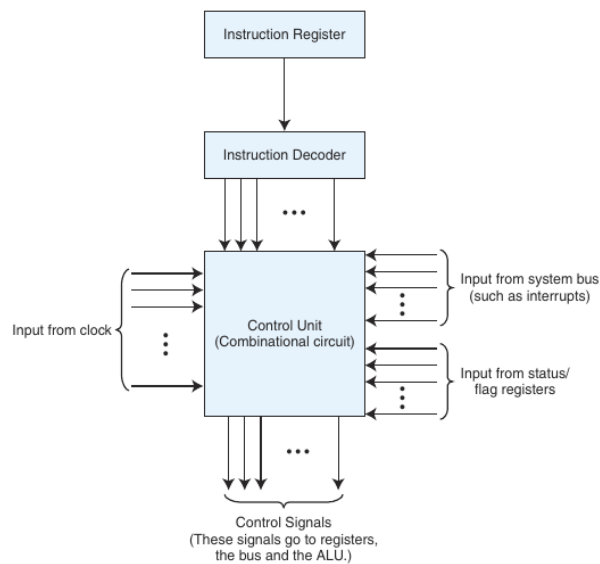


Figura 6: Ejemplo de UC cableada

- UC Microprogramada: todas las instrucciones son entradas a un microprograma, que convierte estas instrucciones en señales de control. Este microprograma es programa escrito en un lenguaje de bajo nivel que se implementa directamente por hardware. Para cada instrucción hay una subrutina y para modificar una instrucción o agregar una nueva alcanza con modificar la subrutina o agregar otra. La ventaja de esto es que es flexible (no hace falta modificar el hardware) y simple de diseñar, pero más lenta que la cableada.

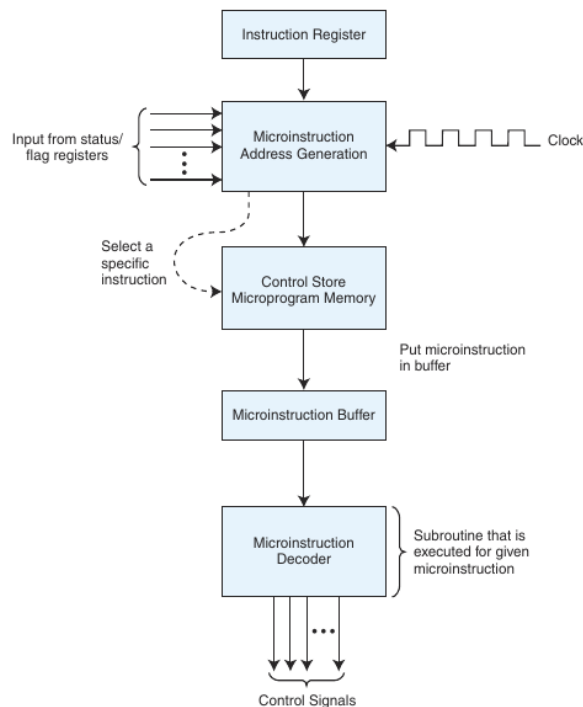


Figura 7: Ejemplo de UC microprogramada

7. Memorias

Hay dos tipos básicos de memoria:

- RAM (Random Access Memory): la memoria RAM es la memoria principal. Se utiliza para almacenar programas y datos que la computadora necesita cuando se ejecutan los programas. Esta memoria es volátil, pierde su información al cortarse la corriente.

- DRAM (dynamic): construida con pequeños capacitores que pierden electricidad. Necesita una recarga cada pocos milisegundos para mantener sus datos. Es más densa y usa menos corriente que la SRAM. Se suele usar para la memoria principal.
- SRAM (static): construida con circuitos similares a los flip-flops D. Mantiene sus datos mientras tenga corriente. Es más rápida, pero más cara que la DRAM. Se suele usar para la memoria caché.
- ROM (Read Only Memory): en esta memoria se almacena la información necesaria para usar el sistema, como el programa necesaria para arrancar la computadora. La ROM no es volátil, mantiene su información aunque se corte la corriente. Este tipo de memoria suele ser utilizada en sistemas encrustados o en cualquier sistema donde la información no necesite modificarse. Se dividen según cuando pueden ser escritas.
 - ROM: se escribe una única vez en la fábrica.
 - PROM (Programmable): puede ser escrita por el usuario una única vez.
 - EPROM (Erasable): puede ser escrita varias veces por el usuario, con luz ultravioleta, eliminando toda la información de antemano.
 - EEPROM (Electrical): puede ser escrita o borrada varias veces de manera electrónica y se borra un byte a la vez.
 - Memoria Flash: igual a la EEPROM, pero permite borrar de a bloques.

7.1. Jerarquía de memoria

Vimos que no todas las memorias son creadas de la misma manera, y algunos tipos son menos eficientes y más baratos que otros. Para tratar esta disparidad, las computadoras utilizan una combinación de tipos de memoria para obtener la mejor relación precio-eficiencia posible. A este enfoque se le llama jerarquía de memoria. Cuanto más rápida es una memoria, más caro es el almacenamiento por bit. Los tipos básicos que normalmente constituyen la jerarquía de memoria son los registros, la caché, la memoria principal y la memoria secundaria. Hoy en día, las computadoras tienen pequeña cantidad de memoria muy rápida, llamada caché. La caché está conectada a la memoria principal, que es más grande y de mediana rapidez. La memoria principal se complementa con una memoria secundaria mucho más grande y lenta. Estas memorias las clasificamos según cuantos ciclos le requieren a la CPU

para poder acceder a ellas. A continuación vemos terminología usada cuando hablamos de jerarquía de memoria. La caché es un tipo de memoria, chica y rápida, que sirve como buffer para un acceso a datos frecuente.

8. Performance

9. Bibliografía

- The Essentials of Computer Organization and Architecture - Linda Null, Julia Lobur
- Apuntes 1^{er} cuatrimestre 2018