



Universidad Nacional de la Patagonia San Juan Bosco
Facultad de Ingeniería. Sede Puerto Madryn
Arquitectura de Computadoras

Apunte para TP 2: Unidad de Control

La Unidad de Control es responsable de gestionar las señales necesarias para coordinar la operación de los diferentes bloques de la computadora. Se encarga de interpretar instrucciones y activar los componentes adecuados para ejecutar cada operación.

Funciones de la Unidad de Control

- Coordinar la ejecución de instrucciones mediante señales de control.
- Sincronizar la transferencia de datos entre los registros y la memoria.
- Gestionar el flujo de información entre la ALU, registros y buses.
- Controlar saltos y decisiones en la ejecución del programa.

Términos clave

- Puntos de control: Señales que activan distintos bloques de la computadora.
- Microinstrucción: Sentencia de control que define una operación.
- Microoperación: Acción específica sobre un punto de control, como activar la habilitación de MDR para que actúe como transmisor del bus.

Tipos de Microprogramación

- Horizontal: Cada microinstrucción controla muchos puntos de control simultáneamente, permitiendo operaciones complejas en menos ciclos. Se asigna un bit de ancho por cada punto de control.
- Vertical: Cada microinstrucción realiza una sola operación, con mayor codificación y menor ancho de palabra, lo que hace que el microprograma sea más largo.
- Oblicua: Combina horizontal y vertical, agrupando operaciones en campos codificados para un uso eficiente de la memoria de control. Por ejemplo, si hay 8 registros, se pueden codificar en 3 bits.

Secuenciamiento y Micromemoria

El secuenciamiento determina la siguiente microinstrucción a ejecutar. Existen dos tipos:

- Explícito: La microinstrucción contiene un campo con la siguiente dirección (usado en microprogramación horizontal y a veces en vertical).
- Implícito: Se usa un microcontador de programa que avanza secuencialmente, con posibilidad de saltos condicionales o incondicionales.

La micromemoria es la memoria de control donde se almacenan microinstrucciones que determinan el comportamiento del procesador. Puede ser una ROM (para control cableado) o una memoria reprogramable (para microprogramación).



Universidad Nacional de la Patagonia San Juan Bosco
Facultad de Ingeniería. Sede Puerto Madryn
Arquitectura de Computadoras

Registros clave

- IR (Instruction Register): Contiene la instrucción en ejecución.
- PC (Program Counter): Guarda la dirección de la próxima instrucción a ejecutar.
- MAR (Memory Address Register): Contiene la dirección de memoria a acceder.
- MDR (Memory Data Register): Almacena datos leídos o escritos en memoria.
- ACC (Acumulador): Registro usado para operaciones aritméticas y lógicas.
- SR (Status Register): Indica banderas de estado como carry, overflow o cero.

Funcionamiento de las Microrrutinas

Antes de ejecutar un programa, el sistema operativo:

1. Carga el programa en memoria.
2. Inicializa el PC con la dirección de la primera instrucción.
3. La Unidad de Control lee y ejecuta instrucciones hasta finalizar el programa.

Fases de Ejecución

1. Búsqueda de instrucción: Se carga la instrucción desde memoria en el IR.
2. Decodificación: Se identifica el código de operación y operandos. Una forma práctica es hacer que el IR seleccione directamente la dirección de micromemoria a la que se debe ir.
3. Ejecución: Se realiza la operación correspondiente con la ALU o registros.
4. Acceso a memoria: Si la instrucción requiere lectura o escritura en memoria.
5. Actualización del PC: Se apunta a la siguiente instrucción o se salta según la lógica del programa.

Microprogramación

Pasos para diseñar microprogramación:

1. Definir el set de instrucciones: Describir cómo funcionan las instrucciones en términos lógicos.
2. Definir la secuencia RTL (Register Transfer Level): Desglosar los pasos necesarios para ejecutar cada instrucción.
3. Asignar microinstrucciones: Determinar qué microinstrucciones se requieren y en qué orden deben ejecutarse.

Ejemplo: Ejecución de la instrucción ADD B

MAR \leftarrow PC

READ (MDR \leftarrow Mem[MAR])

Oper1 \leftarrow A

Oper2 \leftarrow B

Resultado \leftarrow Oper1 + Oper2



Universidad Nacional de la Patagonia San Juan Bosco
Facultad de Ingeniería. Sede Puerto Madryn
Arquitectura de Computadoras

$A \leftarrow \text{Resultado}$

$PC \leftarrow PC + 1$

Manejo de la Pila y Subrutinas

Para ejecutar llamadas y retornos de subrutinas, se usa una pila en memoria con un Stack Pointer (SP):

- CALL: Guarda la dirección de retorno en la pila antes de saltar.
- RET: Recupera la dirección desde la pila y actualiza el PC.

Ejemplo de PUSH y POP:

- PUSH: Guarda un valor en la pila y decrementa SP.
- POP: Recupera un valor de la pila y aumenta SP.

En una implementación práctica, es importante definir si el SP apunta al último ocupado o al primero libre.

Formato de Instrucciones

Las instrucciones se organizan en campos:

- Código de Operación (CO): Indica la operación a realizar.
- Operandos: Datos o direcciones utilizados en la operación.

Modos de Direccionamiento

Direccionamiento Absoluto

En este modo, el operando es una dirección de memoria fija, es decir, la instrucción contiene directamente la dirección del dato en memoria. Esto significa que durante la ejecución de la instrucción, se realiza la siguiente operación:

- $MAR \leftarrow AP$ (donde AP contiene la dirección absoluta del operando en memoria)

Luego, se accede a la memoria para obtener el valor almacenado en esa dirección. Este modo es útil para trabajar con datos que se encuentran en ubicaciones fijas dentro del programa.

- Inmediato: El operando está dentro de la instrucción.
Ej: LOAD X: Carga el valor en la dirección X al acumulador.
STORE X: Guarda el valor del acumulador en la dirección X.
- Directo: La instrucción contiene la dirección del operando.
Ej: LOAD [x] → trata a x como la dirección de memoria donde está el valor que se quiere cargar en el acumulador.
- Indirecto: La dirección del operando está en otra dirección de memoria.
Ej: LOAD [[x]] → trata a x como la dirección de memoria donde está el operando.
- Relativo: La dirección del operando se calcula sumando un desplazamiento al PC.



Universidad Nacional de la Patagonia San Juan Bosco
Facultad de Ingeniería. Sede Puerto Madryn
Arquitectura de Computadoras

Direccionamiento Absoluto vs. Direccionamiento Directo

El direccionamiento absoluto y el direccionamiento directo suelen considerarse equivalentes, ya que en ambos casos la instrucción contiene explícitamente la dirección del operando en memoria. Sin embargo, en algunos contextos, se hace una distinción:

- **Direccionamiento Absoluto:** Se refiere a cuando la dirección proporcionada es una dirección real en memoria, sin modificaciones ni cálculos adicionales.
- **Direccionamiento Directo:** Es un caso específico del direccionamiento absoluto donde la dirección del operando está directamente en la instrucción, sin ningún tipo de interpretación adicional.

En este modo, el operando es una dirección de memoria fija, es decir, la instrucción contiene directamente la dirección del dato en memoria. Esto significa que durante la ejecución de la instrucción, se realiza la siguiente operación:

- **MAR \leftarrow AP** (donde AP contiene la dirección absoluta del operando en memoria)

Luego, se accede a la memoria para obtener el valor almacenado en esa dirección. Este modo es útil para trabajar con datos que se encuentran en ubicaciones fijas dentro del programa.

Ejemplo de Micromemoria

Microinstrucción	Descripción
MAR \leftarrow PC	Cargar la dirección en el registro MAR
MDR \leftarrow Mem[MAR]	Leer memoria en MDR
IR \leftarrow MDR	Cargar la instrucción en el IR
PC \leftarrow PC + 1	Incrementar PC
Oper1 \leftarrow ACC	Cargar el acumulador en Oper1
Oper2 \leftarrow MDR	Cargar el operando en Oper2
ACC \leftarrow Oper1 + Oper2	Realizar la operación
Goto 0	Volver a la búsqueda de la siguiente instrucción