



Universidad Tecnológica de Panamá

Centro Regional de Panamá Oeste

Facultad de ingeniería de Sistemas computacionales

Lic. en Ingeniería de Sistemas y Computación

Organización y Arquitectura de Computadoras

Documento de la presentación magistral

Funcionamiento de la Unidad de Control

Facilitador: Prof. Bolivar Quijada

Estudiantes:

Joy Nelaton, 8-902-1282

Josue Pérez, 8-987-200

Grupo: 9IL-131

Fecha de entrega: 19/10/2023

Índice

Contenido	
Introducción.....	3
2.2 FUNCIONAMIENTO DE LA UNIDAD DE CONTROL	4
2.2.1 MICRO-OPERACIONES DEL CICLO DE INSTRUCCIÓN	4
2.2.1.1 Ciclo de Captación:.....	4
2.2.1.2 Ciclo Indirecto:	5
2.2.1.3 Ciclo de Interrupción:	6
2.2.1.4 Ciclo de Ejecución:	7
2.2.2. CONTROLES DEL PROCESADOR	8
2.2.2.1 Requisitos funcionales	8
2.2.2.2 Señales de control	9
2.2.3. Segmentación de instrucciones	11
2.2.3.1 Estrategia.....	12
2.2.3.2 Tratamiento de saltos.....	13

Introducción

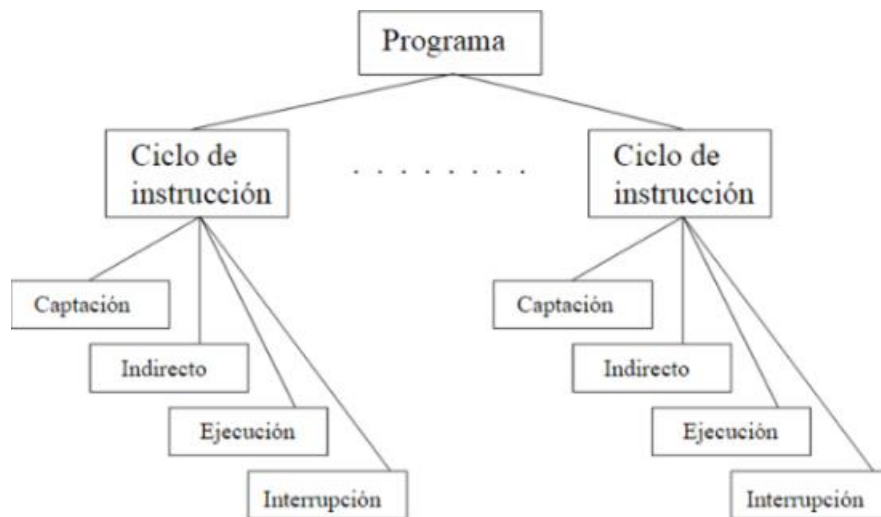
La unidad de control es un componente esencial en la arquitectura de una computadora o un microprocesador, encargada de coordinar y controlar el funcionamiento de todas las operaciones del sistema. Su principal tarea es ejecutar y supervisar las instrucciones del programa almacenadas en la memoria, asegurando que se lleven a cabo en el orden correcto y de acuerdo con las necesidades del software en ejecución.

La unidad de control trabaja en estrecha colaboración con la unidad aritmético-lógica (ALU) y otros componentes de la CPU para realizar operaciones como decodificar instrucciones, gestionar registros, controlar el flujo de datos, y administrar la interacción con la memoria y las unidades de entrada/salida. En resumen, actúa como el "cerebro" de la computadora, tomando decisiones lógicas y coordinando todas las operaciones necesarias para que la máquina funcione correctamente.

La eficiencia y precisión de la unidad de control son esenciales para el rendimiento general de un sistema informático, y su diseño puede variar dependiendo de la arquitectura del procesador y la complejidad de las tareas que debe realizar. La unidad de control es una pieza clave en la ejecución de programas y la operación de sistemas informáticos, ya que garantiza que todas las operaciones se realicen de manera ordenada y eficaz.

2.2 FUNCIONAMIENTO DE LA UNIDAD DE CONTROL

Es la parte de la unidad central de procesamiento (CPU) que se encarga de coordinar y controlar las operaciones internas de la computadora. La CPU es el cerebro de la computadora y está compuesta por varias unidades funcionales, una de las cuales es la unidad de control.



2.2.1 MICRO-OPERACIONES DEL CICLO DE INSTRUCCIÓN

Esta lleva una serie de micro-operaciones del Ciclo de Instrucción, las cuales las dividimos de la siguiente forma:

2.2.1.1 Ciclo de Captación:

Se refiere a una de las etapas fundamentales en el funcionamiento de una unidad central de procesamiento (CPU) en una computadora. En resumen, el ciclo de captación comprende las siguientes etapas:

- **Búsqueda de Instrucción:** Durante esta fase, la CPU accede a la memoria principal o la caché de instrucciones para recuperar

la siguiente instrucción a ejecutar. Esta instrucción es identificada mediante un contador de programa.

- **Decodificación de Instrucción:** La unidad de control de la CPU interpreta y decodifica la instrucción recuperada, determinando la operación a realizar y las direcciones de memoria o registros involucrados.
- **Actualización del Contador de Programa:** Después de la decodificación, el contador de programa se actualiza para apuntar a la siguiente instrucción en secuencia, asegurando que las instrucciones se ejecuten en orden.

2.2.1.2 Ciclo Indirecto:

Se utiliza para realizar operaciones que involucran la manipulación de direcciones de memoria en lugar de datos directos. En resumen, el ciclo indirecto comprende las siguientes características:

- **Obtención de una Dirección:** En este ciclo, la CPU recupera una dirección de memoria que se almacena en un registro específico o se encuentra dentro de una instrucción. Esta dirección será utilizada para acceder a un dato o realizar una operación específica en la memoria.
- **Acceso a la Dirección:** La CPU utiliza la dirección obtenida para acceder a la memoria y recuperar o modificar el dato almacenado en esa ubicación de memoria.

- **Realización de Operaciones Indirectas:** El ciclo indirecto se utiliza cuando se necesita trabajar con direcciones almacenadas en lugar de valores directos. Puede involucrar el uso de registros de desplazamiento, registros de índice o registros de base para calcular direcciones efectivas y acceder a datos en la memoria.

2.2.1.3 Ciclo de Interrupción:

Es un proceso fundamental en la operación de una computadora que permite responder a eventos externos o situaciones especiales que requieren la atención inmediata del sistema. En resumen, el ciclo de interrupción consta de los siguientes pasos:

- **Detección de la Interrupción:** Se produce una interrupción cuando ocurre un evento externo, como una solicitud de entrada/salida, un error o una solicitud de servicio por parte de un dispositivo o programa externo. La CPU detecta esta interrupción a través de líneas de interrupción o señales específicas.
- **Almacenamiento del Estado Actual:** Antes de responder a la interrupción, la CPU almacena su estado actual, que incluye la dirección de la instrucción en curso y los registros importantes, en una estructura de datos conocida como la pila o en registros especiales.
- **Manejo de la Interrupción:** La CPU redirige su ejecución hacia un manejador de interrupciones (también llamado rutina de interrupción) específico que está diseñado para tratar la causa de la interrupción. Este manejador puede ser parte del sistema operativo o del firmware de la computadora.

- **Ejecución del Manejador:** El manejador de interrupción lleva a cabo las acciones necesarias para abordar la interrupción, que pueden incluir el procesamiento de datos de entrada/salida, la corrección de errores o la gestión de solicitudes de servicios.
- **Restauración del Estado Anterior:** Una vez que se ha manejado la interrupción, la CPU restaura el estado que se almacenó previamente en la pila o en registros especiales. Esto le permite retomar la ejecución del programa interrumpido exactamente donde se detuvo.

2.2.1.4 Ciclo de Ejecución:

Es el proceso que describe cómo se lleva a cabo la ejecución de una instrucción. En resumen, el ciclo de ejecución se compone de las siguientes etapas:

- **Captación de Instrucción:** La CPU busca y recupera la siguiente instrucción a ejecutar desde la memoria principal o la caché de instrucciones. El contador de programa se actualiza para apuntar a la siguiente instrucción.
- **Decodificación de Instrucción:** La unidad de control de la CPU interpreta y decodifica la instrucción, determinando qué operación realizar y qué datos o registros se utilizan.
- **Ejecución de Instrucción:** La operación especificada por la instrucción se realiza, ya sea una operación aritmética, lógica, de

transferencia de datos u otra acción. Los datos se procesan según las instrucciones.

- **Acceso a Memoria (si es necesario)**: Si la instrucción implica acceder a la memoria, se lleva a cabo el acceso para cargar o almacenar datos.
- **Escritura de Resultados**: Los resultados de la operación se escriben en registros o en la memoria, según sea necesario.
- **Actualización del Contador de Programa**: El contador de programa se actualiza para apuntar a la siguiente instrucción en secuencia, y el ciclo de ejecución se repite con la siguiente instrucción.

2.2.2. CONTROLES DEL PROCESADOR

Reduciendo el funcionamiento del procesador a su nivel más básico, podemos definir exactamente qué es lo que tiene que hacer, es decir, sus requisitos funcionales. La definición de estos requisitos es la base del diseño e implementación de la Unidad de Control - UC, el corazón de la CPU.

2.2.2.1 Requisitos funcionales

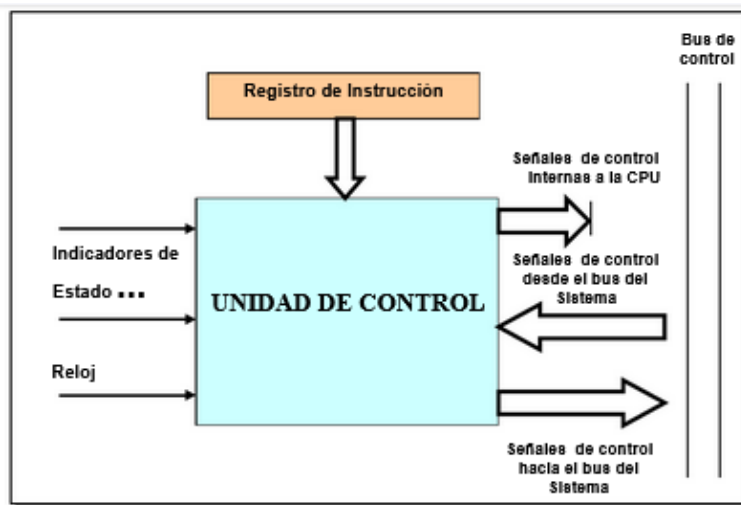
Veamos una caracterización de sus funciones a través de tres puntos:

- Controlar los elementos básicos del procesador.
- Describir las microoperaciones que ejecuta el procesador.

- Determinar las señales de control que se deben realizar para hacer que se ejecuten las microoperaciones.

2.2.2.2 Señales de control

Para que la UC realice su función, debe tener entradas que le permitan determinar el estado del sistema, y salidas que le permitan controlar el comportamiento de este. Éstas son las especificaciones externas de la UC. La siguiente figura ilustra un modelo general de la UC:



Modelo de la Unidad de Control con sus entradas/salidas

Las entradas mostradas en la ilustración son las siguientes:

- Reloj: es el encargado de “mantener la hora exacta” en el procesador. La UC hace que se ejecute una microoperación (o conjunto de microoperaciones simultáneas) en cada pulso de reloj (tiempo de ciclo del procesador o período de reloj).

- Indicadores: Son requeridos para determinar el estado del procesador y el resultado de anteriores operaciones de la ALU, para verificar si se puede continuar el proceso.
- Registro de instrucción: el código de operación de la instrucción en curso se usa para determinar qué micro operaciones hay que realizar durante el ciclo de ejecución.
- Señales de control del bus de control: La parte de control del bus del sistema suministra señales a la UC, tales como señales de interrupción y de reconocimiento.

Las salidas son las siguientes:

- ✓ Señales de control internas al procesador: son de 2 tipos:
 - Para transferir datos de un registro a otro
 - Las que activan funciones específicas de la ALU
- ✓ Señales de control hacia el bus de control: Igual, hay de 2 tipos
 - Señales de control de la memoria
 - Señales de control de módulos de E/S.
- ✓ En cuanto a las señales de control, veremos tres tipos:
 - Las que activan una función de la ALU
 - Las que activan un camino de datos y,
 - Las que son señales del bus del sistema externo u otra interfaz externa

Todas estas señales se aplican al final, como entradas binarias, directamente a puertas lógicas individuales.

La UC se mantiene al tanto de dónde está dentro de cada ciclo de instrucción. Por ejemplo, en un punto determinado, sabe cuándo se va a realizar un ciclo de captación. La UC hace esto enviando las siguientes señales de control simultáneamente:

- Una señal de control permite que el contenido de IP se transfiera a MAR y que de allí este contenido aparezca en el bus de direcciones.
- Una señal de control activa la lectura de memoria, a través del bus de control.
- Una señal de control permite almacenar el contenido del bus de datos, procedente de la memoria, en MBR.
- Se activa una señal de control, que suma “1” al contenido de IP, o dicho de otro modo, avanza al próximo desplazamiento y almacena el resultado de nuevo en IP.
- Después de esto, la UC envía una señal de control que “abre las puertas” adecuadas entre MBR e IR, almacenando la instrucción captada.

2.2.3. Segmentación de instrucciones

La segmentación de instrucciones es similar al uso de una cadena de montaje en una fábrica de

manufacturación. Una cadena de montaje saca partido del hecho de que el producto pasa a través de varias etapas de producción. Disponiendo el proceso de producción como una cadena de montaje, se puede

trabajar en los productos en varias etapas simultáneamente. A este proceso se hace referencia como segmentación de cauce (pipelining), porque como en una tubería o cauce (pipeline), en un extremo se aceptan con anterioridad nuevas entradas antes de que aparezcan como salidas en el otro extremo.

El computador se controla mediante ciclos de reloj, y la idea consiste en que en el mismo ciclo se puedan ejecutar varias instrucciones al mismo tiempo, pero que se encuentran en diferentes etapas de su ejecución. Mientras una instrucción está realizando una operación en la ALU, otra se está decodificando, y otra se está captando en memoria.

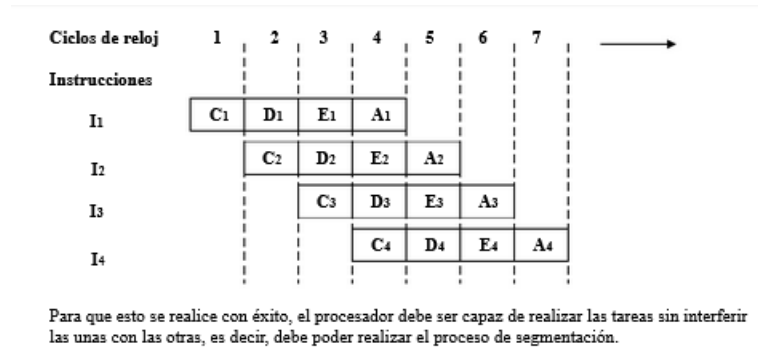
Las instrucciones se introducen al cauce en el orden del programa. Luego, en cada ciclo se van procesando las etapas de diversas instrucciones hasta concluir las todas paso a paso.

2.2.3.1 Estrategia

Para aplicar este concepto debemos recordar que claramente una instrucción presenta varios estados durante su ejecución. (diagrama de estados), pero lo resumiremos solo a los cuatro más relevantes:

- Captación (se lee la instrucción) - C
- Decodificación (se interpreta la instrucción y se captan los operandos) - D
- Ejecución (se realiza la operación) - E
- Actualización (se almacenan los resultados) - A

En este caso, la secuencia de eventos sería como se muestra a continuación:



2.2.3.2 Tratamiento de saltos

Los procesadores más utilizados en la actualidad son capaces de leer y ejecutar más de una instrucción por ciclo. Este hecho hace que los cambios en el flujo de ejecución de un programa (dependencias de control) supongan una degradación importante en el rendimiento debido a que el procesador, al encontrar un salto, ha de bloquear la búsqueda de instrucciones hasta que no sea ejecutado en su unidad funcional.

Una técnica utilizada para evitar estos bloqueos consiste en predecir el comportamiento de cada instrucción de salto (si será efectivo y su dirección destino) antes que sea resuelto. De esta manera el procesador no se bloquea y puede hacer la búsqueda de instrucciones en cada ciclo.

La predicción de saltos se puede realizar en tiempo de compilación (estática) o en tiempo de ejecución (dinámica).

✓ Entre las técnicas de predicción de saltos estática tenemos:

- Predecir todos los saltos como tomados: esta técnica es la más sencilla, pero obviamente su precisión es bastante pobre.

- Predicciones basadas en el código de operación: está basada en estudios que dicen que según el tipo de salto que se realiza, la posibilidad que sea tomado o no es diferente. Lee y Smith hacen un estudio del comportamiento de los saltos según su código de operación para el IBM 370, en él se observa que para ciertos códigos de operación la probabilidad que sean tomados es bastante alta (del orden del 80%-90%).
 - Predecir los saltos en función de su dirección: por ejemplo, los saltos “hacia atrás” predecirlos como tomados y los saltos “hacia adelante” predecirlos como no tomados. Esta técnica está basada en el hecho que una gran mayoría de los saltos “hacia atrás” corresponden a bucles, y por lo tanto serán tomados todas las veces que el bucle se ejecute menos una, en cambio los saltos hacia adelante corresponden más a estructuras if-thenelse. Esta técnica funcionará bien en programas con muchos bucles, mientras que no tendrá gran eficacia en programas con un comportamiento irregular de los saltos.
- ✓ Entre las técnicas de predicción de saltos dinámica tenemos:
- El Branch Target Buffer (BTB):
Es una pequeña memoria asociativa que guarda las direcciones de los últimos saltos ejecutados, así como su destino. A su vez guarda información que permite predecir si el salto será tomado o no.

- Predictores basados en dos niveles de historia:

Debido a la importancia de la predicción de saltos en el diseño de los computadores, ha resultado que el Branch Target Buffer basado en un nivel de predicción (contador saturado up-down

que registra la historia de cada salto) no consigue un porcentaje de aciertos suficiente para las necesidades de los procesadores.

Diversos autores han propuesto predictores más costosos a nivel de hardware que basan su predicción en dos niveles de historia: en un primer nivel recogen la historia de los n últimos saltos ejecutados o de las n últimas ejecuciones de un salto concreto (dependiendo de la implementación). Con esta información, en un segundo nivel, se indexa una tabla de patrones que en cada entrada guarda una máquina de estados (normalmente codificada en dos bits) con la que se decide si el salto es tomado o no.

- Predictores de la siguiente línea de cache:

Los predictores de la siguiente línea de cache (Next cache Line and Set) han aparecido en la literatura como alternativa al Branch Target Buffer. Un predictor NLS nos da un puntero a la cache de instrucciones indicando donde se encuentra la instrucción destino.

Conclusión

La unidad de control es un componente fundamental en la arquitectura de una computadora o microprocesador, desempeñando un papel crucial en la coordinación y control de todas las operaciones del sistema. Su capacidad para decodificar instrucciones, gestionar registros, controlar el flujo de datos y administrar la interacción con la memoria y las unidades de entrada/salida es esencial para el funcionamiento eficiente y preciso de una máquina.

La unidad de control actúa como el "cerebro" de la computadora, garantizando que todas las operaciones se realicen en el orden correcto y de acuerdo con las necesidades del software en ejecución. En resumen, su importancia radica en su capacidad para asegurar que una computadora funcione de manera coherente y eficaz, lo que la convierte en un elemento crítico en la ejecución de programas y la operación de sistemas informáticos.

Infografía

- Universidad Tecnológica de Panamá. (s. f.). *Controles del procesador*. Recuperado 22 de octubre de 2023, de <https://www.studocu.com/latam/document/universidad-tecnologica-de-panama/organiz-y-arquit-de-comput-i/controles-y-segmentacion/5496517>
- González, J., & González, A. (s. f.). *Recopilación de Técnicas de Predicción de Saltos*. Universitat Politècnica de Catalunya.

Anexo

Presentación: https://utpac-my.sharepoint.com/:p:/g/personal/joy_nelaton_utp_ac_pa/EflCXEXUDk5Pm9V7nzAwXrcBYu8-qjLa7xGtR2R3iDUWqw?e=WA806P