

Publicación 02

LESTER GARCÍA, ANDREA PÉREZ, ALEJANDRA SAMAYOA

Universidad Rafael Landívar

Resumen

Cundo se habla de la arquitectura del computador entran varios temas a discusión como lo son software y hardware, y estos dos términos guardan relación entre sí cuando de el desarrollo y funcionamiento de un equipo de cómputo se habla. En esta unidad se estudiaron temas de ambas áreas, desde el funcionamiento de diferentes arquitecturas de computadores actualmente utilizadas hasta el lenguaje de máquina el cual va por la parte del software. Dichos elementos funcionan de forma cooperativa, sin uno no tendría funcionamiento el otro, por esa razón es necesario tanto un buen diseño de software como de hardware para asegurar un funcionamiento eficaz y óptimo.

I. INTRODUCCIÓN

EL lenguaje de máquina es el único que entiende la computadora digital, los teléfonos y todos los dispositivos con inteligencia digital, a esto nos referimos que es su lenguaje natural. En él solo se pueden utilizar dos símbolos o caracteres, los cuales son 0's y 1's. Por ello al lenguaje de maquina también se le denomina lenguaje binario. Una dificultad añadida a los lenguajes binarios es el hecho de que son dependientes de la máquina, referente al procesador, es decir que cada procesador utiliza un lenguaje de maquina distinto para referirnos a las instrucciones que reciben los diferentes componentes, el cual está definido en su propio software.

En otras palabras, un programa escrito para una específica arquitectura o tipo de procesador no se podrá usar en otro equipo que utilice un procesador o arquitectura distinta, debido a que un programa no será transferible. Para que el programa pueda ser ejecutado en una segunda computadora se tendrá que traducir todas las instrucciones escritas en el lenguaje de máquina del primer equipo al lenguaje de la segunda computadora, de esta forma se tendrá una compatibilidad válida entre difrentes arquitecturas.

Una computadora solo puede ejecutar un grupo definido y limitado de órdenes que recibe el nombre de repertorio o set de instruc-

ciones, lenguaje maquina o código absoluto de esa computadora. Cada fabricante establece el set correspondiente a una máquina en particular, de modo que ellos constituyen una característica de diseño.

Sin embargo, existe un grupo básico de operaciones que todas las máquinas incluyen solo que con características propias por cada una de ellas.

II. LENGUAJE DE MÁQUINA: ASSEMBLER

El lenguaje ensamblador es el lenguaje de programación utilizado para escribir programas informáticos de bajo nivel, y constituye la representación más directa del código máquina específico para cada arquitectura de computadoras legible por un programador. Aun hoy se utiliza en la programación de manipuladores de dispositivos de hardware.

Consiste en un conjunto de mnemónicos que representan instrucciones básicas para los computadores, microprocesadores, microcontroladores y otros circuitos integrados programables. Implementa una representación simbólica de los códigos de máquina binarios y otras constantes necesarias para programar una arquitectura de procesador y constituye la representación más directa del código máquina específico para cada arquitectura legible por un programador.

Cada arquitectura de procesador tiene su propio lenguaje ensamblador que usualmente es definida por el fabricante del hardware, y está basada en los mnemónicos que simbolizan los pasos de procesamiento como son las instrucciones, los registros del procesador, las posiciones de memoria y otras características del lenguaje.

Un lenguaje ensamblador es por lo tanto específico de cierta arquitectura de computador física o virtual. Esto está en contraste con la mayoría de los lenguajes de programación de alto nivel, que idealmente son portables.

En la siguiente imagen se aprecia un poco de lo que es el lenguaje de Ensamblador, lenguaje de bajo nivel, único lenguaje el cual un computador entiende.

```

D ← not condition
@IF_TRUE
D;JEQ
code block 2
@END
0;JMP
(IF_TRUE)
code block 1
(END)
code block 3

```

III. ARQUITECTURA DEL COMPUTADOR

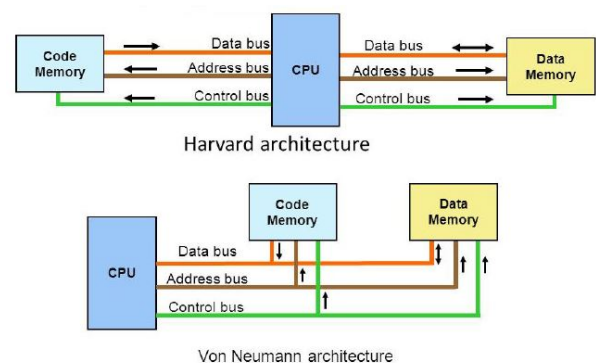
Se le da nombre de arquitectura del computador al concepto del diseño de un sistema que conforma el computador abarcando de igual manera la estructura operacional funcional de la misma, es un modelo y una descripción funcional. El diseño vela por varios requerimientos e implementaciones en distintas áreas del computador, principalmente en el CPU ya que esta trabaja de manera interna y de igual for-

ma accede a las direcciones de memoria. Dicho concepto explica como los componentes se desarrollan y cuales son las posibilidades de estos, con una configuración determinada.

Al realizar el trabajo en clase sobre las arquitecturas CISC y RISC se logró tener un panorama más amplio de lo que es una arquitectura concreta de un procesador y/o computador, y poder estudiar los orígenes de dos de las arquitecturas principales hoy en día en el mercado computacional dejó en evidencia que se compone de un proceso lógico grande el cual busca ser eficiente y crear relación software-hardware útil para la realización de múltiples procesos lógicos y secuenciales.

También se evaluaron las arquitecturas tanto de Harvard como de Von Neumann, entre los cuales existen diferencias en el direccionamiento de los buses de datos y los controles de dichos buses.

De manera gráfica las diferencias se aprecian de la siguiente manera:



La arquitectura Von Neumann fue desarrollada por John Von Neumann en conjunto con John Mauchly y John P. Eckert en el año 1945 en Pensilvania, Estados Unidos.

Este tipo de arquitectura, que se utiliza tradicionalmente en los sistemas de micro procesamiento, permite que la unidad central de proceso esté conectada a una memoria la cual en su mayoría suele ser RAM. En esta se almacenan los datos y las instrucciones de un programa. Se utiliza un acceso de control, dirección y datos con un sistema de buses único.

Cabe mencionar que a pesar de ser utilizada de manera regular esta tiene varias deficiencias tales como su limitación en la cantidad de instrucciones que esta puede ejecutar ya que por el bus de datos que utiliza se requiere que el microprocesador tenga que repetir ingresos a la memoria para poder encontrar instrucciones más complejas.

La Arquitectura de Harvard fue desarrollada por Harvard Mark I, lo que se buscaba al crear un nuevo tipo de arquitectura es que al separar las instrucciones y los datos cuando estas se guardaran en caches distintos el rendimiento fuese mejor.

Ya que los buses como anteriormente se mencionaba son distintos e independientes permiten que el CPU puede acceder de una manera simultánea a la memoria de datos o a la memoria de instrucciones. También una ventaja de estos es que pueden almacenar diferentes contenidos en la misma dirección y diferente longitud. Para optimizar el uso de la memoria general la longitud de los datos y las direcciones pueden ser distintas.

Cuando se utiliza este tipo de arquitectura se utilizan los mismos elementos que para la arquitectura de John Von Neumann a diferencia que al computador se le va a agregar dos memorias más, no una sola memoria como se utiliza regularmente.

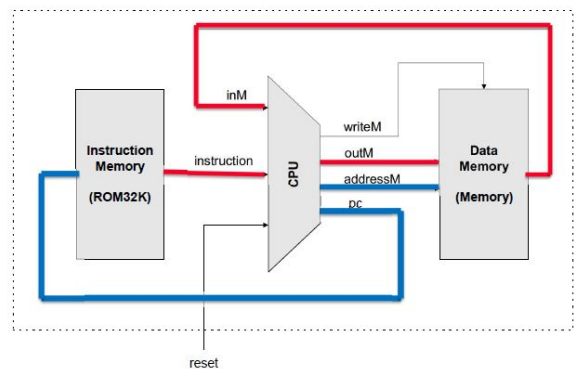
Una de las grandes ventajas que tienen esta arquitectura es que el tamaño las instrucciones no están relacionadas con el tamaño de la de los datos por ende se puede optimizar a tal manera que si alguna instrucción ocupa una posición en la memoria de programa este sea muchísimo más veloz, pero con menos longitud en él.

La mayor parte del tiempo este tipo de arquitectura es frecuentemente utilizada en DSPs que se utiliza para productos de audio como en productos de video.

Por otro lado, cuando se habla de los componentes básicos de un computador se pueden enlistar los siguientes:

- Memoria RAM
- Disco duro
- Procesador
- Dispositivos de entrada y salida
- Procesador

En la siguiente imagen se puede apreciar el diagrama de la arquitectura de una computadora Hack:



IV. DISCUSIÓN DE RESULTADOS

A partir de los resultados obtenidos se puede observar y comprobar el funcionamiento de un computador a nivel interno (Software) como externo (Hardware) y poder comparar el desempeño de diferentes arquitecturas.

A partir del trabajo realizado tanto a nivel de lenguaje de máquina en ensamblador como en el estudio y evaluación de diferentes arquitecturas existentes en el mercado actual se pudo observar que la compatibilidad es un tema controversial ya que un tipo de arquitectura no será compatible con otra, y si lo es, será en un pequeño porcentaje sin poder sacar a luz su máxima potencia y eficacia al momento de echarlo a andar.

V. CONCLUSIONES

- El lenguaje de máquina es un paquete de instrucciones el cual es únicamente interpretable por un componente micro-programable como lo es un procesador o un microprocesador.
- Cuando se habla de archivos con extensión .ASM y archivos .HACK se sabrá que de por medio existe un ensamblador el cual codificará todas las instrucciones de alto nivel a instrucciones de bajo nivel para que sean entendibles para el computador en cuestión.
- Las diferentes arquitecturas de computadoras existentes hoy en día proporcionan

una gran gama de posibilidades con las cuales entra en juego la eficiencia y eficacia de cada una de ellas, las cuales siempre buscarán ser la mejor entre todas.

- Puede llegar a ser confuso el entender el funcionamiento interno de una máquina, pero el llegarlo a dominar con el tiempo facilitará el desarrollo y manejo de futuras arquitecturas que estén en nuestras manos.

VI. TRABAJO FUTURO

A partir de lo desarrollado se seguirán construyendo y simulando diferentes componentes los cuales darán paso a la finalización de una arquitectura robusta y ordenada la cual tenga como objetivo principal el poder entender como funciona dicha arquitectura estudiándola desde el más bajo nivel de ensamblaje posible y sus respectivas operaciones y resolución de problemas de sets de instrucciones cada vez más complejos.

REFERENCIAS

- [1] Nisan, N. (2005). The elements of computing systems. Prentice-Hall of India.
- [2] <http://www.carlospes.com/minidiccionario/lenguaje>
- [3] <https://www.codifica.me/traducir-lenguaje-ensamblador>
- [4] <https://lenguajesdeprogramacion.net/assembler/>