

PONTIFICIA UNIVERSIDAD CATÓLICA DEL PERÚ
FACULTAD DE CIENCIAS E INGENIERÍA



MOS 6502

INTEGRANTES (Código PUCP):

Franco Chiroque	20173126
Ivan Galvan	20172361
Franco Ochoa	20171301
Lloyd Castillo	20142280
Christian Prada	20171397
Joaquín Moscoso	20163600
Javier Palacios	20162756
Paolo Galliquio	20173144
Jose Mendoza	20175706

Profesores:

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

ÍNDICE GENERAL

Introducción	3
Previo a Pipeline	3
Etapas del pipeline	3
Estructuras MOS 6502	3
Descripción de Prototipos de las Instrucciones Implementadas	3
Programa de prueba	4
Bibliografía	5

Introducción

El MOS 6502 o MOS Technology 6502 es un microprocesador de 8 bits diseñado por MOS Technology en 1975. Cuando fue introducido fue, con bastante diferencia, la CPU más barata con características completas de mercado, con alrededor de un sexto del precio o menos que las CPU con las que competía de compañías más grandes como Motorola e Intel. Era, sin embargo, más rápido que la mayoría de ellos, y, junto con el Zilog Z80, fueron la chispa de una serie de proyectos de computadores que finalmente darían lugar a la revolución del ordenador personal de finales de los 1970 y principios de los años 1980.

El siguiente proyecto consiste en la emulación del procesador mencionado a través de un programa codificado en C. Se siguió una metodología funcional y descendente donde a través de estructuras, arreglos y variables se puede simular los registros y la memoria del MOS 6502.

Para la creación del proyecto, se usaron las clases y asesorías de los profesores y asistentes del curso de Organización y Arquitectura de Computadoras del ciclo 2020-2.

Previo a Pipeline

Antes de realizar la ejecución de algún programa, el emulador asigna a una sección de la memoria las instrucciones disponibles para esta. Esto se realiza a través de arreglos asignados a funciones.

Etapas del pipeline

El emulador después de ejecutar un *reset* a sus registros, realiza un *fetch* a las instrucciones, cargando la dirección del registro *program counter* a un registro en el CPU que le permita ubicar la instrucción a ejecutar. Luego se realiza la decodificación de la instrucción encontrada. Una vez decodificada, se procede a ejecutar la instrucción o a realizar el *writeback* respectivo. Estos dos últimos pasos se realizan en una sola función. Este *bucle* se realiza hasta que el contador de programa apunte a la dirección 0x10, lo cual causaría el rompimiento de este y la finalización del programa para el procesador simulado. El simulador imprime antes y después de la ejecución del programa un reporte de los registros.

Estructuras MOS 6502

Para el caso del presente emulador MOS 6502, se pensó en definir como parte de las estructuras básicas del mismo, la creación de los registros tales como el acumulador, registro del índice X, registro de índice Y, el contador del programa (PC), el puntero a la pila (SP), el registro de banderas, el registro de instrucciones y el registro de direcciones englobados en una estructura que se maneja como el MOS 6502. Además, se utilizará también una estructura para el manejo de la memoria que representará la RAM del procesador.

Descripción de Prototipos de las Instrucciones Implementadas

Las funciones que emulan las instrucciones del procesadores comienza ubicando el registro de dirección en el primer *Opcode* de la instrucción. Luego, una vez identificada la función, se realizan las operaciones correspondientes a la instrucción. Después, se revisa si las operaciones anteriores pueden modificar los registros de bandera. Finalmente, el *program counter* salta se aumenta dependiendo el número de argumentos usados por la instrucción (1 + número de argumentos de la instrucción).

Ejm:

```
void LDX_immediate(MOS6502* cpu, MEMORIA* mem){

    // Se ubica el registro de dirección en el primer OPcode
    cpu->addr = cpu->pc+1;

    //Se realizan las operaciones correspondientes
    cpu->x = mem->ram[cpu->addr];

    //BANDERAS
    if(!cpu->x) cpu->sr|= 0x02;
    else cpu->sr &= 0xFD;
    if(cpu->x & 0x80) cpu->sr|= 0x80;
    else cpu->sr &= 0x7F;

    //El program counter se aumenta en 2 debido a que solo hay un argumento
    cpu->pc += 2;
}

void ADC_absolute (MOS6502* cpu, MEMORIA* mem){
    // Se ubica el registro de dirección en el primer OPcode
    cpu->addr = cpu->pc+1;
    //Se realizan las operaciones correspondientes
    uint16_t aux = mem->ram[mem->ram[cpu->addr+1]*256 + mem->ram[cpu->addr]] + cpu->a;
    //BANDERAS
    if(cpu->sr & 0x01) aux += 1;
    if(aux & 0x100) cpu->sr |= 0x01;          //MODIFICAR CARRY
    cpu->a = aux;
    if(cpu->a & 0x80) cpu->sr |= 0x80;        //MODIFICAR NEGATIVE
    else cpu->sr &= 0x7F;
    if(!cpu->a) cpu->sr |= 0x02;              //MODIFICAR ZERO
    else cpu->sr &= 0xFD;
    //El program counter se aumenta en 2 debido a que solo hay un argumento
    cpu->pc += 3;
}
```

Programa de prueba

En el archivo de texto "Array-len-code.txt", se encuentra un programa escrito en OPcodes. Este guarda un arreglo en memoria con una marca de fin: 0x00 (cero). Luego ejecuta un bucle "while" para contar cuántos elementos tiene el arreglo y lo guarda en un registro. Este programa fue creado para probar funcionalidades como guardar en memoria, mod ,registro, branch, etc.

Bibliografía

WIKIPEDIA

MOS 6502. Consulta: 16 de octubre de 2020.

https://es.wikipedia.org/wiki/MOS_6502

SOSA, Stefano

2020 Apuntes sobre el 6502. Lima.

CARRI, Albertina

2020 Los rubios [videgrabación]. Nueva York: Vimeo. Consulta: 26 de noviembre de 2013.

<http://vimeo.com/44770680>

MASSWERK

6502 Instruction Set. Consulta: 16 de octubre de 2020.

https://www.masswerk.at/6502/6502_instruction_set.html#LDA

Porcentaje	Franco Chiroque	Ivan Galvan	Javier Palacios	Christian Prada	Franco Ochoa	Paolo Galliquio	Lloyd Castillo	José Mendoza	Joaquín Moscoso
Franco Chiroque	100	100	100	100	100	100	100	100	100
Ivan Galvan	100	100	100	100	100	100	100	100	100
Javier Palacios	100	100	100	100	100	100	100	100	100
Christian Prada	100	100	100	100	100	100	100	100	100
Franco Ochoa	100	100	100	100	100	100	100	100	100
Paolo Galliquio	100	100	100	100	100	100	100	100	100
Lloyd Castillo	100	100	100	100	100	100	100	100	100
José Mendoza	100	100	100	100	100	100	100	100	100
Joaquín Moscoso	100	100	100	100	100	100	100	100	100

Columna representa puntaje del evaluado

Fila representa puntaje del evaluador