**1. 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?**

**BUS DE DATOS:**

Cumple la función de la transferir datos de un lugar a otro.

**BUS DE DIRECCIONAMIENTO:**

Se encarga de indicar/transferir la dirección adonde se va a leer o se va a escribir el dato.

**BUS DE CONTROL:**

Se encarga de la gestión de los otros *buses* y de los dispositivos conectados a ellos, también gestiona pedidos, interrupciones, señales de sincronización y acknowledgements (reconocimiento).

No solo está presente en el procesador, también lo encontramos en otras partes de la computadora:

Para los dispositivos E/S de diferentes bradwidtch (ancho de banda) tenemos los buses internos de E/S.

Para agregar más dispositivos y en especial dispositivos viejos tenemos los buses de expansión.

Para los dispositivos que necesitan conectarse "directamente" al procesador tenemos los buces locales que poseen altas velocidades pero se los pueden conectar a una cantidad muy limitada de dispositivos similares.

Por ultimo está el bus para *transportar datos* en el camino de datos, o transportar señales desde la unidad de control, este se conoce como bus interno al procesador.

**2. Determine si las siguientes afirmaciones son verdaderas o falsas según corresponda. Justifique su respuesta.**

**• Los buses más largos tienden a ser más rápidos.**

**Falso,** hay que tener en cuenta la condición de carrera de cada bit mejor conocido como sesgo del bus, es una cuestión física que influye en la velocidad.

**• Los buses de expansión y locales son de alta velocidad.**

**Falso**, los *buses de expansión* están para conectar más dispositivos también permiten conectar dispositivos más viejos, por este motivo no se considera de alta velocidad.

**• El bus entre procesador y memoria tiene que ser corto y de alta velocidad.**

**Verdadero**, son diseñados a “medida” porque deben aprovechar al máximo la transferencia de datos o *brandwitch*.

**• Los buses locales van directo a memoria.**

**Falso**, si bien es un ejemplo común ver ejemplos de *bus local entre procesador y memoria* también hay buces locales entre el procesador y dispositivos que se pueden conectar por **PCI Express**.

**• Los buses internos de E/S son más largos que lo de procesador-memoria.**

**Verdadero**, por poner ejemplo, si tenemos un periférico de entrada como un mouse o un teclado se comunican con él procesador mediante estos buses.

**• El bus de memoria es multipunto.**

**Falso**, el *bus de memoria* es un bus **punto-a-punto** únicamente entre procesador y memoria principal.

**• El bus de entrada/salida es multipunto.**

**Verdadero**, por poner ejemplo, si tenemos un periférico de entrada como un mouse o un teclado se comunican con él procesador mediante estos buses.

**• Tener mayor cantidad de buses aumenta el rendimiento a costo de la inversión**.

**Falso**, el *bus de memoria* es un bus **punto-a-punto** únicamente entre procesador y memoria principal.

**3. En cada frase, reconozca cuáles dispositivos son primarios y cuáles son secundarios**

**• El procesador escribe datos a la memoria.**

Dispositivo **primario**: procesador. Dispositivo **secundario**: memoria.

**• La memoria recibe datos del disco duro, gestionado por el DMA.**

Dispositivo **primario**: DMA. Dispositivo **secundario:** disco duro.

**• Para visualizar una imagen en el monitor, los datos se transfieren de memoria al procesador y después a la tarjeta de video.**

**Primera operación:**

Dispositivo **primario**: memoria. Dispositivo **secundario**: procesador.

**Segunda operación:**

Dispositivo **primario**: procesador. Dispositivo **secundario**: tarjeta de video.

**4. 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?**

El bus se conoce como **bus multiplexado**.

**¿Cuál sería la ventaja y desventaja de hacerlo así?**

|  |  |
| --- | --- |
| Ventajas | Desventajas |
| nos ahorramos tener dos buses (bus  que al mismo tiempo actúa para *datos*  y para *direcciones)* | disminuye el rendimiento |
| el espacio demandado (para buses) se reduce. | nos obliga a tener que introducir un mecanismo de **arbitraje**. |

**¿Es bueno tener un solo bus? Justifique su respuesta.**

El uso de uno solo bus o varios será conveniente o no conveniente según la situación, los dispositivos y que rendimiento se busca en general, para poner un ejemplo, si tenemos dispositivos que no son de mucha prioridad o no son críticos sería un gasto extra tener un bus de cada tipo con estos dispositivos, aquí se podría emplear un bus multiplexado para abaratar costos. Otro ejemplo seria usar un bus multiplexado entre procesador y memoria, para este caso es fundamental emplear buses de cada tipo con el fin de sacar todo el provecho posible.

**5. ¿Cuál es la principal función de un dispositivo de 3 estados?**

Cumple la función principal de actuar como interfaz entre los buses internos y externos de un dispositivo, con la finalidad de producir una comunicación optima entre este dispositivo y por ejemplo el bus que lo conecta con otros dispositivos. Un ejemplo rápido que podemos dar es que las señales internas de un dispositivo son diferentes con las señales que maneja el bus externo, aquí entra en juego un decodificador que se encargará de convertir esas de señales del tipo bus interno en señales de bus externo

**6. Explique en detalle sobre los cuatros principales parámetros de diseño de un bus.**

El diseño de un bus se compone de los siguientes parámetros:

* Ancho de bus
* Temporización del bus
* Arbitraje del bus
* Operaciones del bus

**Ancho de bus**

Este parámetro se relaciona con la cantidad de memoria que podemos direccionar, su diseño se ve condicionado por el hecho (por ejemplo) de que un bus de **64bit** es mejor que uno de **32bit** pero este último es más barato de producir. De acá deducimos que un mayor ancho de bus se vuelve incompatible con dispositivos más viejos.

Como bien los dice su nombre el ancho del bus es un parámetro ya que este limita la cantidad de memoria que podemos direccionar. Podríamos suponer que con solo **“aumentar”** el ancho del bus mejoramos el rendimiento pero aparecen otras problemáticas como el incremento del costo además del espacio que supone tener más líneas de bus y por último el desarrollo siempre está condicionado a que pasa a ser incompatible con dispositivos más antiguos.

**Temporización del bus**

Cuando se tiene en cuenta este parámetro a la hora de diseñar un bus entra en juego la temporización del reloj del bus, a mayor frecuencia el periodo es más corto y la condición de carrera de cada bit es más significativo.

Este parámetro también influye en la velocidad de bus, está dominado por el reloj del bus, a medida que mayor es la frecuencia tenemos un periodo de periodo más corto se vuelve más significante la condición de carrera de cada bit conocido también como sesgo del bus

**Arbitraje del bus**

A partir de aquí empezamos a tener en cuenta la implementación del **bus multiplexado** para abaratar costos pero pasamos a tener un **recurso compartido** y esto nos lleva a tener dos *sub parámetros*:

|  |  |  |  |
| --- | --- | --- | --- |
| CENTRALIZADO | NO CENTRALIZADO | | |
| Consiste en implementar  un circuito llamado árbitro,  los dispositivos que quieren  usar el bus se conectan a este  en forma de encadenamiento  circular, el dispositivo más  cercano tiene mayor prioridad.  En algunos casos existe  una línea de acuse para  que un dispositivo  pueda informar que tomó control de bus | | En este método no implementamos un circuito de arbitraje pero se divide en dos estrategias: | |
| Primera estrategia es para una cantidad determinada de dispositivos, hay una línea de solicitud por cada dispositivo, pero cada uno de estos puede monitorear el resto de las líneas de solicitud, es opcional definir una solicitud.  Cada dispositivo que quiere tomar control del bus habilita su línea de solicitud y si el bus está disponible el dispositivo toma control del bus | La segunda estrategia es no limita la cantidad de dispositivos y tiene tres líneas de control: línea de solicitud, ocupado y línea de arbitraje. La línea de solicitud es un **OR-alambrado**, la línea de ocupado se habilita cuando un dispositivo toma control de bus, la línea de arbitraje siempre está habilitada y su propagación depende de los dispositivos conectados, el dispositivo que quiere chequea la línea de arbitraje, si está disponible habilita la línea de solicitud, niega la propagación de la línea de arbitraje, toma control y habilita nuevamente la línea de solicitud. Caso contrario, cuando no se encuentra habilitada la línea de arbitraje el dispositivo no habilita su línea de arbitraje de salida. |

**Operaciones del bus**

Este es último parámetro a la hora de diseñar el bus. Para el caso de usar un procesador de varios núcleos se busca evitar que dos de estos lean un mismo dato y por ende se corrompa. Para evitar esto existe un valor booleano que el núcleo analiza para saber si es *estructura* está en uso. En caso de no ser así (de valor 0) el núcleo cambia su valor a 1 antes de liberar el bus, esto se conoce como *leer-modificar-escribir*. Después tenemos otra operación que es el **manejo de interrupciones** donde el procesador da instrucciones a un dispositivo, mientras este dispositivo se encuentra ocupado el procesador se encarga de otras tareas esperando que este dispositivo genere una *interrupción* (lo comunica a través del bus), se utiliza un *árbitro* para manejar las interrupciones (normalmente centralizado) para manejar las interrupciones con los dispositivos críticos. Después de producida la interrupción el procesador busca mediante un identificador al dispositivo e indexa una tabla de **vectores de interrupción** (punteros) para ordenar la rutina de servicio a la interrupción.

**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?**

Si buscamos aumentar algunos de los parámetros previamente mencionados el primer problema que aparece es la compatibilidad con dispositivos más antiguos. Las consideraciones a tener en cuenta son los dispositivos que vamos a conectar, si estos son críticos.

**7. ¿Qué es el Sesgo de bus?**

**Sesgo de bus** o ***bus skew*** es como se denomina a la “incapacidad” que tiene la información de moverse *más rápido* dentro de un bus, otro lugar donde también puede influir es en la forma del bus (la longitud del mismo), el sesgo de bus es una **cuestión física**.

**¿Qué otro gran problema hay que afrontar cuando analizamos la temporización del bus?**

Otro problema es que logra un que un **bus sea más rápido** hace que este ya no sea incompatible con dispositivos más viejos.

**8. 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.**

Es conveniente emplear la primera estrategia del arbitraje descentralizado, donde se emplean la misma cantidad de líneas de solicitud que dispositivos, al mismo tiempo cada uno de los dispositivos puede saber el estado de la otra línea de solicitud pero solo puede activar la suya. Cuando un dispositivo quiere tomar control del bus habilita su línea y cuando el bus está disponible toma control de este según sea su prioridad. La prioridad de que tienen las líneas es opcional.

**9. Mencione y describa las operaciones típicas de un bus**

Una de las operaciones que maneja el bus son las interrupciones, mediante el bus un dispositivo comunica al procesador su interrupción que luego será indexada.

**10. Tabule las diferencias entre las características del PCI y PCI Express. ¿Cuál es la máxima velocidad de transferencia que llegó cada tipo de bus?**

|  |  |
| --- | --- |
| PCI | PCI Express |
| Es compatible con el puente ISA | No es compatible con el puente ISA |
| Diseño más específico para un dispositivo | Diseño más general hacia los dispositivos |
| Trasferencia de información en forma paralela | Transferencia de información en forma de paquetes |

**11. Explique brevemente la arquitectura del PCIe**

Consiste simplemente en emplear un *switch* para conectar todos los dispositivos con una conexión de par de canales unidireccionales del tipo **punto a punto**. Los dispositivos tienen un mismo trato y la forma en la envían y reciben información es con **paquetes de datos**.