PONTIFICIA UNIVERSIDAD CATÓLICA DEL PERÚ
FACULTAD DE CIENCIAS E INGENIERÍA
SECCIÓN INGENIERÍA INFORMÁTICA

ORGANIZACIÓN Y ARQUITECTURA DE COMPUTADORAS

Horario: **681, 682**

Grupo: 6

Profesores:

- Jimenez Garay, Gabriel Alexandro
- Romero Gutierrez, Stefano Enrique

Integrantes:

- De La Cruz Sairitupa, Juan Alberto (20161395)
- Lescano León, Carlos Alberto (20161313)
- Maldonado Alvarez, Mauricio (20180423)
- Velasquez Valenzuela, Roy Erick (20182002)

Índice

Pág.

1. Descripción de las estructuras utilizadas

Se posee 2 estructuras en el programa, la primera se llama mos6502_t contiene en 8 bits a la dirección de memoria, al stack pointer, a los registros A, X, Y y status. También contiene en 16 bits el program counter y una variable long que acumula la cantidad de ciclos totales de ejecución.

La segunda estructura se llama instrucción_t y contiene en 8 bits el número opcode y el puntero al operando. Además, contiene en 16 bits la dirección del operando y almacena en una variable short la cantidad de ciclos por instrucción.

2. Descripción de la implementación para cada una de las etapas del pipeline.

2.1. Fetch

En la etapa de fetch, se procede a leer el opcode de la instrucción a ejecutar, este dato es previamente almacenado en memoria, para luego acceder a esta posición mediante el uso del PC, una vez leído el opcode, el PC aumenta en uno para leer los demás datos necesarios para la instrucción o continuar la lectura de una nueva instrucción.

2.2. Decode

En la etapa de decode, se identifica el tipo de direccionamiento de la instrucción a ejecutarse.

En caso de ser inmediato, página 0, página 0-X o página 0-Y se almacena el valor obtenido en memoria en un puntero de 8 bits, pues para estos casos no se usará más que 1 byte de almacenamiento. Por otro lado, para los tipos de direccionamiento restantes se tuvo que utilizar un registro de de 2 bytes para almacenar la dirección de memoria a la que se quiere acceder. Una vez concluida estas operaciones se debe actualizar el PC para ir apuntando a la nueva instrucción a ejecutarse.

2.3. Execute

En esta parte se realiza el análisis de cada uno de los opcodes de las instrucciones y en base a ello se determinaba si se accedía a memoria o se realizaba el writeback, es decir, se procedía a realizar la operación requerida por la instrucción y de ser necesario se accedía en memoria o se escribía el resultado de dicha operación en algún registro. Además, en este parte, en nuestra implementación, se realizan los cambios de los flags según sean requeridos.

3. Descripción de los prototipos utilizados para desarrollar las instrucciones.

Las instrucciones almacenadas en ins son comparadas con las instrucciones en el proceso execute para identificar cual es la instrucción que se desea realizar. Una vez identificada se realiza la instrucción, y se realizan cambios y/o modificaciones en el status de flag si es necesario en la instrucción.

Para cada una de estas se utilizaron funciones auxiliares del tipo void, estas se dividen en 3 según los parámetros que poseen. Una función para las que no usan ni un tipo de parámetro (InstruccionSinDir), otra para las que usan cualquier tipo de parámetro (InstruccionConDir) y la última especial para las instrucciones del tipo salto(jump, InstruccionConSalto). Dentro de cada una de estas funciones, además, se guardaba en memoria el opcode de cada instrucción, y se ejecutaban una serie de funciones que representaban los pasos de ejecución(Fetch, Decode, Execute,Mem).

4. Descripción de los prototipos utilizados para desarrollar los modos de direccionamiento.

Para el desarrollo de cada modo de direccionamiento de una instrucción, se utilizó la estructura selectiva switch, mediante el uso de esta, se establecieron diferentes “cases” para cada direccionamiento. Dentro de cada caso, se realizaron las operaciones correspondientes para cada tipo de instrucción mediante el uso de los parámetros requeridos según el tipo de direccionamiento.

Link del repositorio: https://github.com/RoyEricvv/Grupo_6_trabajo.git

5. Anexo

Código	Nombre	Participación
20161395	De La Cruz Sairitupa, Juan Alberto	100%
20161313	Lescano León, Carlos Alberto	100%
20180423	Maldonado Alvarez, Mauricio	100%
20182002	Velasquez Valenzuela, Roy Erick	100%