

ALDAVERA GALLAGA IVÁN	INSTITUTO POLITÉCNICO NACIONAL	ESCUELA SUPERIOR DE CÓMPUTO
TAREA 03 (SEGUNDO PARCIAL)	CICLOS DE EJECUCIÓN Y ETAPAS DE EJECUCIÓN DE UNA INSTRUCCIÓN	FECHA DE ENTREGA: 09 DE ABRIL DEL 2019
GRUPO (3CM2)	ARQUITECTURA DE COMPUTADORAS	

EXPLICAR EL CICLO DE INSTRUCCIÓN, LAS ETAPAS DE EJECUCIÓN DE UNA INSTRUCCIÓN

Un **ciclo de instrucción** es el período que tarda la unidad central de proceso (CPU) en ejecutar una instrucción de lenguaje máquina. Comprende una secuencia de acciones determinada que debe llevar a cabo la CPU para ejecutar cada instrucción en un programa. Cada instrucción del juego de instrucciones de una CPU, puede requerir diferente número de ciclos de instrucción para su ejecución. Un ciclo de instrucción está formado por uno o más ciclos máquina.

1. **Fetch Instruction (FI):** Se vuelca el valor del contador de programa sobre el bus de direcciones. Entonces la CPU pasa la instrucción de la memoria principal a través del bus de datos al Registro de Dirección de Memoria (MAR). A continuación el valor del MAR es colocado en el Registro de Instrucción Actual (CIR), un circuito que guarda la instrucción temporalmente de manera que pueda ser decodificada y ejecutada.
2. **Decode Instruction (DI):** El decodificador de instrucción interpreta e implementa la instrucción. El registro de instrucción (IR) mantiene la instrucción en curso mientras el contador de programa (PC, program counter) guarda la dirección de memoria de la siguiente instrucción a ser ejecutada. También se lee la dirección efectiva de la memoria principal si la instrucción tiene una dirección indirecta, y se recogen los datos requeridos de la memoria principal para ser procesados y colocados en los registros de datos.
3. **Calculate Operand (Address calculation) (CO):** Calcule la dirección efectiva de cada operando fuente. Esto puede implicar desplazamiento, registro indirecto, indirecto u otras formas de cálculos de dirección.
4. **Fetch Operand (FO):** Recupera operandos de la memoria si es necesario: si algunos de los operandos son direcciones de memoria, inicia los ciclos de lectura de la memoria para leerlos en los registros de la CPU. Si un operando está en la memoria, no en un registro, entonces la dirección de la memoria del operando se conoce como la dirección efectiva, o DE para abreviar. Por lo tanto, la obtención de un operando se puede denotar como Registro \leftarrow Memoria [DE]. En las computadoras de hoy en día, las CPU son mucho más rápidas que la memoria, por lo que la obtención de operandos generalmente requiere varios ciclos de reloj de CPU para completarse.
5. **Execute Operation (data operation) (EI) or (Execute Instruction):** La unidad de control de la CPU pasa la información decodificada como una secuencia de señales de control a las unidades de función relevantes de la CPU para realizar las acciones requeridas por la instrucción, como leer valores de registros, pasarlos a la ALU para realizar funciones matemáticas o lógicas en ellos, y escribir el resultado de nuevo en un registro. Si la ALU está involucrada, envía una señal de condición a la CU. El resultado generado por la operación se almacena en la memoria principal o se envía a un dispositivo de salida. En base a los comentarios de la ALU, la PC puede actualizarse a una dirección diferente desde la que se buscará la siguiente instrucción.

6. **WO --> Write Operand (The result is store in memory):** Almacena el resultado en la memoria: Si el destino es una dirección de memoria, inicia un ciclo de escritura en la memoria para transferir el resultado de la CPU a la memoria. Dependiendo de la situación, la CPU puede o no tener que esperar hasta que se complete esta operación. Si la siguiente instrucción no necesita acceder al chip de memoria donde se almacena el resultado, se procede a continuar con la siguiente instrucción mientras la unidad de memoria está realizando la operación de escritura.

EJEMPLO

Un ejemplo de un ciclo de instrucciones completo lo proporciona la siguiente instrucción VAX, que utiliza direcciones de memoria para los tres operandos.

mull x, y, producto

- Se obtiene el código de instrucción de la memoria [PC]
- Se decodifica la instrucción. Esto revela que es una instrucción de multiplicación, y que los operandos son las ubicaciones de memoria de X, Y y el producto.
- Se recupera X y Y de la memoria.
- Se multiplica X y Y, almacenando el resultado en un registro de CPU.
- Se guarda el resultado de la CPU en el producto de ubicación de memoria.

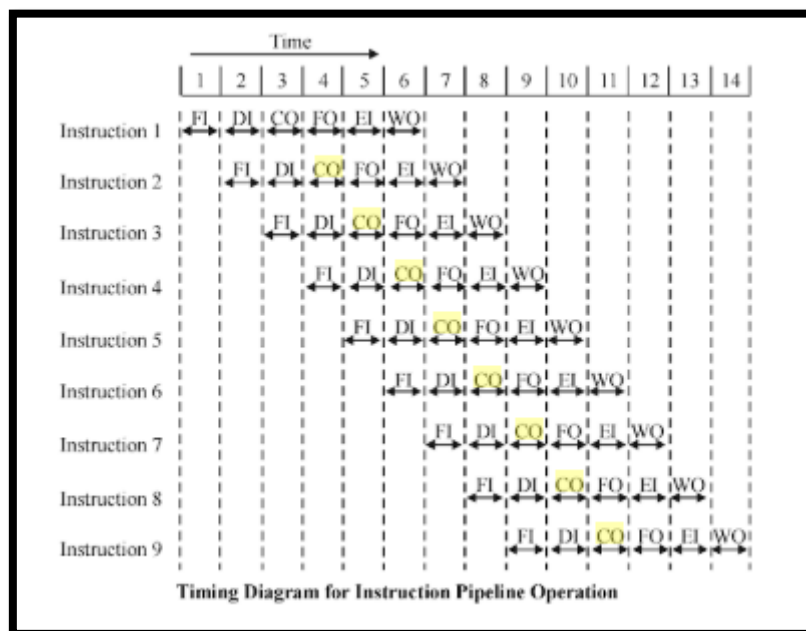


Figura 1. Diagrama de secuencia para una instrucción Pipeline.

BIBLIOGRAFÍA:

[1] Disha Expert. (2017). *Professional Knowledge for IBPS/ SBI Specialist IT Officer Exam with 10 Practice Sets - 3rd Edition*. EEUU: Disha Publications. p. 32

[2] John Y. Hsu. (2001). *Computer Architecture: Software Aspects, Coding, and Hardware*. EEUU: CRC Press. p. 85.