

Simulador de Computador

Facultad de Ingeniería, Universidad de Cuenca
ORGANIZACIÓN Y ARQUITECTURA DE COMPUTADORES
Christian Collaguazo Malla, Freddy Abad León, Bryan Aguilar Yaguana
Cuenca, Ecuador

christian.collaguazo, freddy.abadl, bryan.aguilar}@ucuenca.edu.ec

Abstract — La invención del computador abrió un mundo de desarrollo, donde cada año se presentan nuevas computadoras más potentes, menos costosas y con más características. Si bien, el lenguaje máquina e incluso la programación en nivel ensamblador es complejo y tedioso, el desarrollo y uso de un simulador permite entender este importante proceso de cómputo. El presente informe, busca sintetizar el proceso llevado a cabo por un computador al recibir operaciones básicas. La comprensión de estos temas son esenciales conocer las bases de la computación cuando uno se dedica a la programación, y así llegar a intuir cómo afecta nuestros programas a la interacción del sistema.

Index Terms — ARM, MICROPROCESSOR, INTEL

I. INTRODUCCIÓN

El modelo de von Neumann o arquitectura Princeton, “es una arquitectura de computadoras basada en la descrita en 1945 por el matemático y físico John von Neumann” el cual fue desarrollada con el fin de emplearla en el EDVAC.

Este modelo describe una arquitectura de diseño para un computador digital electrónico con partes que constan de una UNIDAD DE PROCESAMIENTO que contiene una Unidad Aritmético Lógica y Registros Del Procesador, una Unidad De Control que contiene un Registro De Instrucciones y un Contador de Programa, una Memoria para almacenar tanto datos como instrucciones, Almacenamiento Masivo Externo, y Mecanismos de I/O.

Si bien, los ordenadores digitales actuales se basan en la arquitectura Harvard, su similitud con la arquitectura von Neumann es marcada. Teniendo así que un ordenador digital de programa almacenado mantiene sus instrucciones de programa, así como sus datos, en una memoria de acceso aleatorio (RAM) de lectura-escritura. En la gran mayoría de las computadoras modernas, se utiliza la misma memoria tanto para datos como para instrucciones de programa, y la distinción entre von Neumann vs. Harvard se aplica a la arquitectura de memoria caché, pero no a la memoria principal.

II. MARCO TEORICO

Von Neumann describió el fundamento de todo ordenador electrónico con programas almacenados. Describe cómo puede funcionar un ordenador con sus unidades conectadas permanentemente y su funcionamiento está coordinado desde la unidad de control (a efectos prácticos es la CPU). El modelo de von Neumann o arquitectura Princeton, “es una arquitectura de computadoras basada en la descrita en 1945 por el matemático y físico John von Neumann” el cual fue desarrollada con el fin de emplearla en el EDVAC.

Este modelo describe una arquitectura de diseño para un computador digital electrónico con partes que constan de una UNIDAD DE PROCESAMIENTO que contiene una Unidad Aritmético Lógica y Registros Del Procesador, una

Unidad De Control que contiene un Registro De Instrucciones y un Contador de Programa, una Memoria para almacenar tanto datos como instrucciones, Almacenamiento Masivo Externo, y Mecanismos de I/O.

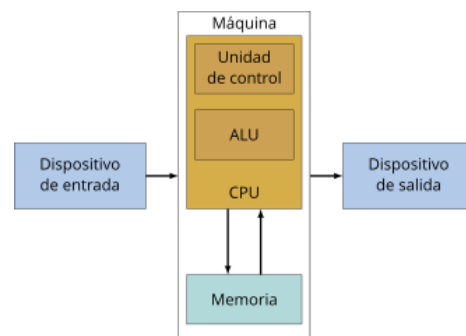


Figura 1: Arquitectura von Neumann

Unidades pertenecientes a la arquitectura de von Neumann:

Registros: se define como el lugar donde se almacenan temporalmente los datos que están en movimiento para procesarlos.

Buses: se define como las uniones entre las distintas unidades, la memoria y los periféricos.

A. UNIDAD DE PROCESO CENTRAL (CPU)

Es la unidad que controla y gobierna todo el sistema que comprende una computadora. La CPU consiste en un circuito integrado formado por millones de transistores, que está diseñado para poder procesar datos y controlar la máquina, es un factor clave para la potencia de la computadora. La CPU dispone de dos unidades en su interior: la unidad de control y la unidad aritmético-lógica.

B. UNIDAD DE CONTROL

La unidad de control se encarga de leer las instrucciones (de los programas almacenados en la memoria) y envía las órdenes a los componentes del procesador para que ejecuten las instrucciones.

El proceso empieza cuando llega una instrucción al registro de instrucciones (llega como una cadena de bits con distintas partes, referidas a la propia instrucción y a los datos que se usarán). Posteriormente el decodificador interpreta la instrucción a realizar y cómo deben de actuar los componentes del procesador para llevarla a cabo. Esta acción se realiza mediante el secuenciador que envía micro-órdenes marcadas por el reloj (que genera pulsos de forma constante, se suele expresar su velocidad en gigahercios o GHz, para los procesadores actuales).

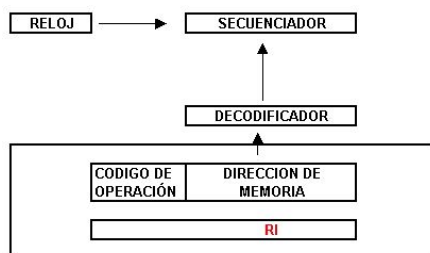


Figura 2: Unidad de proceso central

C. UNIDAD ARITMÉTICO LÓGICA (ALU)

La unidad aritmético lógica realiza todas las operaciones aritméticas y lógicas. Esta unidad puede tener distintos diseños, el de la imagen superior muestra el diseño más básico (donde el acumulador vuelve a usarse en la operación), en la actualidad lo normal es que el acumulador no se mezcle con los registros de entrada.

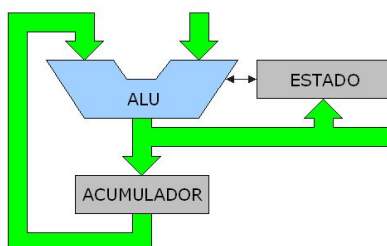


Figura 3: Unidad Aritmetico Logica

El funcionamiento comienza cuando le llega al registro de entrada un dato (una cadena de bits que representan un número), posteriormente el circuito operacional (ALU) y se procesa junto al contenido del acumulador y posteriormente se deposita de nuevo en el acumulador. Repitiendo esta acción se generan los cálculos.

D. MEMORIA PRINCIPAL

Tabla de memoria: La memoria principal en la arquitectura inicial era directamente la RAM, pero esto ha evolucionado y se han añadido memorias caché e implementado algoritmos.

Tabla de memoria	
Dir.	Contenido
0000	00000100
0001	00000101
0010	01100111
0011	01110000
0100	00000101
0101	00001011
0110	00000000
0111	00000000

Figura 4: Tabla de memoria

La memoria RAM es bastante sencilla, en comparación con la CPU, se puede decir que es una tabla, que contiene la dirección (o lugar) donde está cierto dato y el contenido del propio dato. La memoria dispone de un registro de direcciones (RDM) y un registro de intercambio de memoria (RIM o registro de datos). En el registro de direcciones se almacena la dirección en la que se almacenará o leerá un dato, y en el registro de intercambio de memoria se almacena ese dato leído o que se almacenará.

Cuando hablamos de direcciones de memoria a muchos os sonará de los "punteros" de C, y es por esto que es esencial conocer la arquitectura de Von Neumann antes de aprender a programar en ciertos lenguajes que actúan a más bajo nivel.

La unidad de control contiene el registro contador de programa, que contiene la dirección de memoria de la siguiente instrucción, que se incrementa tras realizar una instrucción y así va recorriendo la memoria y ejecutando el programa.

E. BUSES

Todos estos elementos se comunican entre si a través de buses, ya sea para manejar las acciones a realizar por la máquina o para mover datos. Hay tres tipos de buses.

El *bus de datos* permite el intercambio de datos (ya sean instrucciones o datos) con el resto de elementos de la arquitectura. Es decir, mediante el bus de datos la unidad de control recibe las instrucciones y la ALU los datos desde la memoria, al igual que también los envían por este medio.

El *bus de instrucciones* transmite las direcciones de memoria que van a ser usadas desde la CPU, para poder seleccionar los datos que serán usados.

El *bus de control* es el que transporta las órdenes generadas por la CPU para controlar los diversos procesos de la máquina.

III. PROCEDIMIENTO

El simulador del computador tiene los siguientes componentes:

- CPU: Refiriendo a la parte I., el CPU es la "unidad encargada de controlar y gobernar todo el sistema que comprende una computadora"[1].
- Memoria RAM: Asimila a una "tabla, que contiene la dirección (o lugar) donde está cierto dato y el contenido del propio dato"[2].
- Dispositivo I/O: Módulo que permite intercambiar

datos entre CPU y el usuario.

- **Compilador de Ensamblador a Lenguaje Máquina:**
"Tipo de traductor que transforma un programa entero de un lenguaje de programación (código fuente) a otro, por lo genera código máquina"[2].

OPCODES:

0001- ADD - 1
0010 - SUB - 2
0101 - LOAD - 3
0100 - STORE - 4
0110 - JMP-5

A. Herramientas

En el proceso de desarrollo se optó primero por dos desarrollos, un desarrollo en la web mediante Javascript y JQuery, sin embargo se decidió migrar a Java con interfaz gráfica Swing. Ubicando todos los datos en memoria RAM de ejecución, y no utilizar persistencia de datos en Base de Datos o manejo de archivos.

B. INTERFAZ

Al no usar conexión a base de datos ni manejo de archivos, el desarrollo fue bastante rápido. Dando énfasis a la interfaz gráfica para que resulte didáctica para el usuario.

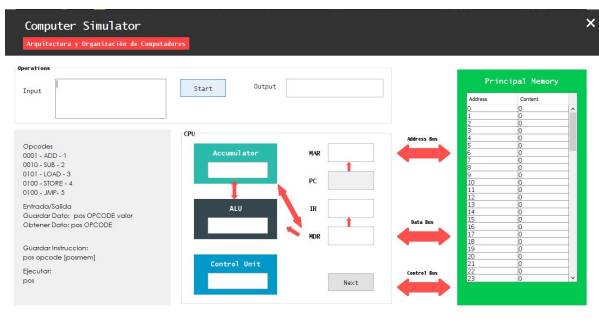


Figura 5: Interfaz simulador de computadora.

C. PRUEBAS

Los requisitos iniciales refieren a que las operaciones básicas se deban cumplir sean la suma, resta, asignación, salto condicional, lectura y escritura desde y hacia memoria, lectura y escritura desde dispositivos I/O.

Dadas estos requerimientos, se propone probar el sistema con las siguientes sentencias.

```
1 STORE 2
2 STORE 3
10 LOAD [1]
11 ADD [2]
12 JMP [14]
14 STORE [3]
15 SUB [2]
16 STORE [3]
PC : 10
```

Opcodes	
0001 - ADD - 1	
0010 - SUB - 2	
0101 - LOAD - 3	
0100 - STORE - 4	
0110 - JMP - 5	
Input/Output	
Save: pos OPCODE value	
Get : pos OPCODE	
Save instruction:	
pos opcode [posmem]	
Execue:	
pos	

Figura 6: Sentencias en el Sistema

Las sentencias de prueba como Opcodes, los resultados I/O. Para ejecutar se debe dar una posición para el program counter.

Principal Memory	
Address	Content
0	0
1	0
2	0
3	0
4	0
5	0
6	0
7	0
8	0
9	0
10	0
11	0
12	0
13	0
14	0
15	0
16	0
17	0
18	0
19	0
20	0
21	0
22	0
23	0
24	0
25	0

Figura 7: Memoria Principal

IV. CONCLUSIONES

- El desarrollo del simulador de la arquitectura de von Neumann permite ser implementado sin dificultado, ante cualquier metodología de desarrollo. Sin embargo, la cco esta abstracción, toma tiempo.

I. REFERENCIAS BIBLIOGRÁFICAS

- [1]"¿Cómo funciona la computación actual?: funcionamiento de la arquitectura de Von Neumann", Es.wikipedia.org. [Online PEDRO GUTIÉRREZ. Available: <https://www.genbeta.com/development/como-funciona-la-computacion-actual-funcionamiento-de-la-arquitectura-de-von-neumann> [Accessed: 24- Apr- 2019].
- [2]"Compilador", Es.wikipedia.org. [Online]. Available: <https://es.wikipedia.org/wiki/Compilador> [Accessed: 24- Apr- 2019].
- [3]"The Simple Computer Simulator", Es.wikipedia.org. [Online 2019]. Available: <https://www.youtube.com/watch?v=lkhwYZ0Nq3A> [Accessed: 24- Apr- 2019].