**Informática**

**Guía Teórica**

**Ingeniería en Mecatrónica**

**Alumno: Miranda Francisco  
Legajo: 13250**

**Arquitectura de computadoras**

1. **¿Por qué las computadoras se estructuran en capas?**  
   Una computadora es una máquina que puede resolver problemas ejecutando las instrucciones que recibe de los humanos. Una secuencia de instrucciones es un programa. Dichas instrucciones permiten a los humanos comunicarse con la computadora y se basan en: Sumar dos números, verificar si un número es 0, copiar un dato de una parte de la memoria a otra. En conjunto se denominan lenguaje de máquina y es el más primitivo. Los humanos encontraron que este lenguaje era muy difícil y tedioso de usar y es por esta razón que nacen las capas. Las capas son estructuras abstractas que logran que la comunicación entre el humano y la máquina sea más organizada y mucho más cómoda logrando una excelente experiencia de usuario.
2. **Describa la relación entre capas, máquinas virtuales y lenguajes**  
   Como se dijo anteriormente, las **capas** son estructuras abstractas que logran que la comunicación entre el humano y la máquina sea más organizada y mucho más cómoda, y un conjunto de instrucciones se denomina **lenguaje**. Para lograr explicar la relación entre ellos hay que partir del lenguaje que entiende la máquina.   
   La máquina entiende un conjunto de instrucciones básico y primitivo llamado Lenguaje de máquina, los humanos encontraron que dicho lenguaje es tedioso y difícil de usar, por lo tanto crearon otros lenguajes un poco más fáciles de usar. La máquina solo entiende lenguaje de máquina, por lo tanto es necesario realizar una traducción entre el lenguaje de máquina y el lenguaje que quieren utilizar los humanos. Para organizar estos “niveles de traducción” suele resultar más fácil imaginar que existe una máquina imaginaria que sí es capaz de entender el lenguaje que crearon los humanos. Esta máquina imaginaria se denomina **máquina virtua**l y cada máquina tendrá su propio lenguaje. Entonces los humanos pueden escribir programas para la máquina virtual y esta se encargará de traducirlo para que la máquina real pueda realizar las instrucciones. Aquí podemos ver claramente dos capas, cada una con su respectivo lenguaje, pero podrían ser muchas más con distintos lenguajes, donde cada lenguaje se basa en su predecesor. De esta forma se puede pasar por muchas capas, cada una con su lenguaje y su respectiva máquina virtual, logrando la traducción entre un lenguaje complejo y cómodo para los humanos y el lenguaje simple que entiende la máquina.  
   Así, una máquina de n capas o niveles puede verse como n máquinas virtuales, cada una con cada lenguaje de máquina. A menudo se utilizan los términos “capas” y “máquina virtual” se forma indiferente. A la maquina real se la ubica en la capa 0.
3. **Describa el concepto de trayectoria de datos**  
   El concepto de trayectorias de datos surge de la conexión de un circuito denominado ALU (Unidad de aritmética lógica) con una colección de registros (generalmente de 8 a 32 registros). Gracias a esta conexión se puede realizar una operación básica fundamental que consiste en tomar uno o dos registros para que luego la ALU opere con ellos y finalmente almacenar el resultado en otro registro.  
   Concepto de Registro: Es un conjunto o grupo de memorias de un bit. Generalmente se forman en grupos de 16, 32, 64, etc.
4. **¿Qué es el microprograma y cuál es su función?**  
   Un microprograma es un conjunto de instrucciones el cual tiene la función de controlar la operación de la trayectoria de datos.
5. **Realice una tabla con las capas de una arquitectura típica, y comente brevemente la función de cada una.**

|  |  |
| --- | --- |
| **Capas** | **Función** |
| **Nivel 0: Nivel de lógica digital** | Su función es calcular una salida a partir de entradas digitales y almacenar los datos en registros. |
| **Nivel 1: Nivel de microarquitectura** | Su función básica consiste en tomar uno o dos registros especificados por el microprograma para que luego la ALU opere con ellos y finalmente almacenar el resultado en otro registro. |
| **Nivel 2: Nivel de arquitectura del conjunto de instrucciones (Nivel ISA)** | Su función consiste en interpretar un conjunto de instrucciones para que luego puedan ser ejecutadas por el microprograma o por los circuitos de ejecución en hardware. |
| **Nivel 3: Nivel de sistema operativo** | Su función es idéntica a la función del nivel 2, pero a diferencia del nivel dos, en este nivel existen un nuevo conjunto de instrucciones, diferente organización de memoria y la capacidad de para ejecutar dos o más programas al mismo tiempo. |
| **Nivel 4: Nivel de lenguaje ensamblador** | Su función principal es lograr la traducción entre el nivel 4 y los niveles anteriores, mediante un programa denominado ensamblador, ya que a diferencia de los niveles anteriores (Siendo estos niveles de lenguaje numérico), este nuevo nivel presenta un lenguaje mucho más cómodo y comprensivo para las personas. Este nivel está destinado a resolver problemas relacionados con el diseño e implementación de nuevas máquinas virtuales. |
| **Nivel 5: Nivel de lenguaje orientado hacia problemas** | Su función principal es resolver problemas. En este nivel los lenguajes están diseñados para programadores de aplicaciones que buscan resolver problemas específicos. Los programas de este nivel luego se traducen al nivel 3 o 4 mediante compiladores. Este nivel proporciona datos y operaciones que la gente conoce y puede entender fácilmente. |

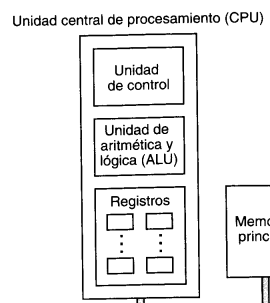
1. **¿Qué es la arquitectura de una computadora, y cuáles son los aspectos de los que se ocupa?**

La arquitectura de una computadora es el conjunto de tipos de datos, operaciones y características de cada nivel. Se ocupa de los aspectos que el usuario de un nivel puede apreciar, como la cantidad de memoria disponible. Esto incluye el estudio del diseño de las partes de un sistema de cómputo que los programadores pueden ver.

1. **¿Qué contiene la CPU?**

La CPU se compone de las siguientes partes:

* La unidad de control: Se encarga de buscar instrucciones de la memoria principal y determinar su tipo.
* La unidad de aritmética y lógica (ALU: Arithmetic Logic Unit): Realiza operaciones como suma y AND booleano necesarias para ejecutar instrucciones.
* Una memoria pequeña y de alta velocidad que sirve para almacenar resultados temporales y cierta información de control. Esta memoria está compuesta por varios registros, cada uno de los cuales tiene cierto tamaño y función, pero por lo general son todos del mismo tamaño.



1. **Mencione los 3 grandes pasos que realiza la CPU para ejecutar una instrucción, y describa brevemente cada uno**

Búsqueda: Se busca la siguiente instrucción en la memoria y se coloca en el registro de instrucciones.

Además, en caso de que la instrucción utilice una palabra de la memoria, esta se buscará y en caso de ser necesario se la colocara en el registro de la CPU.

Decodificación: Se determina el tipo de instrucción que se trajo. Identifica si es necesario colocar una palabra en el registro del CPU.

Ejecución: Se ejecutan las instrucciones correspondientes.

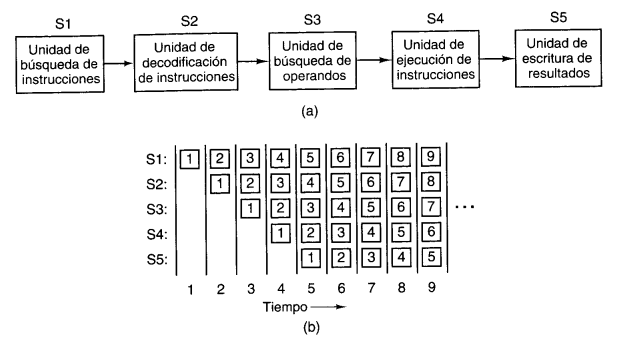
1. **¿Cuál es la ventaja de las CPU RISC sobre CISC?**

La ventaja de una arquitectura RISC frente a la CISC es la rapidez de ejecución de las instrucciones. Las instrucciones de una arquitectura RISC son más simples por no requerir la interpretación y pueden llegar a ejecutarse 10 veces más rápido que las instrucciones de una arquitectura CISC.  
Las instrucciones de la RISC son más reducidas y en consecuencia se requieren varias para realizar lo que una máquina CISC hace en una. A pesar de ello la máquina RISC presenta mayor rapidez, dado su alta velocidad de ejecución.

1. **Verdadero o Falso: Todas las instrucciones se ejecutan en 1 ciclo de CPU.**

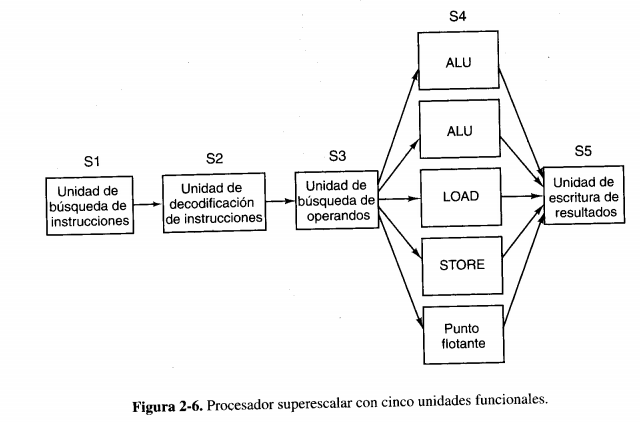
Esto es falso debido a que para ejecutar una instrucción se requieren de5 etapas: Búsqueda de instrucciones, decodificación de instrucciones, búsqueda de operando, ejecución de instrucciones y escritura de resultados y cada una de estas etapas requiere un ciclo de reloj.

Las etapas de instrucciones consecutivas se realizan en paralelo. Esto conlleva a que en cada ciclo se complete una instrucción, como muestra la imagen:



1. **¿Cuál es la diferencia entre el uso de filas de procesamiento y las arquitecturas superescalares?**

La diferencia es que la arquitectura de fila de procesamiento se basa en el principio de que si una fila de procesamiento trabaja a una determinada velocidad, tener 2 filas de procesamiento aumentaría al doble la velocidad de procesamiento. Los procesadores con arquitecturas superescalares reales en vez de usar 2 filas de procesamiento o más, lo cual implica un aumento en requerimiento de hardware, utilizan 1 sola fila de datos, pero con la idea de que la etapa S3 puede emitir instrucciones mucho más rápido que la etapa S4. Casi todas las unidades funcionales de la etapa 4 tardan mucho más que un ciclo de reloj en ejecutarse. Entonces se colocan varios ALU en la etapa 4.



1. **¿Cuál es la diferencia entre los multiprocesadores y las multicomputadoras?**

La diferencia está en que un multiprocesador es un sistema con varias CPU que comparten una memoria común. Mientras que las multicomputadoras son sistemas que consisten en un gran número de computadoras interconectadas, cada una de las cuales tiene su propia memoria, sin que haya una memoria común.

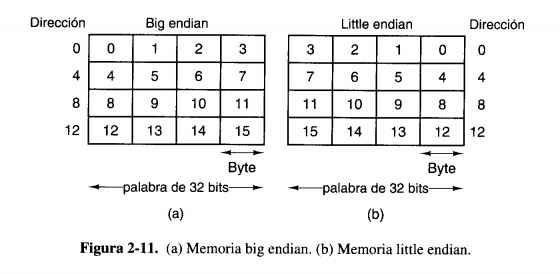
1. **Verdadero o Falso: el tamaño de la palabra de memoria es igual para todas las memorias**

Esto es Falso debido a que los bytes se agrupan en palabras. Por ejemplo: Una computadora con palabras de 32 bits tiene 4 bytes/palabra, mientras que una con palabras de 64 bits tiene 8 bytes/ palabra.

Una **palabra** es una cadena finita de [bits](https://es.wikipedia.org/wiki/Bit) que son manejados como un conjunto por la máquina.

1. **¿Qué diferencia hay entre el esquema little endian y el esquema big endian?**

La diferencia es que el esquema big endian se basa en que la numeración comienza por el extremo “grande” (de izquierda a derecha) a diferencia del little endian donde es de derecha a izquierda.



1. **¿Cuál es la función de la memoria caché?**

La función de la memoria caché es que las palabras de memoria de mayor uso o de las cuales se necesiten pronto se mantengan en la memoria caché para que cuando la CPU necesite una palabra primero la busque en la memoria caché, ya que es la memoria más rápida. Cuando la palabra no está ahí recurre a la memoria principal. Si una fracción sustancial de las palabras está en el caché, el tiempo de acceso promedio puede reducirse considerablemente.

1. **¿Cómo se conecta la CPU con la memoria principal y dispositivos de E/S?**

Para esta explicación debemos recordar que la función del CPU es ejecutar programas almacenados en la memoria principal buscando sus instrucciones y examinándolas para después ejecutarlas una tras otra.

Luego los componentes están conectados por un bus, un bus es una colección de alambres paralelos para transmitir direcciones, datos y señales de control. Los buses pueden ser externos a la CPU, cuando la conectan a la memoria y a los dispositivos de E/S, pero también pueden ser internos.

1. **¿Para qué sirve el registro denominado “Program Counter” (Contador de Programa)?**

Debemos tener en cuenta que cada registro puede contener un número, hasta algún máximo determinado por el tamaño del mismo. El registro más importante es el **contador de programa (PC,** Program Counter) que apunta a la siguiente instrucción que debe buscarse para ejecutarse. Por lo tanto el nombre “contador de programa” no tiene en sí la función de contar, pero es un término que se adoptó por ser de uso universal.

1. **Defina y enuncie las diferencias entre compilación, ensamblado e interpretación.**

La compilacióny lainterpretación son necesarias cuando se debe ejecutar un programa en una máquina y lenguaje que no se corresponde directamente con el lenguaje máquina. En este caso podemos interpretarlas como tareas de traducción, partiendo de un lenguaje de alto nivel a un lenguaje objetivo de bajo nivel. Por lo tanto, esto solo tiene sentido cuando nuestra maquina está compuesta por muchos niveles.

La compilaciónes la conversión de código de un lenguaje a otro, en un paso previo a su ejecución.   
Si pensamos en la compilación como su versión más tangible, que nos da un binario ejecutable como salida, se denomina compilación ahead-of-time (AOT).   
Si la compilación directamente pasa a su ejecución, se conoce cómo compilación just-in-time (JIT)

La forma alternativa de ejecutar un programa a partir del código en un lenguaje de programación es analizar el código y realizar los cómputos que éste indique. En este caso se dice que el programa es interpretado y estamos en el caso de una “interpretacióndel lenguaje”, el programa que permite esta alternativa se denomina intérprete del lenguaje.

Entonces se nos presenta el caso del nivel de lenguaje ensamblador que difiere de los niveles de microestructura, ISA y sistema operativo en que se implementa por traducción en lugar de interpretación. Es decir el lenguaje en el cual se escribe el programa original, lenguaje fuerte, se convierte mediante traductores en el otro, llamado lenguaje objetivo.

Una de las principales diferencias entre traducción (compilación o ensamblado)e interpretación, es que en la traducción no se ejecuta directamente el programa original en el lenguaje fuente sino que al convertirse en un programa equivalente llamado programa objeto recién se ejecuta.

Si el lenguaje fuente es en lo esencial una representación simbólica de un lenguaje numérico, el traductor se llama ensamblador y el lenguaje fuente se llama lenguaje ensamblador,y este proceso se denomina ensamblado**.**   
Si el lenguaje fuente es de alto nivel y el lenguaje objetivo es un lenguaje de máquinas numérico o una representación simbólica del lenguaje, el traductor se llama compilador**.**

Usar un lenguaje ensamblador tiene algunas propiedades, entre ellas es que desde este lenguaje se tiene acceso a todas las características e instrucciones disponibles en la máquina objetivo a diferencia de otros lenguajes.

Todo lo que se puede hacer en lenguaje de máquina puede hacerse en lenguaje ensamblador pero muchas instrucciones y registros no están disponibles para el programador en lenguajes de alto nivel.   
La desventaja en relación a los lenguajes de alto nivel es que éstos últimos si tiene el potencial de ejecutarse en muchas máquinas, a diferencia del lenguaje de ensamblador que sólo puede ejecutarse en una determinada familia de máquinas.