

Presiona **Esc** para salir pantalla completa

Arquitectura del Computador y Sistemas Operativos

Quinta Clase

Modos de direccionamiento (1/2)

Llamamos modos de direccionamiento a las distintas formas que el procesador nos da para especificar una dirección de memoria de donde sacar un dato o donde colocar un resultado.

Los más comunes son:

Inmediato: El dato es parte de la instrucción, no es necesario buscarlo en memoria.

Directo: La instrucción contiene una dirección de memoria a la que hay que acceder para obtener el dato (o donde hay que guardar el resultado).

Indirecto: La instrucción contiene una dirección de memoria a la que hay que acceder para buscar la dirección del dato.

Indexado: La instrucción contiene una dirección de memoria base. Para obtener la dirección de donde sacar el dato (o colocar el resultado) se debe sumar a esa base el contenido de un registro. Muy útil para vectores.

Modos de direccionamiento (2/2)

0x00000000 b8 04 03 02 01 mov eax,0x1020304

0x00000005 8b 1c 25 02 00 01 00 mov ebx,DWORD PTR 0x10002

0x00000010 67 8b 95 04 00 03 00 mov edx,DWORD PTR [ebp+0x30004]

0x00000017 67 8b 44 4d 00 mov eax,DWORD PTR [ebp+ecx*2+0x1]

Modos de direccionamiento (2/2)

0x00000000 b8 04 03 02 01 mov eax,0x1020304

INMEDIATO

0x00000005 8b 1c 25 02 00 01 00 mov ebx,DWORD PTR 0x10002

DIRECTO

0x00000010 67 8b 95 04 00 03 00 mov edx,DWORD PTR [ebp+0x30004]

INDEXADO

0x00000017 67 8b 44 4d 00 mov eax,DWORD PTR [ebp+ecx*2+0x1]

INDEXADO

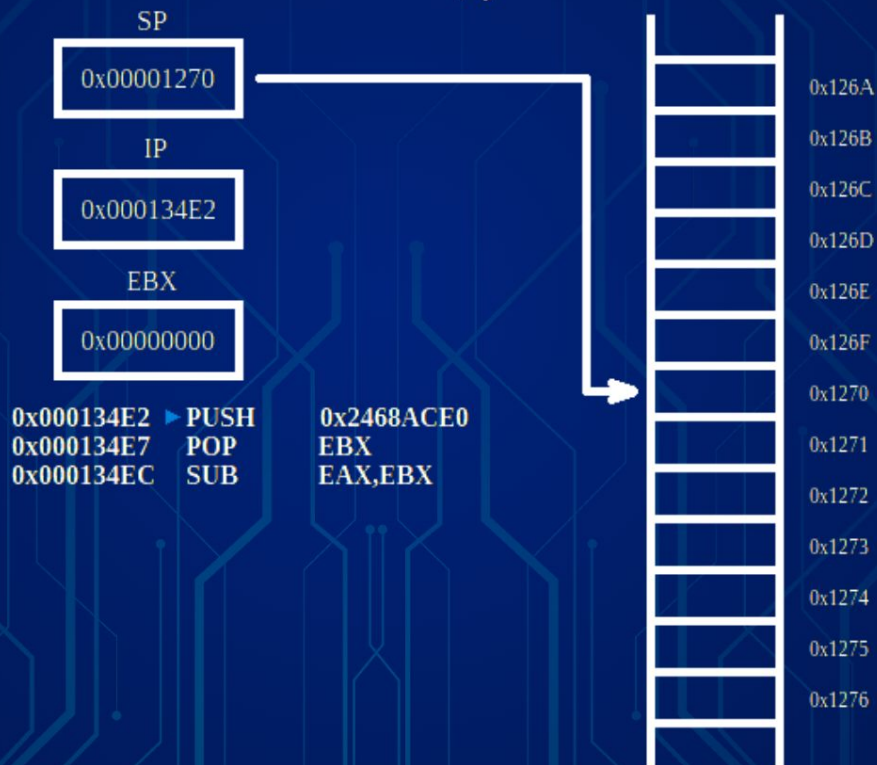
La Pila o Stack (1/2)

Al avanzar en las clases veremos que los procesadores se diseñan con el objetivo de simplificar las tareas de los programas que ejecutarán. Todo lo que el procesador haga en forma nativa (con el agregado de hardware) es *muchísimo* más rápido que si se lo implementa por software.

El primero de estos mecanismos que veremos el stack. Consiste en un registro que es usado como puntero a una zona de memoria y un conjunto de instrucciones que los usan/manipulan.

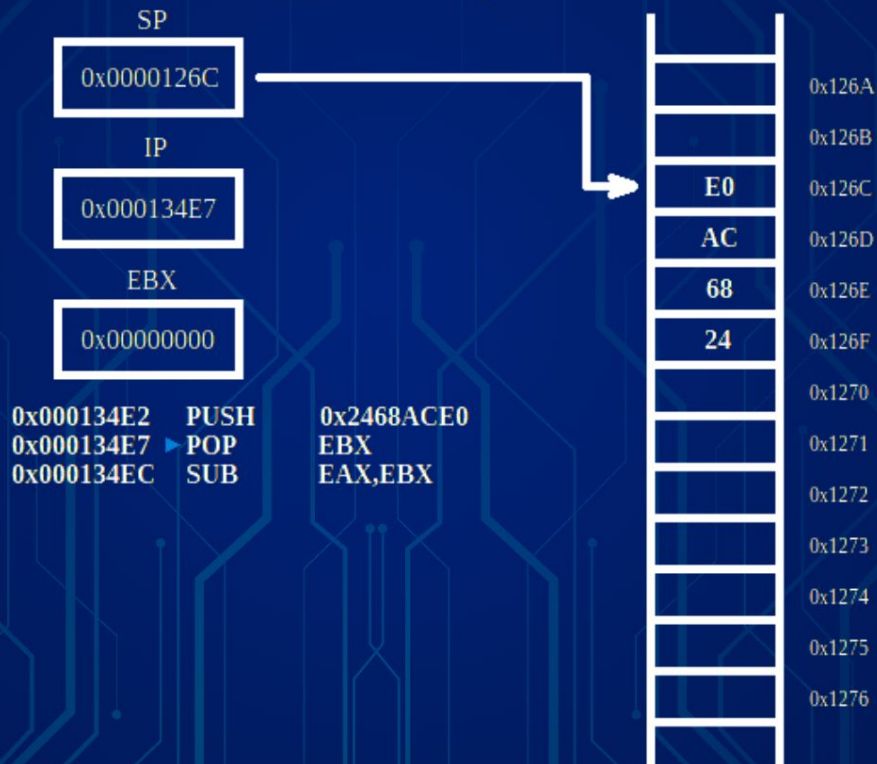
La Pila o Stack (2/2)

A continuación vemos cómo funciona, y definimos dos instrucciones, PUSH y POP.



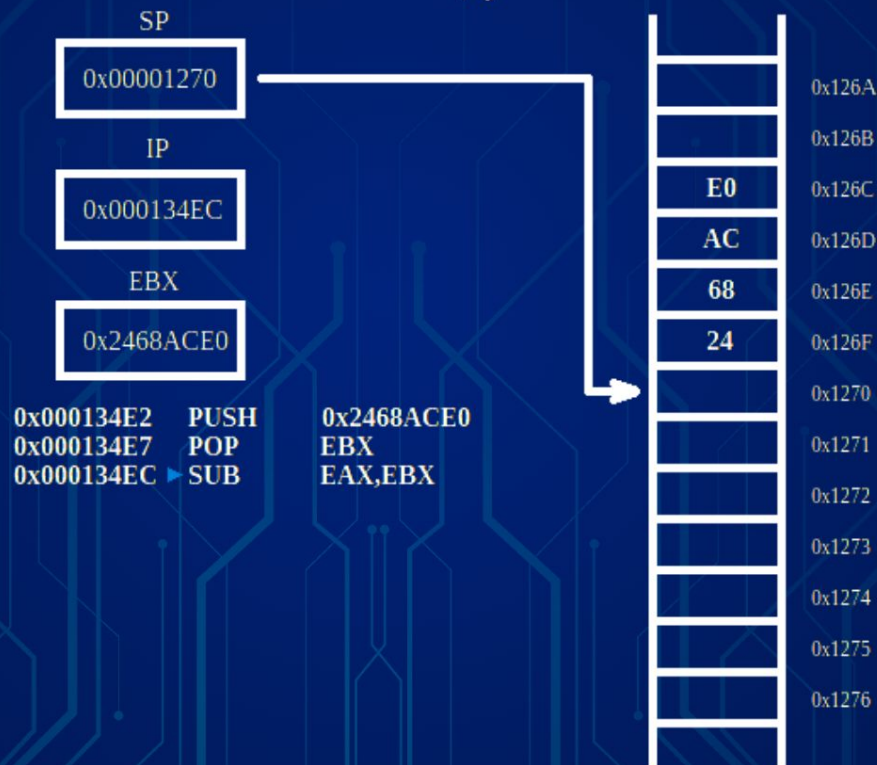
La Pila o Stack (2/2)

A continuación vemos cómo funciona, y definimos dos instrucciones, PUSH y POP.



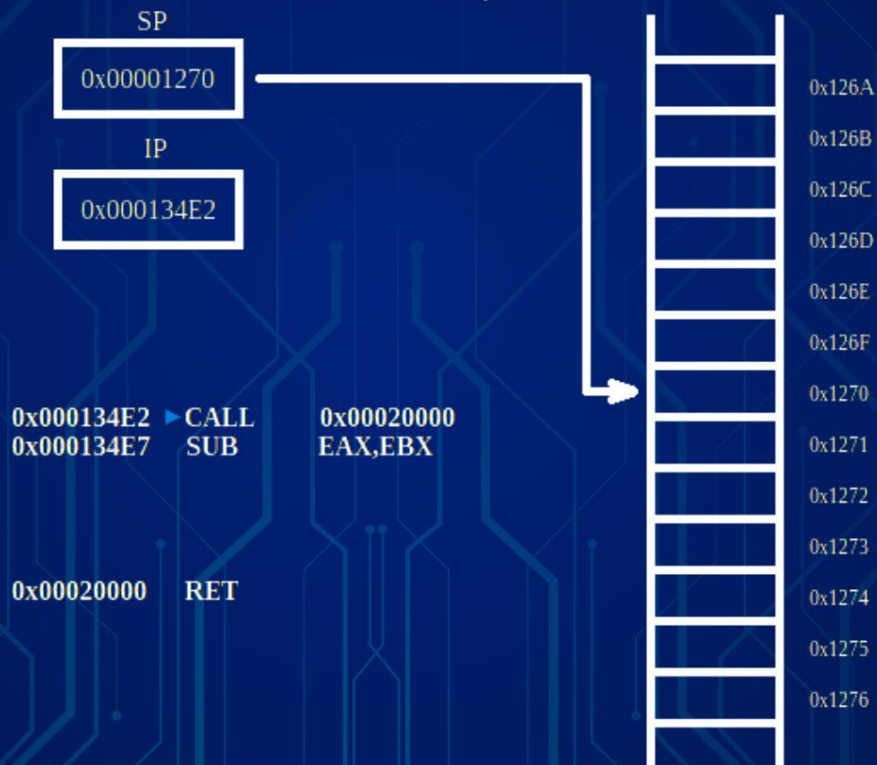
La Pila o Stack (2/2)

A continuación vemos cómo funciona, y definimos dos instrucciones, PUSH y POP.



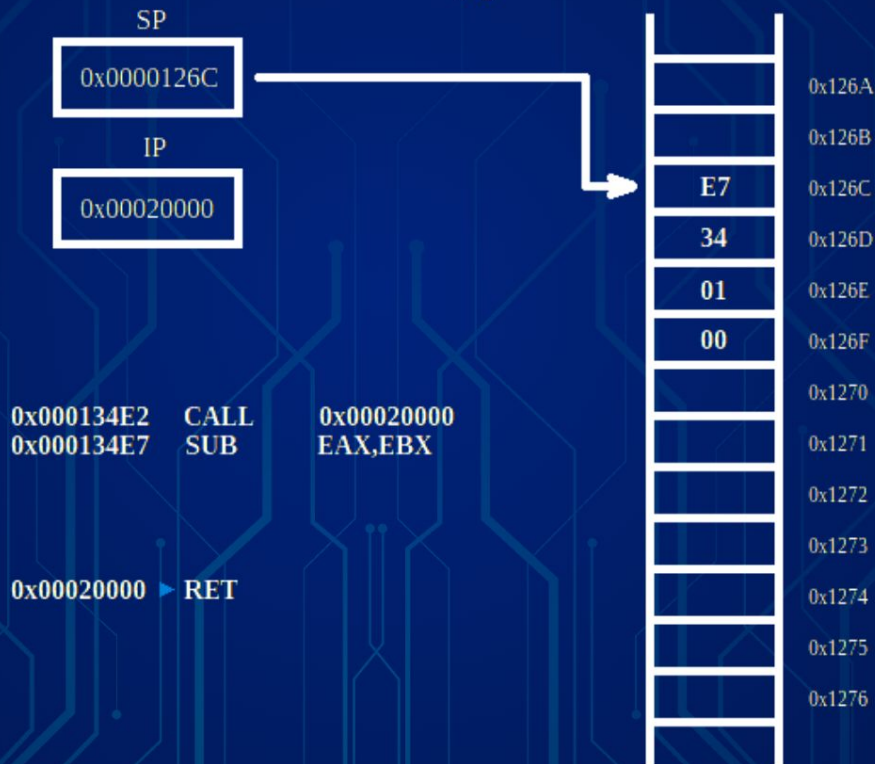
La Pila o Stack (2/2)

A continuación vemos cómo funciona, y definimos dos instrucciones, PUSH y POP.



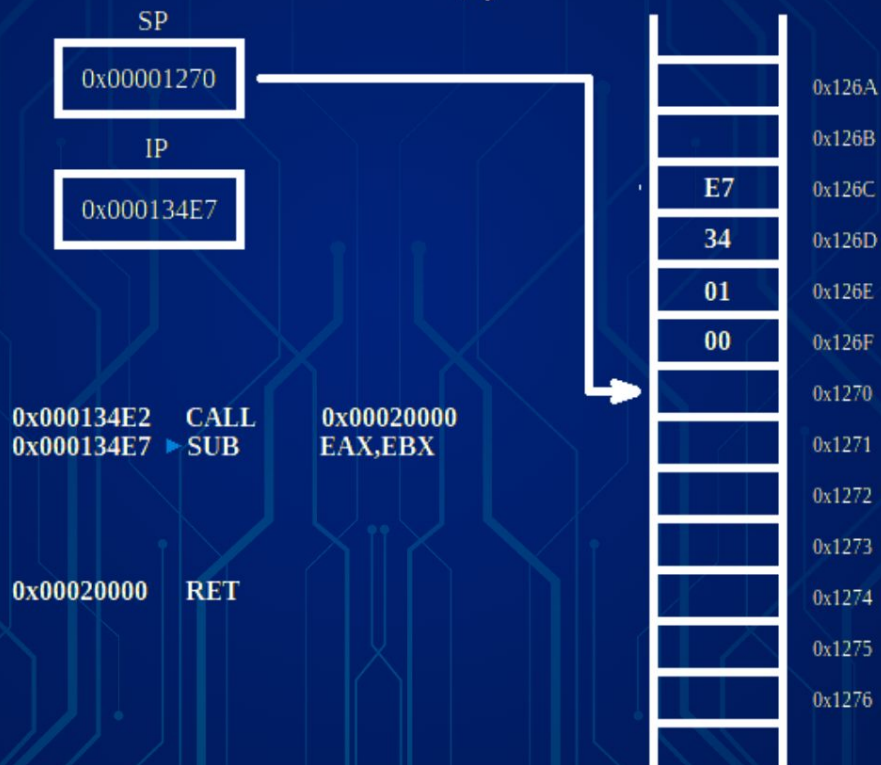
La Pila o Stack (2/2)

A continuación vemos cómo funciona, y definimos dos instrucciones, PUSH y POP.



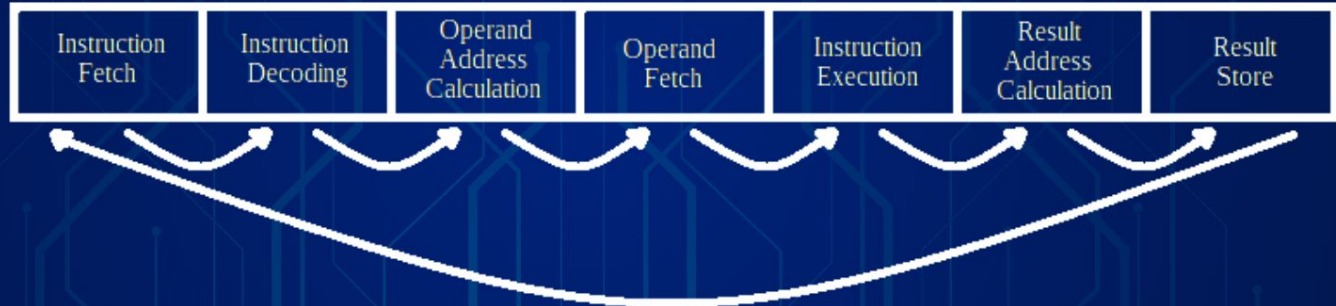
La Pila o Stack (2/2)

A continuación vemos cómo funciona, y definimos dos instrucciones, PUSH y POP.



Decodificador de Instrucciones (1/4)

La CPU que analizamos anteriormente era una versión simplificada. No necesitábamos analizar la instrucción que levantábamos porque todas las instrucciones tenían dos palabras. Sin embargo en las CPUs que encontramos en la práctica ésto no es así, y el proceso de una instrucción involucra varios pasos:



Luego de levantar la instrucción, se analiza qué operandos lleva. Además, en función del modo de direccionamiento hay que determinar de dónde se debe sacar los operandos. Una vez determinado el lugar de donde sacar los datos se los obtiene y recién ahí se puede ejecutar la instrucción. Luego hay que determinar dónde almacenar los resultados y por último guardarlos ahí.

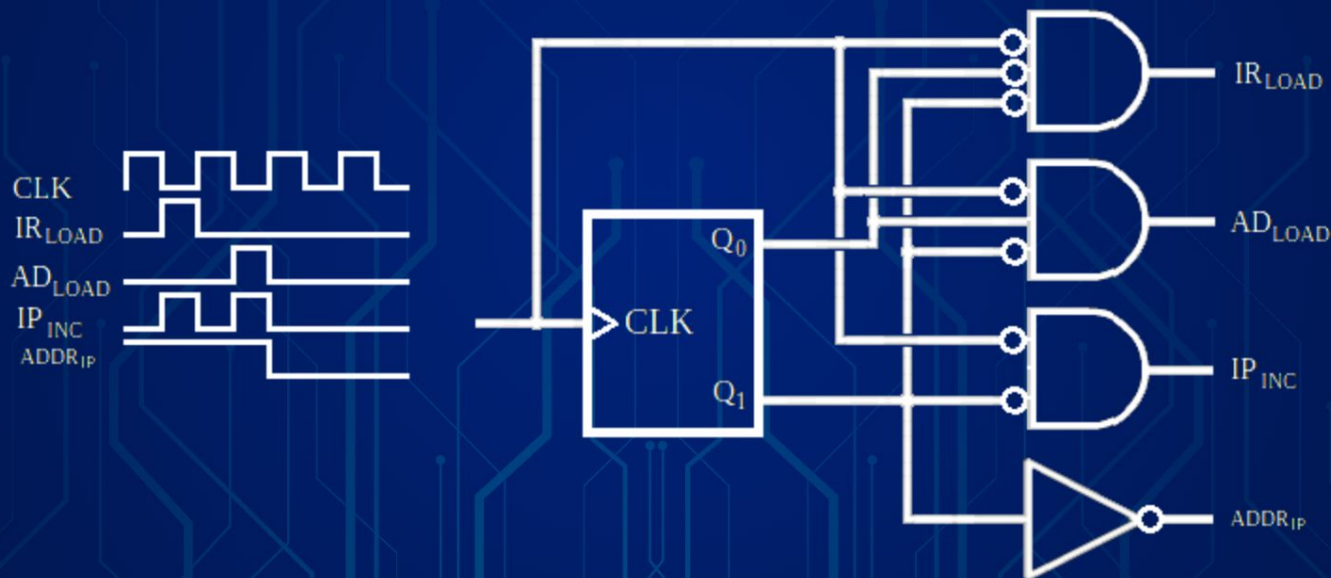
Decodificador de Instrucciones (2/4)

Como vimos al estudiar la CPU de ejemplo, la Unidad de Control es el módulo que genera todas las señales internas para ejecutar las instrucciones. El Decodificador de Instrucciones es el sub-bloque de la UC que genera la parte de esas señales que son instrucción-dependiente.

Ésto puede hacerse de dos formas diferentes. Se puede usar lógica combinacional o una memoria.

Decodificador de Instrucciones (3/4)

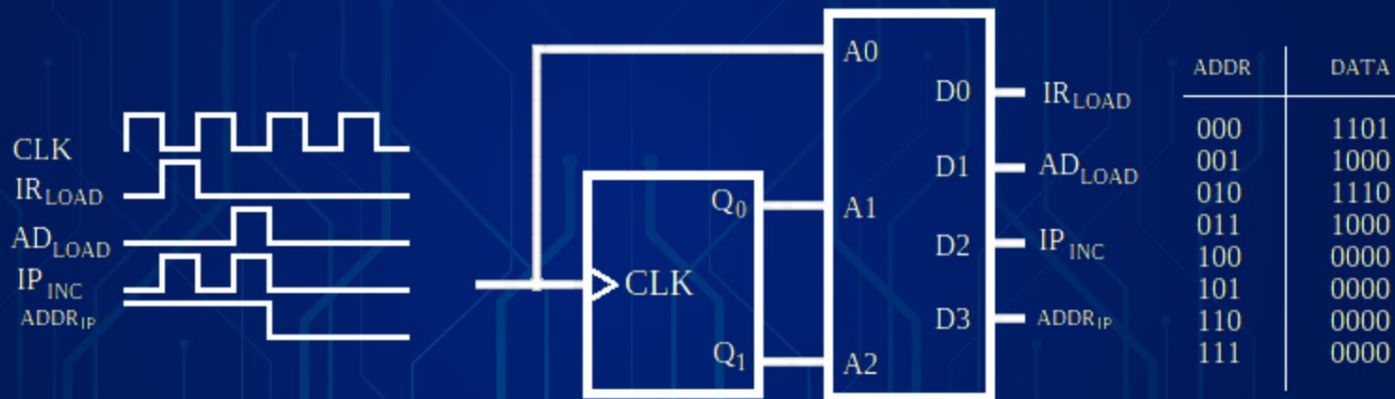
Tomemos parte de las órdenes internas que vimos en el CPU de ejemplo y veamos cómo se implementaría con lógica combinacional:



Este método es muy rápido, pero requiere muchas compuertas lógicas.

Decodificador de Instrucciones (4/4)

Ahora tomemos las mismas órdenes internas y veamos cómo se implementarían con una memoria:



Este método es más lento, pero más sencillo de implementar. El contenido de la memoria se conoce como *microcódigo*.

Ejercicio

El procesador estudiado soporta dos instrucciones de salto. El salto incondicional (Op Code 0007) y el salto si es igual (Op Code 0006). Encuentre la lógica combinacional necesaria para generar la señal LOAD del registro IP.

An abstract pattern of glowing blue lines and dots on a dark blue background, resembling a circuit board or data network, located on the left side of the slide.

Fin
¿Preguntas?

An abstract pattern of glowing blue lines and dots on a dark blue background, resembling a circuit board or data network, located on the right side of the slide.