Detalle por qué es beneficioso generar capas de abstracción entre el lenguaje de maquina y el lenguaje humano. ¿Cómo funciona este concepto en términos de máquinas virtuales? ¿Cuántas máquinas se pueden crear sobre una máquina M0? Es beneficioso generar capas de abstracción porque trabajar a nivel lenguaje de máquina es una tarea extremadamente compleja. Por esta razón se crea una serie de abstracciones sobre ese lenguaje, con el fin de obtener un lenguaje mucho más sencillo de utilizar para las personas. El objetivo de la máquina virtual es crear dichas capas, de manera que el usuario sea capaz de trabajar con un lenguaje más fácil de utilizar, sin percatarse de lo que sucede a bajo nivel, como si la computadora realmente trabajara con ese lenguaje. Y por esta misma razón cada máquina virtual tiene su propio lenguaje. Sobre una máquina M0, se puede crear tantas máquinas virtuales como se necesite hasta llegar al nivel de abstracción deseado. De todas maneras, está limitado por el hardware que se está por utilizar Defina que son los traductores e intérpretes. Tabule sus diferencias. Entre en la siguiente página: https://survey.stackoverflow.co/2023/%23technology y realice una tabla que lista cuáles lenguajes de programación más populares son interpretados y cuáles son traducidos. Siendo L1 un lenguaje de nivel superior a L0. El traductor toma el programa escrito en L1 y lo convierte en un programa equivalente escrito en L0. El intérprete, por otro lado, es un programa escrito en L0 que toma un programa escrito en L1, examina cada instrucción y ejecuta la secuencia equivalente de instrucciones en L0. La diferencia es que el traductor crea un nuevo programa en L0, descartando el programa original en L1, y ejecuta el nuevo programa. Durante la ejecución, el programa tiene control sobre la computadora. El intérprete no requiere generar un nuevo programa y, durante la ejecución, es quien tiene el control sobre la computadora. Diagrame la máquina multinivel contemporánea, detalle cada nivel y como es la comunicación entre cada uno: Nivel 5: programación orientada a problemas – Nivel 4: Nivel de lenguaje de ensamblador – Nivel 3: Nivel de Sistema operativo – Nivel 2: Nivel de arquitectura de conjunto de instrucciones – Nivel 1: Nivel de Microarquitectura – Nivel 0: Nivel de lógica digital. 5-4: Traduccion (copilador). 4-3: Traduccion (ensamblador). 3-2: Interpretacion parcial (S.O). 2-1: interpretación o ejecucion directa. 1-0: Hardware. En el nivel cero, nivel de lógica digital, se encuentran las compuertas lógicas, las cuales se pueden combinar de manera de formar un bit de memoria, que a su vez se puede agrupar para formar registros de memoria. También se pueden combinar para formar la calculadora principal. El siguiente nivel es el nivel de microarquitectura. En este se encuentran un conjunto de registros y la Unidad Aritmética Lógica (ALU) conectados para formar un trayecto de datos. En este trayecto se pueden realizar operaciones básicas como enviar datos desde los registros a la ALU para que opere con ellos y devuelva el resultado a uno de los registros. El nivel 2 es el nivel de microarquitectura de conjunto de instrucciones (Instruction Set Architecture, ISA). Es-te nivel tiene las instrucciones que el microprograma o los circuitos de ejecución en hardware ejecutan de forma interpretativa. El nivel de Sistema Operativo suele ser referido como un nivel hibrido donde están presentes muchas de las instrucciones de nivel 2 y además un nuevo conjunto de instrucciones, diferente organización de memoria, capacidad de ejecutar más de un programa al mismo tiempo y otras características. El nivel 4, nivel de lenguaje de ensamblador, es una forma simbólica de los lenguajes anteriores, provee una manera de escribir programas para los niveles más bajos de una forma más sencilla. Los niveles más bajos son de naturaleza numérica, mientras que a partir del nivel 4 se tiene lenguajes que contienen palabras y abreviaturas con más significado para las personas. El nivel 5, programación orientada a problemas es el nivel usado por la mayoría de los programadores. Está compuesto por lenguajes diseñados para aplicación sobre problemas, como C, C++, Python, Ruby, PHP, etc. ¿Cuáles son los elementos fundamentales de una computadora? Describa brevemente el propósito de cada uno. Los elementos fundamentales de una computadora son: • Procesador: es conocido como el “cerebro” de la computadora y su función es la de ejecutar las instrucciones presentes en un programa. Otra manera de referirnos a él es como CPU (unidad de procesamiento central). • Memoria: es la porción de hardware que consiste en celdas donde en cada una de ellas se almacena parte de la información. Es el lugar donde se cargan los programas y sus respectivos datos. • Bus: es el canal de comunicación que permite la transferencia de datos entre todos los componentes de una computadora. ¿Qué diferencia hay cuando nos referimos a memoria y almacenamiento? ¿Cuáles de estos es la memoria principal y la memoria secundaria? La diferencia entre memoria y almacenamiento es que la memoria se refiere a la capacidad de un sistema para almacenar y acceder rápidamente a datos y programas mientras se está utilizando, mientras que el almacenamiento se refiere a la capacidad de un sistema para retener datos a largo plazo que no poseen conexión directa con el procesador. En esencia la memoria almacena datos y programas, se podría decir que es un tipo de almacenamiento. Pero, en términos informáticos, al hablar de almacenamiento uno se refiere a almacenamiento de disco, datos y programas que queda almacenados "permanentemente" (memoria secundaria), y al hablar de memoria uno se refiere a la memoria principal, registros o memoria caché. Explique en detalle la analogía del modelo inteligente. Podemos hacer la analogía del modelo inteligente de una computadora mediante la estructura de una fábrica: La misma se subdivide en 4 partes: • Recepción: donde se recibe la materia prima y las instrucciones de lo que tenemos que hacer (Entrada). • Depósito central: donde primeramente se almacena o archiva lo que ingresa a la recepción (Memoria). • Unidad productiva: al momento de realizar el producto, tanto la materia prima como las instrucciones se trasladan a la unidad productiva. Las instrucciones van a la mesa de instrucciones de la oficina de control y la materia prima se dirige a la unidad transformadora. La oficina de control se encarga de controlar que todo lo que se realice dentro de la unidad transformadora se lo haga de manera correcta. La unidad transformadora recibe los “pasos” a seguir para poder realizar el producto a través de la unidad de control (Procesamiento). • Expedición: el producto final puede volver al depósito central (memoria) o directamente ser enviado desde el sector de expedición hacia los diferentes destinos para ser vendido (Salida). De la misma manera podemos explicar el modelo inteligente de una computadora: • Entrada: los diferentes periféricos que posee una computadora pueden hacer de entrada de datos e instrucciones dentro de la computadora (mouse, teclado, modem, etc). El bus hace de nexo entre la entrada y la siguiente etapa de memoria. • Memoria: donde se almacena los datos e instrucciones que provienen desde la entrada. • Procesamiento: cuando sea momento de ejecutar dichas instrucciones con sus datos, éstos se trasladan a la unidad de procesamiento que posee una unidad de control que asegura que el procesamiento o “transformación” se realice de manera correcta. En la computadora la “transformación” pasa a llamarse camino de datos o unidad de ejecución. Luego de realizar el mencionado procesamiento, dicho resultado vuelve temporalmente a la unidad de memoria. • Salida: desde la memoria y a través del bus, la información ya procesada se dirige hacia algún periférico de salida para presentarla (monitor, unidad de disco, modem, parlante, etc). Mencione los elementos principales de la Arquitectura de la JVM (Java Virtual Machine) y describa brevemente cada uno. Los elementos principales de la Arquitectura de la JVM son: • Class Loader: es el encargado de cargar el código compilado a la memoria principal, linkear las clases ya cargadas y de ejecutar el método inicial de la clase principal. • Runtime Data Area: es el área donde su encuentra almacenadas: o Constantes y atributos de las clases o Código para los constructores y métodos o Objeto o dato instanciado en tiempo de ejecución o Variables locales o Llamados a métodos o Resultados parciales o Dirección de memoria de la instrucción actualo Métodos escritos en otro lenguaje diferente a Java • Execution Engine: posee el intérprete y el compilador para la ejecución precisa y rápida del código. Además, libera memoria removiendo objetos que no son referenciados dentro del Heap Area (ubicado en el Runtime Data Area). • Native Method Interface: es el puente que permite la comunicación entre código Java y llamados a código nativo del sistema. • Native Method Library: son las librerías escritas en otro lenguaje diferente de Java.

Nombre las tres categorías de bus mencionadas en clase y describa brevemente su propósito. ¿Se encuentran sólo en el procesador o también se pueden encontrar en el resto de la computadora? Hay tres categorías de bus: – Bus de datos: transmiten el dato mismo. – Bus de control: gestiona los otros bus y los dispositivos conectados a ellos. Gestiona pedidos, interrupciones, señales de sincronización y acknowledgements. – Bus de direccionamiento: transmiten la dirección adonde se va a leer o escribir el dato. Se encuentran tanto en el procesador como fuera de él, por ejemplo: para conectar el procesador con la memoria principal o para conectar dispositivos de entrada y salida. Determine si las siguientes afirmaciones son verdaderas o falsas según corresponda. Justifique su respuesta. • Los buses más largos tienden a ser más rápidos. F (Cuanto menor sea la distancia entre ambos extremos del mismo bus, más rápida probablemente será la transferencia de datos). • Los buses de expansión y locales son de alta velocidad. F (Los buses están orientados a dispositivos más antiguos que son de baja velocidad) • El bus entre procesador y memoria tiene que ser corto y de alta velocidad. V • Los buses locales van directo a memoria. F (Los buses locales son dedicados y van directo al procesador) • Los buses internos de E/S son más largos que lo de procesador-memoria. V • El bus de memoria es multipunto. F (El bus de memoria es punto a punto) • El bus de entrada/salida es multipunto. V • Tener mayor cantidad de buses aumenta el rendimiento a costo de la inversión. V En cada frase, reconozca cuáles dispositivos son primarios y cuáles son secundarios a) El procesador guarda datos en la memoria. • Primario: el procesador. Secundario: la memoria b) La memoria recibe datos del disco duro. • Primario: la memoria. Secundario: disco duro c) Para visualizar una imagen en el monitor, los datos se transfieren de memoria al procesador y después a la tarjeta de video. • Primario: el procesador. Secundario: la memoria. • Primario: el procesador. Secundario: tarjeta de video Supongamos que un diseñador de sistemas decide que, para el bus del sistema, usará solo uno de 64bits que se use tanto para datos como para direcciones. ¿Cómo se llama este tipo de bus? ¿Cuál sería la ventaja y desventaja de hacerlo así? ¿Es bueno tener un solo bus? Justifique su respuesta. Este tipo de bus es conocido como el bus del sistema (system bus), que es multiplexado. • Ventaja: menor costo • Desventaja: bajo rendimiento ya que muchos dispositivos comparten el recurso En general no es bueno tener un solo bus ya que dispositivos importantes (p.e. el procesador, la memoria), que requieren un alto rendimiento, deberían tener bus dedicado mientras que el resto de los dispositivos pueden compartir el recurso. ¿Cuál es la principal función de un dispositivo de 3 estados? La principal función de un dispositivo de 3 estados es la protección eléctrica de los periféricos/dispositivos que se encuentran conectados en un bus compartido entre ellos. Al tener el tercer estado “desconectado” se impide el ingreso de información (señal eléctrica) que el mismo no haya solicitado. Explique en detalle sobre los cuatros principales parámetros de diseño de un bus. Para mejorar el bus, ¿es tan fácil como aumentar la capacidad de uno de ellos? o ¿qué consideraciones hay que tomar cuando se busca aumentar algún parámetro en particular? Los parámetros principales en el diseño de un bus son: • Ancho de bus: se refiere a la cantidad de líneas o cables que tiene el bus y por tanto la capacidad del mismo. Para modificar este parámetro se debe tener en cuenta el espacio físico, ya que más líneas ocupan más espacio y la relación costo-beneficio de aumentar líneas ya que esta decisión incrementaría los costos. • Temporización de bus: este parámetro se refiere a la velocidad con que el bus transporta los datos. Está influenciado principalmente por el reloj. En este parámetro se debe tener en cuenta que cuanto más rápido es el bus se hace más significativa la condición de carrera de cada bit. Además, debemos tener en cuenta la compatibilidad para dispositivos más lentos. • Arbitraje de bus: Este término hace referencia a la coordinación en el uso del bus, es decir la manera en que se usará para que todos los dispositivos tengan una disponibilidad razonable del recurso. • Operaciones de bus: o Transferir una palabra o Transferir un bloque de datos: se debe indicar al bus la cantidad de palabras, el dispositivo y se transmite una palabra por ciclo de reloj. o Leer-modifica-escribir: sirve para evitar problemas de concurrencia. o Manejo de interrupciones: para evitar tiempo ocioso en el bus. Al momento de realizar cambios en los diseños de bus, se debe tener en cuenta los costos que implican en referencia a los objetivos que se alcanzan con el nuevo diseño ¿Qué es el Sesgo de bus? ¿Qué otro gran problema hay que afrontar cuando analizamos la temporización del bus? El sesgo de bus es un problema que puede ocurrir en una computadora cuando ciertos componentes están transmitiendo o recibiendo datos con mucha más frecuencia o en mayor cantidad que otros, lo que puede provocar cuellos de botella y ralentizar el rendimiento general del sistema. Otro problema es la incompatibilidad entre un bus de alta velocidad con dispositivos diseñados para buses más lento. Se tiene un sistema que sólo va a tener 10 dispositivos y se quiere asegurar que, si se me rompe un dispositivo, no afecte a los otros. ¿Cuál tipo de arbitraje conviene usar en ese sistema? Describa brevemente qué tendría y cómo funcionaría. El arbitraje más conveniente para este caso es el descentralizado de líneas independientes: Se define una cantidad limitada de líneas de solicitud, donde cada dispositivo monitorea todas, pero solo puede habilitar la suya. Cuando un dispositivo quiere usar el bus, habilita su línea de solicitud. Cuando el bus está disponible, el dispositivo con más prioridad toma control del bus. Se puede o no definir una prioridad de cada una de ellas. Tabule las diferencias entre las características del PCI y PCI Express. ¿Cuál es la máxima velocidad de transferencia a que llegó cada tipo de bus? Explique brevemente la arquitectura del PCIe.

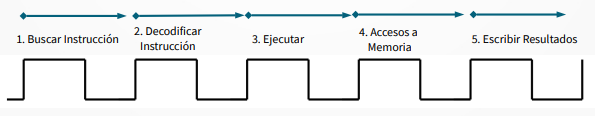
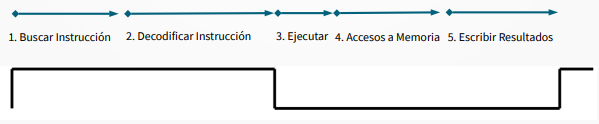
PCI- PCI Express // Velocidad máxima- 528 MB/s-Hasta 126 GB/s // Conexión-Paralela-Serie // Arquitectura-Multipunto-Punto a punto (switch) Dispositivos-Primarios y secundarios-Mismo trato (físicamente) // Modular-No-Si // Latencia-Alta-Baja // Topología-Daisy Chain-Árbol Punto a punto. Explique brevemente la arquitectura del PCIe. El bus PCIe es una mini red de conmutación de paquetes. Cada dispositivo esta conectado directamente al switch (a través de una conexión serial para envío y otra para recepción) que se encarga de conmutar los “paquetes” de información que cada uno envía o recibe. El mencionado switch es el que realiza la comunicación con la memoria y el procesador. Esta nueva forma de configuración fue lograda gracias al concepto aplicado en redes Ethernet y solucionó diversos problemas que traía consigo el bus PCI como ser limitación en la velocidad máxima y limitación para aumentar la capacidad necesaria para un dispositivo.

Si tenemos una memoria de 1.073.741.824 Bytes: a) ¿Cuántos Nibbles contiene esta memoria? Conversión de Byte a bits 1 Nibble = 4 bits Entonces: (1.073.741.824\*8/4) = 2.147.483.648 Nibbles b) ¿Cuántas palabras podemos guardar si tenemos una arquitectura de 64 bits? 134.217.728 palabras de 64 bits c) Con respecto a su direccionamiento, ¿por qué tiene sentido elegir diseñar una memoria de 1.073.741.824 Bytes y no una de un número más simple como 1.000.000.000? Para direccionamiento siempre se tiene en cuenta valores en bytes que sean potencia de 2. 1.073.741.824 bytes es potencia de 2 y es ideal para el direccionamiento. En cambio 1.000.000.000 no lo es y requiere una lógica adicional para evitar conflictos con el direccionamiento. Tenemos las siguientes posibles configuraciones: a) 1 GiB de Cache, 1GiB de Memoria Principal, 1 TiB de Memoria Secundaria b) 1 MiB de Cache, 8 GiB de Memoria Principal, 1 TiB de Memoria Secundaria c) 1 MiB de Cache, 8 GiB de Memoria Principal, 1 MiB de Memoria Secundaria d) 10 GiB de Cache, 1GiB de Memoria Principal, 1 TiB de Memoria Secundaria e) 10 GiB de Cache, 8 GiB de Memoria Principal, 1 GiB de Memoria Secundaria f) 1 MiB de Cache, 8 GiB de memoria Principal, 100 GiB de Memoria Secundaria Describa qué tecnología(s) podría usar para cada tipo de memoria y por qué. Compare los costos de cada configuración usando la tabla vista en clase. ¿Cuáles configuraciones implementaría? Justifique su respuesta. • Para Caché: SRAM: mantiene los datos sin necesidad de circuitos adicionales de refresco. • Para Principal: DRAM • Para Secundaria: dependiendo el tamaño puede ser Flash (SSD/M2) o Magnetic Disk (HDD) Basados en las respuestas anteriores las configuraciones a implementar son la B y F. A D y E son configuraciones imposibles por el alto costo en las memorias caché de gran tamaño y haciendo que las memorias principales no sirvan para nada por ser iguales o más pequeñas en tamaño que la memoria caché. C por la poca capacidad de memoria secundaria que no reduce el costo de la computadora. . Tenemos una Memoria de 128 bits. Diagrame 3 configuraciones de celdas (8, 12, y 16 bits). De 8 y de 16 se puede, serian 16 filas (iniciando desde 0) y para 16 serian 8 (iniciando desde 0) . Si mi procesador trabaja con palabras de 16 bits, ¿cuáles configuraciones funcionarían con mi arquitectura? • Las configuraciones compatibles con palabras de 16 bits son la configuración de 8 bits (Media palabra por línea) y la de 16 bits. • La configuración de 12 bits no es posible porque 128 (la memoria total) no es divisible por 12. Si decido guardar la palabra 0xA50C en dirección 0, ¿cómo sería guardada si mi sistema es little-endian? ¿Y si es big-endian? • Para el caso de la configuración de 8 bits, la palabra es más grande que el tamaño de la celda y por eso se la tiene que dividir en dos partes. Una mitad se almacenará en dirección 0 y la otra en dirección 1. Para esta configuración si se aplica el concepto de Big Endian y Little Endian ya que se debe dividir la palabra en 2 o más celdas. La palabra se divide, en este caso, en: A5 y 0C o En el caso de Big Endian A5 estará en la celda con dirección 0 de la memoria y 0C estará en la celda con dirección 1. o Para Little Endian, 0C estará en la celda con dirección 0 de la memoria y A5 estará en la celda con dirección 1. • Para el caso de la configuración de 16 bits, toda la palabra entra en una celda por lo que no se aplica el concepto de Little Endian y Big Endian al no ser necesaria la división de una palabra en 2 celdas de memoria diferentes. ¿Qué es y para qué se usa la memoria “caché”? Brevemente, ¿cómo es que funciona y qué tipo de problema resuelve? La “caché” es una memoria pequeña, rápida y cara. Se utiliza como primer lugar de búsqueda que tiene el procesador a la hora de buscar palabras en el sistema de memoria de la computadora. Es mucha más rápida y, por ende, más cara que la memoria principal (RAM) y se encuentra ubicada entre ésta y el procesador. Esto permite aumentar la velocidad en acceso a memoria. El problema fue que al pasar de los años los procesadores se volvieron mucho más rápido que el sistema de memoria. Esto causaba que el procesador tenga tiempo muerto a la espera de la memoria principal. Al ser muy serio este problema en términos de rendimiento se concibió el concepto de la caché. En el caso particular de la arquitectura Nehalem de Intel, ¿cuántos niveles de caché tiene? Describirlos brevemente incluyendo los tamaños en cada caso. La arquitectura Nehalem de Intel cuenta con dos niveles de caché privadas (L1 y L2) y un nivel de caché compartida (L3). • Caché de datos L1 de 32 KB • Caché de instrucciones L1 de 32 KB. • Caché unificada (contiene datos e instrucciones) L2 de 256 KB. • Caché unificada L3, varía de 4 a 15 MB según el modelo del procesador. En el caso particular visto en clase, el tamaño es de 8MB Con sus palabras, escriba el concepto de póliza de mapeo de caché. ¿Por qué surgió este concepto? Enumere las posibles pólizas de mapeo existentes. El concepto de póliza de mapeo de caché surgió del problema que tenía el procesador a la hora de saber si una palabra en memoria principal se encuentra en la caché o no, y si es que se encontraba, adonde. Para esto, se utiliza una parte de los bits más significativos la dirección de memoria para lógicamente dividir la memoria principal en varios grupos. Cada grupo contienen un bloque de palabras (varias palabras), y es lo que se mapea a una línea de cache. En conjunto a esto, se usa un bit adicional para indicar que la línea de cache esta válida. Las posibles pólizas de mapeo son: o Direct Mapped Cache o Fully Asociative Cache o Set Associative Cache ¿Para qué se utilizan las pólizas de reemplazo y escritura? ¿Qué ventaja y desventaja tienen la póliza write-through? La póliza (o política) de reemplazo, como su nombre lo indica, se utiliza para decidir en qué momento una línea de caché debe ser reemplazada. Uno de los objetivos de la póliza es minimizar lo que se llama Trashing (cuando un bloque es constantemente eliminado y reinsertado a la caché). Los algoritmos más populares son: • Least Recently Used (LRU) • First In, First Out (FIFO) La póliza de escritura, decide en qué momento y en qué lugar se almacenará el resultado del procesador quiere guardar en memoria. Por lo general, es probable que el resultado de un procesador sea reutilizado pronto, por lo tanto, es conveniente que se guarde en caché. Pero, tarde o temprano, se necesitará sincronizar esta nueva información a la memoria principal para evitar que existan dos versiones de un mismo dato. Para esto existen dos tipos de pólizas: • Write-through • Write-back La principal ventaja del método Write-through es que tanto el bloque de caché como la memoria principal van a tener siempre la misma información. Pero la desventaja de este método es que para garantizar que la cache y la memoria principal estén siempre coherente, cada escritura a cache tiene que ser seguida por una a memoria principal. Esto ralentizará todo el sistema porque se tienen que esperar hasta que la nueva información esté escrita en la memoria principal. ¿Qué es un registro? Es la memoria volátil (temporal) de mayor velocidad pero que sólo puede almacenar pocos datos. Se encuentran dentro del procesador y contiene la información que el mismo está utilizando actualmente. ¿Cuáles son las dos motivaciones importantes para el uso de memoria Virtual? • Una motivación importante es que la memoria virtual permite traducir la dirección del programa a una dirección física donde se pueden aplicar protecciones y seguridad para que no se accedan ciertas direcciones. • La otra motivación importante es que el espacio de memoria se puede definir más grande que el espacio en la memoria principal, permitiendo que un programador pueda crear un programa que exceda el tamaño de la memoria principal. Queda a la discreción del sistema operativo como distribuir el programa en ejecución entre memoria principal y secundaria. Explique con sus palabras el concepto de mapeo de dirección o traducción de dirección. El mapeo de dirección o traducción de dirección se utiliza para enviar a la memoria física un programa que se desea ejecutar y se encuentra en su propio espacio en la memoria virtual. Para ello, se utiliza el concepto de páginas, que son bloques de tamaño fijo, para dividir dicho programa y de esta manera facilitar la carga del programa por parte del sistema operativo. Para dicha carga solo hace falta que el S.O. encuentre una cantidad adecuada de páginas para el programa. Tengo un sistema con 4GiB de memoria principal y el sistema operativo provee 8GiB de Memoria Virtual. Si cada solapa de Chrome usa 100MiB de memoria, potencialmente, ¿cuántas solapas de Chrome podría tener abiertas? ¿Hasta cuántas funcionarían más rápida que las otras? • La memoria virtual define el espacio completo de memoria, es decir, los 8GiB definidos de memoria virtual incluye los 4 GiB de memoria principal más 4 GiB de memoria secundaria. Por lo tanto, se pueden tener abiertas hasta 80 pestañas de Chrome. • Hasta 40 pestañas de Chrome serían más rápidas que las demás ya que se pueden alojan en la memoria principal de 4 GB. El resto de las pestañas estará alojado en la memoria secundaria. Realice una tabla comparativa entre memorias tipo SRAM y DRAM. // -SRAM-DRAM// Tiempo de acceso-0.5a2.5 ns-50-70 ns//$ por GB-$500a$1000-$10a$20 //Dispositivo de almacenamiento-Transistores- 1 capacitor y 1 transistor//Tiempo de acceso a cualquier celda-Fijo y cercano al ciclo del reloj-Fijo, pero más lentas debido al refresco periódico//Bi-estables-Si-No (es monoestable) //Retención de información-Con una carga mínima-Debe ser refrescada periódicamente//Densidad-Baja-Alta

Enumere y describa brevemente los tipos de DRAM. • Fast page mode DRAM (FPM DRAM): utiliza 2 decodificadores para seleccionar fila y columna de la palabra que se requiere y solo necesita un bus para ambos. Corre de forma asíncrona al reloj principal del sistema. • Extended Data Output DRAM (EDO DRAM): mejora de la FPM DRAM que permite comenzar una nueva operación mientras todavía esta resolviendo la anterior. No reduce latencia, pero si su ancho de banda. Sigue siendo asíncrona al reloj del procesador. • Synchronous DRAM (SDRAM): memoria hibrida entre DRAM y SRAM que se encuentra sincronizada con el reloj del procesador. Al estar sincronizada, se eliminaron los circuitos de control aumentando la velocidad de transferencia entre memoria y procesador. • Double Data Rate (DDR) SDRAM: optimizanción de las SDRAM que puede realizar operación en flancos ascendentes y descendentes duplicando su velocidad. Qué es la memoria ROM? ¿Cuál es su propósito? Es un tipo de memoria no-volátil cuyo concepto es de ser de sólo lectura. Es útil tener un componente que siempre tenga los mismos datos (por ejemplo para el BIOS). Diagrame la jerarquía de memoria incluyendo todo lo visto en clase.



¿Qué significa la palabra arquitectura en el contexto de esta materia? La palabra arquitectura, se refiere a la familia de maquinas que ejecutan las mismas instrucciones (mismo ISA), con implementaciones físicas diferentes (nivel 0 y nivel 1). Camino de Datoss (Data Path):es la parte del procesador que típicamente contienen varios registros y la unidad aritmética-lógica, todo conectado a través de uno o mas buses internos al CPU. Hoy en dia, en arquitecturas mas modernas y complejas, es mas común el termino Unidad de Ejecución (Execution Unit – EU), y puede contener muchos tipos de unidades funcionales, como: unidad de punto flotante (Floating-Point Unit – FPU), unidad de generación de direcciones (Address Generation Unit – AGU), unidad de manejo de memoria (Memory Management Unit – MMU), etc. Unidad de Control (Control Unit):es responsable de, a través de señales de control, modificar el funcionamiento del camino de datos para que se puedan cumplir con las necesidades del ciclo de ejecucion, como por ejemplo, leer una instrucción de memoria. Registros: memorias temporarias de alta velocidad que pueden ser usados para contener instrucciones, datos o también información decontrol. El CPU puede tener registros de uso general y registros de uso especifico. Normalmente todos los registros del sistema tienen el mismo tamaño, pero ciertos registros de uso especifico pueden ser diferente. Unidad Aritmetica Logica – UAL (Arithmetic-Logic Unit – ALU):es una unidad funcional que puede hacer operaciones aritméticas y lógicas como: suma, resta, multiplicación, división, OR, AND, y NOT. Tiene dos entradas y una salida. Puede tener, pero no es requerido, dos registros de entrada y uno de salida para tener los datos de forma temporaria mientras completa su operación. Microprograma: software que se ejecuta a bajo nivel para poder ejecutar instrucciones a nivel ISA. ¿Cuáles son los registros de uso especifico vistos en clase? De una explicación de que hace cada uno. • MAR (Memory Address Register): contiene la dirección del dato que se quiere leer o escribir de memoria • MDR (Memory Data Register): Contiene el dato que se lee o escribe en memoria • PC: el registro mas importante del CPU. Contiene en el, la dirección de la próxima instrucción a ser ejecutada. • MBR (IR): Contiene la instrucción misma, que esta siendo ejecutada. ¿Cómo es, típicamente, el ciclo de ejecución del procesador? Detalles los pasos necesarios para completar el ciclo. Diagrame un ciclo de ejecución de la MIC-1.El ciclo de ejecución del CPU es conocido como el ciclo buscar-decodificar-ejecutar (fetch-decodeexecute cycle).Es la serie de pasos que toma el procesador para ejecutar una instrucción.1)Buscar la próxima instrucción de la memoria y colocarla en el IR.2)Modificar el PC para que apunte a la instrucción que sigue3)Decodificar la instrucción que fue buscada. Ósea, determinar que tienen que ser ejecutado 4)Si la instrucción requiere datos de memoria, determinar adonde están esos datos 5)Buscar los datos, y, si es necesario, colocarlos en algún registro del CPU 6)Ejecutar la instrucción. 7) Volver a paso 1. Explique en detalle cómo surgió el invento de la microprograma y porque eventualmente se lo eliminó. Mencione que ventajas tenia usar este tipo de unidad de control. En las primeras computadoras digitales, había solo dos niveles: le nivel ISA y el nivel de Circuito Lógico. Toda la programación se hacía a nivel ISA (un horror). Los circuitos eran complicados, difíciles de entender y construir. En los 1950s, Maurice Wilkes, un investigador de la Universidad de Cambridge, sugirió un diseño de una nueva máquina, pero de tres niveles, que simplificaría drásticamente el hardware. Esta máquina tendría un intérprete fijo, llamado el microprograma, que ejecutaría programas a nivel ISA de una manera interpretada. Esto permitiría que que el hardware sea más simple, porque solo tenías que ser diseñado para ejecutar la microprograma, con instrucciones simples y limitadas, y ya no ejecutar un programa a nivel ISA, con instrucciones más complejas y mas cantidad de ellas. Se eliminaron las microprogramas porque se hicieron muy voluminosos y con set de instrucciones muy grandes, esto hacia más lento el procesamiento. Que significa un procesador de ciclo único en comparación a un procesador de múltiples ciclos. Apoye la respuesta con un diagrama. ¿Qué tipo de procesador es la MIC-1? Como pudimos ver, el ciclo de reloj es muy importante para definir como se lleva cabo la ejecución de una instrucción. ● En términos generales, si tenemos un CPU donde el ciclo de ejecución completo de una instrucción se lleva a cabo durante un ciclo de reloj, este CPU se dice que es de Ciclo Único. En cambio, si tenemos un CPU donde cada etapa del ciclo de ejecución se lleva a cabo durante un ciclo de reloj, este se dice que es de Múltiples Ciclos (y posiblemente Segmentado). El Procesador MIC-1 es un procesador de uniciclo.



Indique que línea(s) de código pertenece a cuál de los pasos del ciclo que ejecución presentado por la lectura.

