**ARQUITECTURA DE COMPUTADORAS**

CAPITULO IV

**ENTRADAY SALIDA**

**AÑO 2019**

**TEMA UNIDAD DE ENTRADA - SALIDA**

**IV.1 - INTRODUCCIÓN:**

**Las unidades de entrada y de salida, que a veces se agrupan en una sola, denominada unidad de entrada/salida, canal de e/s o módulo para el intercambio de información, tiene o tienen la misión de conectar los módulos de la máquina con los dispositivos externos de la misma, denominados PERIFÉRICOS.**

**Podemos preguntarnos porqué no se conectan los periféricos directamente al bus del sistema, y así nos evitamos una serie de problemas.** **En máquinas muy pequeñas, evidentemente así se lo hace, pero los periféricos que pueden ser así conectados deben tener un alto grado de inteligencia, y no todos permiten una operación de este tipo, sino que en general, si bien poseen cierta lógica de control, son conducidos de diversas maneras, poseen diferentes longitudes de palabra y diferentes formatos de datos.**

**Por otra parte, la velocidad de los periféricos es mucho menor que la del procesador, por tanto es necesario interponer algún tipo de amortiguador (buffer) que contenga los datos el tiempo necesario para que ellos los acepten.**

**Entonces, se requiere siempre un módulo de E/S, que tiene dos funciones primordiales:**

- Interconectar la UCP y la memoria externa vía los buses del sistema o vía un conmutador central.

- Interconectar varios dispositivos periféricos mediante enlaces de datos "hechos a medida".

**Por lo tanto, es razonable estudiar un poco el funcionamiento de los periféricos, para después pasar a la estructura y funcionamiento del módulo de E/S.**

**En la figura I.23, tenemos el esquema general del módulo de E/S, que en caso de pequeñas máquinas puede convertirse en el esquema de la figura IV.1.**

**La principal diferencia entre ambos, es que en el segundo se supone que la memoria central es externa al procesador, y está conectada a los buses de la UCP mediante el módulo de E/S. En el primer caso en cambio, se supone que la memoria es interna al computador, y los módulos de entrada y de salida son los que conectan a todo el computador con el mundo externo a él, entendiendo que la memoria central forma parte integrante del mismo.**

**IV.2 - DISPOSITIVOS EXTERNOS:**

**Ya dijimos que también son denominados periféricos, y conforman toda una pléyade de equipos para entrada o salida de información, entre otros podemos citar:**

- unidades de discos rígidos

- unidades de discos flexibles

- unidades de discos ópticos

- unidades de cinta magnética

- impresoras

- escaneadores

- graficadores

- pantallas

- sensores

- actuadores

- modems

- sistemas de audio

- sistemas de video

- sistemas para interconexión a redes

**Dispositivos estos que pueden clasificarse en cinco grupos:**

- memorias de masa (discos, disquettes, cintas)

- leíbles por el hombre (pantallas, impresoras)

- leíbles por la máquina (escaneadores, sensores)

- comunicaciones (modems, sistemas para redes)

- accionamientos (actuadores, graficadores)

**En la figura IV.2, tenemos el diagrama genérico de un periférico, en el cual, los bloques indicados corresponden a:**

-Señales de Estado: indican el estado del dispositivo (LISTO, OCUPADO, DESCONECTADO)

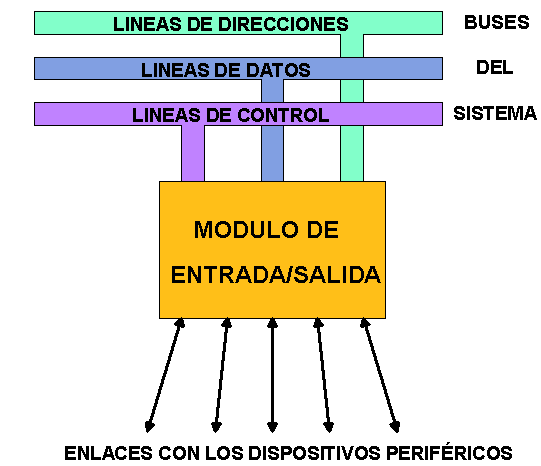
-Señales de Control: determinan la función que el periférico debe desempeñar (LEER, ESCRIBIR, TRANSMITIR, RECIBIR)

-Lógica de Control: controla la actuación del dispositivo de acuerdo con las indicaciones dadas.

-Trasductor: convierte los datos tanto a la forma de energía necesaria para el dispositivo, como en tamaño y/o formato. Lleva asociado el buffer para poder almacenar los datos durante el tiempo necesario para realizar su función.

**IV.2.1 - TECLADOS Y MONITORES:**

**El modo de interacción más común entre el hombre y la máquina, es la combinación teclado-monitor. El operador provee una entrada oprimiendo las teclas, lo que provoca una salida desde el teclado en forma de señales eléctricas en código ASCII o CCITT Alfabeto Internacional nº 5 (Comité Consultatif Intrernationale de Télègraphie et Télèphonie - Comité Asesor Internacional de Telegrafía y Telefonía que actualmente ha cambiado a la denominación UIC, o Unión Internacional de Comunicaciones) donde cada caracter está representado por un conjunto de 7 bits (8 cuando se considera el bit de paridad), lo que posibilita representar 128 caracteres distintos.**



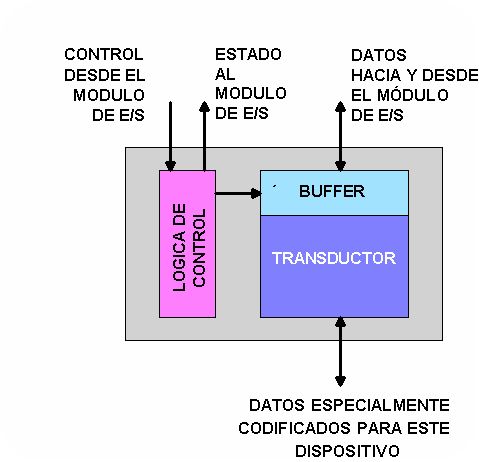
**Figura IV.1 - Módulo de entrada/salida.**

**Estos caracteres son de dos tipos, imprimibles y de control, los primeros a su vez son los alfabéticos, numéricos, de puntuación y especiales y pueden ser representados en una pantalla. Los caracteres de control permiten el ordenamiento de los imprimibles en renglones, páginas etc. así como para su transmisión.**

**La señal eléctrica es interpretada por un codificador incorporado al teclado, y entregada a la máquina a través de la unidad de entrada. En la misma, el caracter puede ser almacenado en el mismo código ASCII, y cuando es entregado mediante la unidad de salida, el transductor del dispositivo convierte las señales electrónicas al formato requerido por el para su presentación o almacenamiento.**

**Existe una amplia variedad de teclados, pueden tener desde unas 15 o 20 hasta más de cien pulsadores, dispuestos en la forma convencional de una máquina de escribir, la distribución de los mismos y los caracteres que codifican pueden variar de país en país. El utilizado por nosotros es denominado qwerty por las primeras seis letras de la primera fila de letras.**

**El principio de funcionamiento de cada pulsador puede ser de diferente índole, los hay por contactos, por variación de capacidad, por variación de reluctancia y por variación de resistencia.**



**Figura IV.2 - Dispositivo externo.**

**En cuanto respecta a las pantallas de video, también existe una amplia variedad, pudiendo ser monocromáticas o policromáticas, con barrido simple o entrelazado y con tamaños que van desde las 12 pulgadas en adelante. Los más empleados actualmente son los monitores color, con barrido entrelazado, de 15 pulgadas de diámetro y un paso de punto de 0,28 mm.**

**El paso de punto es la separación que hay entre los centros de dos puntos sucesivos en la pantalla, lo cual también indica el tamaño (diámetro) de cada punto.**

**Otro factor a tener en cuenta es la definición del sistema gráfico, lo cual es una asociación del paso de punto con la cantidad de puntos que puede almacenar la memoria de pantalla. En general, para un sistema color de 256 colores, y una memoria de pantalla de 1 Megabyte, se tiene una resolución de 1024 x 768 pixel. Un pixel es el punto mínimo direccionable y representable en pantalla.**

**Para sistemas gráficos, se utilizan pantallas de mayores dimensiones, de 17, 19, 21 y hasta 23 pulgadas de diámetro, con paso de punto de 0,27 mm, barrido no entrelazado, y placas de video tipo 3D, con memorias superiores a 8 Megabytes.**

**En los últimos tiempos se han desarrollado pantallas de cristal líquido, también monocromáticas o policromáticas, que son de muy bajo consumo, y elevado brillo. Estas pueden ser iluminadas de dos maneras, o mediante lámparas de descarga gaseosa (tubos fluorescentes) o mediante LEDs, o sea diodos emisores de luz, que son más brillantes y de menor consumo y más duración.**

**V.2.2 - DRIVERS DE DISCOS:**

**Como tercer elemento infaltable en toda computadora, están los discos, tanto el rígido** **como por lo menos uno flexible, aúnque estos ya han sido desplazados por los discos ópticos compactos y los DVD o sea los discos digitales versátiles. Su mecanismo de arrastre y lectura/escritura, es conocido como drive. El drive contiene toda la electrónica necesaria para el control, el intercambio de datos y la emisión de señales de estado, más los mecanismos eléctricos necesarios no solo para hacer girar el disco, sino también para mover las cabezas de lectura/escritura.**

El transductor en este caso es capaz de convertir las señales eléctricas binarias, representativas de cada bit, en pulsos de corriente aptos para excitar las cabezas durante la escritura, y además para convertir las señales leídas en códigos binarios.

V.3 - MODULOS DE ENTRADA/SALIDA:

**Ya dijimos al comienzo que un módulo de entrada/salida es la entidad que dentro de una computadora, es responsable del intercambio de datos con los periféricos. Estos módulos también pueden tener una interfase interna, hacia el resto de la máquina, y una o varias interfaces externas, dirigidas a distintos dispositivos.**

V.3.1 - FUNCIONES DE LOS MÓDULOS DE E/S.

Las principales funciones caen dentro de las siguientes categorías:

- Control y Atiempamiento

- Comunicación con la UCP

- Comunicación con los periféricos

- Almacenamiento y adecuación de datos.

- Detección de errores

**En cualquier momento la UCP puede comunicarse con uno o mas periféricos, de forma no predecible, dependiendo de las necesidades de E/S que tenga el programa. Los recursos internos, tales como memoria principal y bus del sistema, pueden ser compartidos entre un gran número de actividades, incluyendo entrada/salida de datos. Además la entrada/salida de datos incluye requerimientos de atiempamiento y control, para coordinar el tráfico entre las fuentes internas y los dispositivos externos.**

**Por ejemplo, el control de la transferencia de datos desde un periférico a la UPC necesita de la realización de los siguientes pasos:**

1 - La UCP interroga al módulo de E/S para averiguar el estado del dispositivo externo.

2 - El módulo responde entregando su estado.

3 - Si el dispositivo está operacional y preparado para transmitir, la UCP solicita la transferencia de datos, mediante un comando dado al módulo de E/S.

4 - El módulo obtiene una unidad de datos (una palabra) del periférico.

5 - Los datos son transferidos desde el módulo de E/S a la UCP.

**Si el sistema emplea un bus, cada una de las transacciones entre la UCP y el módulo de E/S necesita la realización de uno o más arbitrajes del mismo.**

**Este listado simplificado de eventos, también ilustra sobre la necesidad que tiene la unidad de E/S, de poseer la capacidad de acoplar comunicaciones entre la UCP y el dispositivo externo. Las comunicaciones de la UCP con el periférico necesitan:**

- Decodificación de comandos: el módulo de E/S acepta los comandos de la UCP, comandos que son enviados por el bus de control. Por ejemplo, para un drive de discos, se necesitan los siguientes comandos: LEER SECTOR, ESCRIBIR SECTOR, BUSCAR NÚMERO DE PISTA, Y BARRER LA IDENTIFICACIÓN DE LA GRABACIÓN.

- Datos: Los datos son intercambiados entre la UCP y el módulo de E/S, por medio del bus de datos.

- Reporte de Estados: Debido a que los periféricos son bastante lentos, es importante conocer el estado del módulo de E/S. Por ejemplo, si se le solicita a un módulo de E/S que envíe un dato a la UCP, este puede no estar listo para su transmisión, por cuanto todavía está trabajando sobre un pedido anterior. Esto es reportado mediante una señal de estado. Las señales de estado más comunes son: OCUPADO y LISTO, pero puede haber otras para indicar varias condiciones de error.

- Reconocimiento de la dirección: Dado que toda palabra tiene su dirección en memoria, y lo mismo ocurre con cada periférico, el módulo de E/S deberá reconocer una única dirección para cada periférico que controle.

**Por otra parte, el módulo de E/S debe ser capaz de de llevar a cabo todas las "*comunicaciones con el dispositivo"* que incluyen comandos, información de estado, y datos**.

**Una tarea esencial del módulo de E/S, es el *"tamponamiento de datos".* Lo cual es evidente al observar la tabla siguiente, y que es debido a la diferente velocidad de transferencia entre el interior y el exterior de la máquina.**

**Finalmente, el módulo de E/S es responsable de "*detectar errores"* y el subsiguiente aviso de los mismos la UCP. Algunos de estos errores pueden deberse a fallas mecánicas o eléctricas, reportadas por el periférico (pista dañada en el disco, atascamiento del papel en la impresora, etc)**

**Otras clases de errores consiste en el cambio no intencional del patrón de bits, durante la transmisión, los que pueden ser detectados, por ejemplo, por un control de paridad.**

|  |  |  |  |
| --- | --- | --- | --- |
| **DISPOSITIVO** | **ACCION** | **ASOCIADO A** | **VELOCIDAD**  **(Kbytes/seg)** |
| **Teclado** | **Entrada** | **Humano** | **0,01** |
| **Ratón** | **Entrada** | **Humano** | **0,02** |
| **Entrada por voz** | **Entrada** | **Humano** | **0,02** |
| **Scanner** | **Entrada** | **Humano** | **200.00** |
| **Salida por voz** | **Salida** | **Humano** | **0,6** |
| **Impresora de Línea** | **Salida** | **Humano** | **1.00** |
| **Impresora Láser** | **Salida** | **Humano** | **100.00** |
| **Presentación Gráfica** | **Salida** | **Humano** | **30.00** |
| **UCP a Buffer de Salida** | **Salida** | **Humano** | **200.00** |
| **Terminal de Red** | **Entrada o Salida** | **Máquina** | **0,05** |
| **Red a LAN** | **Entrada o Salida** | **Máquina** | **200.00** |
| **Disco öptico** | **Almacenamiento** | **Máquina** | **500.00** |
| **Cinta Magnética** | **Almacenamiento** | **Máquina** | **2000.00** |
| **Disco Magnético** | **Almacenamiento** | **Máquina** | **2000.00** |

IV.3.2 - ESTRUCTURA DEL MÓDULO DE E/S:

**La Unidades de Entrada y de Salida, o el módulo de E/S, varían mucho en complejidad, así como también es variable la cantidad de periféricos conectados, por lo que solo trataré de hacer una descripción general. En la figura IV.3, se tiene un diagrama de bloques que responde a lo indicado.**

**El módulo se conecta con el resto de la máquina mediante un conjunto de líneas de señales (por ejemplo el bus del sistema). Los datos transferidos desde y hacia el mismo, son soportados por varios registros buffer. También puede haber uno o varios registros de estados, que proveen la información del estado actual.**

**El/los registros de estados, también pueden funcionar como registros de control, pues puede aceptar una detallada información de control remitida por la UCP.**

**La lógica interna del módulo interactúa con la UCP mediante un conjunto de líneas de control. Estas señales de control pueden se utilizadas por el mismo módulo, o puede servir para direccionar dispositivos externos, para lo cual el módulo deberá reconocer y generar las direcciones correspondientes.**

**Finalmente, cada módulo debe contener la lógica especifica para interconectarse con cada dispositivo que el controla.**

**Algunas de estas unidades de E/S, funcionan en modo tal que la UCP prescinde del tipo de dispositivo conectado, de sus señales eléctricas o componentes mecánicos, y aún de su funcionamiento, como para remitir solamente señales de lectura y escritura, por lo cual puede acelerar su velocidad de operación.**

**Tal módulo, es conocido más bien como procesador de E/S, dado que se hace cargo de la mayor parte de las tareas de ese tipo. El más simple y primitivo, en cambio es el que controla un solo dispositivo, y es conocido como controlador de E/S, o controlador de dispositivos.**

**El primero se utiliza en grandes computadores, mientras que el segundo es típico de las microcomputadoras.**

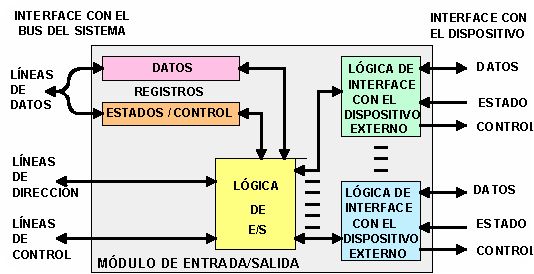
**IV.4 - ENTRADAS Y SALIDAS PROGRAMADAS:**

**Hay tres técnicas posibles para realizar las operaciones de E/S:**

- Entrada/Salida Programada: La UCP ejecuta un programa que le otorga el control total de la operación, incluyendo el sensado del estado del dispositivo, enviar un comando de lectura o escritura, y transferir los datos. Este método es el que menos se aplica, por ser un derrochador del tiempo de la UCP, que no puede hacer ninguna otra operación durante esta ejecución.

- Entrada/Salida manejada por interrupciones: Aquí la UCP entrega un comando para ejecutar una operación de E/S, y continua ejecutando las restantes instrucciones del programa. Una vez terminado el mismo, y el módulo de E/S ha realizado su trabajo previo, la UCP se encargará de entregarle los datos, o de recibirlos y almacenarlos.

- Entrada/Salida por Acceso Directo a Memoria: Suele referirse a éste método como el DMA (Direct Memory Access). En este caso, la memoria y el módulo de E/S intercambian datos, sin intervención de la UCP.



**Figura IV.3 - Diagrama de bloques de un módulo de E/S.**

**IV.4.1 - LINEAMIENTOS GENERALES:**

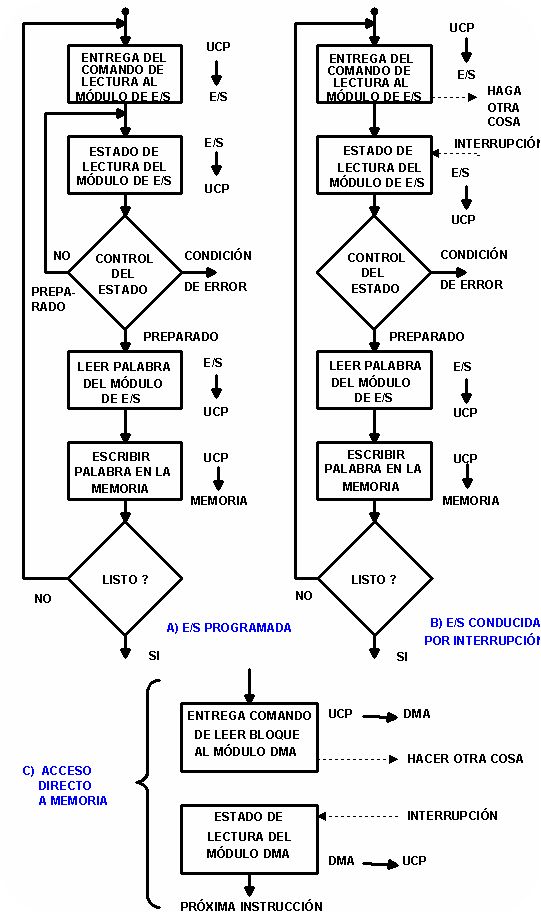
**Mientras la UCP está corriendo un programa y encuentra una instrucción relativa a la E/S, la ejecuta enviando un comando al módulo de E/S adecuado. Con E/S programadas, el módulo puede llevar a cabo la acción requerida y dispone los bits adecuados en el registro de estados. El módulo no toma acciones que avisen a la UPC, en particular no la interrumpe, por tanto la responsabilidad a cargo de ésta, es la de controlar periódicamente el estado del módulo de E/S, hasta que se asegure de la finalización de la operación.**

**La figura IV.4.a, muestra el diagrama de flujo de la aplicación de una operación de E/S programada, para leer un bloque de datos desde un periférico, tal como una cinta magnética, y pasarlo a la memoria.**

**Los datos son leídos de una palabra (por ejemplo 16 bits) por vez, y por cada lectura, la UCP debe quedar en un ciclo de prueba del estado, hasta que reciba la señal que los datos pueden ser tomados en el registro correspondiente del módulo de E/S, este diagrama muestra claramente la pérdida de tiempo que implica este tipo de transferencia para la UCP.**

**IV.4.3 - INSTRUCCIONES DE E/S:**

**Con la E/S programada, hay una fuerte correspondencia entre la instrucción relativa a la E/S que la UCP ha recibido de la memoria, y los comandos de E/S que la misma UCP entrega a un módulo de E/S a fin de ejecutar la instrucción. En consecuencia la instrucción es fácilmente traducida a comandos, a veces en forma tan simple como una relación uno a uno, o sea que cada bit de la instrucción significa un comando específico. El formato de la instrucción depende del modo en el cual son direccionados los periféricos.**



**Figura IV.4 - Diagrama de flujo de tres técnicas para la entrada de un bloque de datos.**

**Normalmente hay varios dispositivos de E/S conectados al sistema mediante módulos de E/S. Cada dispositivo es direccionado mediante un único identificador, que es contenido en el comando remitido por la UCP. En consecuencia, los módulos deben interpretar las líneas de dirección para determinar a quien corresponde ese comando.**

**Cuando la UCP, la memoria principal y la E/S comparten el mismo bus, hay dos modos posibles de direccionamiento, por mapeo de memoria y aislado.**

**1 - *Direccionamiento de e/s por mapeo de memoria*: en este caso existe un solo espacio de direcciones tanto para las locaciones de memoria como para los dispositivos de E/S, por lo que la UCP considera como posiciones de memoria los registros de estados y de datos de los módulos de E/S, utilizando las mismas instrucciones de máquina para acceder a ellos que a las posiciones de memoria. Consecuentemente un bus de direcciones de 10 líneas, es capaz de direccionar 1.024 posiciones, combinando así memoria con dispositivos de E/S. Con éste tipo de direccionamiento, se necesita una sola línea de lectura escritura.**

**2 - *Direccionamiento de E/S aislado:* Si a la línea de lectura y escritura se le agrega otra para seleccionar la memoria interna o el dispositivos de E/S, en un sistema de 10 líneas de dirección, quedarán las 1.024 posibilidades tanto para direccionar la memoria central, como para dispositivos externos. Es así que resulta aislado el espacio de direccionamiento de E/S del correspondiente al de la memoria, de allí el nombre. En general en éste caso se recurre a registros que son los que conectan directamente al dispositivo externo.**

**En la figura IV.5, tenemos contrastadas ambas técnicas, en la parte a, vemos como aparece el sistema mapeado para un programador, y para un simple dispositivo de**

**entrada, tal como un teclado.**

**Suponiendo el mismo sistema de 10 líneas de dirección, con 512 locaciones de memoria de 1 bit (0-511) y hasta 512 direcciones para dispositivos de E/S (512-1023). Dos de ellas son dedicadas a la entrada del teclado. La 516 se refiere al registro de datos y la 517 al registro de estados, el que también funciona como registro de control para la recepción de comandos de la UPC. El programa mostrado debe leer un byte de datos proveniente del teclado y con destino un registro acumulador de la UPC. Notemos que ésta permanece en un lazo hasta que el dato está disponible.**

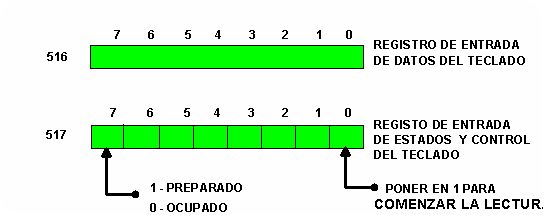
**Con el direccionamiento aislado, parte b de la misma figura, los puertos de E/S solo son accesibles mediante comandos especiales, que activan las líneas de comando del bus.**

**Para diversos tipos de UPC, hay un gran conjunto de instrucciones diferentes para referenciar la memoria, si se utiliza el direccionamiento aislado, solo habrán unas pocas instrucciones. En consecuencia, la ventaja de los sistemas mapeados es que puede utilizarse un largo repertorio de instrucciones, permitiendo una programación más eficiente. La desventaja, es que se usa un espacio de direcciones grande en memoria. Ambos sistemas son de uso común.**

**IV.5 - ENTRADA/SALIDA POR INTERRUPCIÓN:**

**El inconveniente de la E/S programada, es el tiempo que pierde la UPC, lo cual disminuye la performance de todo el equipo.**

**Una alternativa es que ésta, luego de enviar los comandos para una E/S, siga operando hasta que reciba un aviso de que la unidad direccionada está lista para realizar el intercambio solicitado.**



|  |  |  |  |
| --- | --- | --- | --- |
| **DIRECCION** | **INSTRUCCION** | **OPERANDO** | **COMENTARIO** |
| **200** | **CARGAR AC** | **"1"** |  |
| **201** | **ALMACENAR AC** | **517** | **INICIAR LECTURA TECLADO** |
| **202** | **CARGAR AC** | **517** | **OBTENER BYTE DE ESTADO** |
| **203** | **RAMIFICAR SI <SIGNO> = 0** | **202** | **HACER LAZO DESPUÉS DE LEER** |
| **204** | **CARGAR AC** | **516** | **CARGAR EL BYTE DE DATOS** |

**a) E/S por mapeo de memoria.**

|  |  |  |  |
| --- | --- | --- | --- |
| **DIRECCIÓN** | **INSTRUCCIÓN** | **OPERANDO** | **COMENTARIO** |
| **200** | **COMENZAR E/S** | **5** | **INICIAR LECTURA TECLADO** |
| **201** | **PROBAR E/S** | **5** | **CONTROLAR TERMINACIÓN** |
| **202** | **RAMIFICACIÓN NO PREPARADA** | **201** | **HACER LAZO DESPUÉS DE LEER** |
| **203** | **ENTRADA** | **5** | **CARGAR EL BYTE DE DATOS** |

**b) E/S aislada.**

**Figura IV.5 - E/S por memoria mapeada y E/S aislada.**

**Esto funciona así: Desde el punto de vista del módulo de E/S, el ha recibido un comando de LEER de parte de la UCP, por lo que procederá a la lectura del dato en el periférico direccionado. Una vez que el dato está en registro de datos del módulo, éste envía una señal a la UCP (interrupción), por una línea de control, y espera hasta que el dato le sea requerido.**

**Cuando la UCP realiza el requerimiento, el módulo le entrega el dato en bus de datos, y queda a la espera de una nueva operación.**

**Desde el punto de vista de la UCP, el proceso es como sigue: la UCP entrega un comando de Lectura, y sigue con la ejecución de ese u otro/s programa/s. Al final de cada ciclo de instrucción, controla la presencia de interrupciones, cuando encuentre una, la UCP guardará el contexto (o sea los contenidos del contador de programa (CP) y de sus registros) correspondientes al programa en ejecución, y pasa a procesar la interrupción.**

**En nuestro caso, la UCP leerá el contenido del bus de datos y lo almacenará en la memoria. Luego procederá a restaurar el contexto del programa en el que estaba trabajando, o de algún otro, y reasume su ejecución.**

**En la figura IV.5.b, se muestra el uso de una interrupción de E/S para leer un bloque de datos. En comparación con lo mostrado en la figura IV.5.a, la interrupción actúa más eficientemente que la E/S programada, debido a que elimina las demoras por espera. Pero sin embargo, el procedimiento también insume una gran cantidad de tiempo, por cuanto cada palabra dato debe pasar por la UCP en su camino desde la unidad de entrada hasta la memoria.**

**IV.5.1 - PROCESAMIENTO DE INTERRUPCIONES:**

**La presencia de una interrupción dispara cierta cantidad de eventos, tanto en el hardware del procesador como en el software. La figura IV.6, muestra una secuencia típica.**

**Cuando un dispositivo de E/S completa una operación, ocurre la siguiente secuencia de eventos de hardware:**

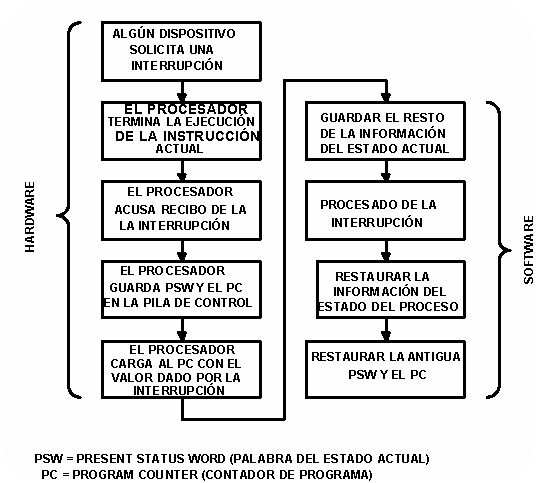
***1 - El periférico envía una señal de interrupción al procesador.***

***2 - El procesador termina la ejecución de la presente instrucción antes de responder a la interrupción.***

***3 - El procesador interroga por si hay interrupciones, al determinar que hay una en espera, emite una señal de reconocimiento al dispositivo que realizó la solicitud. Esto hace que el dispositivo retire la señal de interrupción.***

***4 - El procesador ahora necesita prepararse para transferir el control a la rutina de interrupción. Para empezar, debe guardar la información necesaria para reasumir la ejecución del programa que tenía en curso. El mínimo de información requerido para ello es el estado actual del procesador, que se va actualizando en un registro denominado PSW (Present Status Word - Palabra del Estado Actual), y la ubicación de la próxima instrucción que debe ser ejecutada, la que es contenida en el Contador de Programa (CP). Ambas pueden ser guardadas en Pila de Control del Sistema (PCS) (System Control Stack).***

***5 - El procesador ahora carga el CP con la locación de inicio de la rutina de manejo de interrupciones. En forma dependiente de la arquitectura y del sistema operativo, puede haber un único programa, o un programa por cada dispositivo de E/S y por cada tipo de interrupción. Al haber más de una rutina, la UCP debe primero determinar cual de ellas utilizar, por tanto en cada solicitud de interrupción puede estar incluida la información correspondiente, o el procesador puede tener que interrogar al dispositivo que emitió la señal de interrupción, para que le indique sus necesidades.***



**Figura IV.6 - Procesamiento de interrupciones.**

**Una vez que ha sido cargado el CP, el procesador prosigue con el próximo ciclo de instrucción, el que comienza con una búsqueda, dado que la dirección buscada es dada por el nuevo contenido del CP, el resultado es una transferencia del control al programa de manejo de las interrupciones, lo cual lleva a la realización de las siguientes operaciones:**

***6 - En este momento, la información relativa al CP y a la PSW del programa interrumpido, ha sido guardada en la pila del sistema, sin embargo, hay mas información que es considerada parte de la ejecución del programa, tal como el contenido de los registros del procesador, cuyo contenido también debe ser salvado, dado que pueden ser utilizados por la rutina de manejo de las interrupciones. En consecuencia, los mismos son también guardados en la pila del sistema. En la figura IV.8 se muestra un ejemplo simple, un programa es interrumpido, luego de la instrucción ubicada en la locación N. El contenido de todos los registros, mas el de la próxima instrucción N+1 es ubicado en el Stack. El puntero del mismo es actualizado para que apunte a la nueva entrada, y el CP es actualizado para apuntar a la primera instrucción de la rutina de interrupciones.***

***7 - El manejador de interrupciones puede ahora llevar adelante su procedimiento. Esto puede incluir un examen de la información del estado con relación a la operación de E/S o a cualquier otro evento provocado por la interrupción. También puede involucrar el envío de comandos o señales de reconocimiento al dispositivo solicitante.***

***8 - Cuando el proceso de la interrupción ha sido cumplido, los valores de los registros salvados son tomados de la pila del sistema y vueltos a sus depositarios originales.***

***9 - El acto final es la restauración de la PSW y del valor del CP. Como resultado se ejecutará la próxima instrucción del programa previamente interrumpido.***

**Debe notarse la importancia de salvar toda la información referente al programa en ejecución al momento de presentarse la interrupción, a fin de que el mismo pueda ser reasumido integralmente.**

**IV.5.2 - RECOMENDACIONES PARA EL DISEÑO:**

**Cuando se presenta una interrupción, la UCP debe averiguar ante todo de donde proviene, luego si hay más de una solicitud, determinar a cual atender primero, para ello hay cuatro técnicas en uso actualmente:**

**1 - Múltiples líneas de interrupción**

**2 - Votación por software.**

**3 - Encadenado margarita (Votación por Hardware vectorizada)**

**4 - Arbitraje del bus (vectorizado)**

**La solución más directa es la de proveer varias líneas de interrupción entre la UCP y los módulos de E/S, pero no es práctico utilizar mas de unas pocas líneas del bus para éste propósito. En consecuencia, cuando esto sucede, se dedica cada una de las líneas posibles a varios dispositivos, y se utilizará alguna de las otras técnicas en cada una de ellas.**

**Sobre la alternativa de la Votación por Software, al detectar la UCP una interrupción, pasa a una rutina de servicio de la interrupción, la que tiene por objeto consultar a cada uno de los módulos a fin de determinar cual emitió la solicitud. Esta consulta puede ser en forma de una línea separada de comando, en cuyo caso, cuando la UCP detecta una señal de interrupción, emite una señal de Prueba (TEST I/O), colocando la identificación de cada uno de los periféricos, en forma secuencial, sobre las líneas de dirección, hasta que uno conteste afirmativamente.**

**Una forma alternativa, es la de disponer un registro de estados direccionable en cada módulo de E/S, que son leídos por la UCP a fin de determinar cual ha emitido el pedido de interrupción, con lo que el procesador carga la rutina específica para ese dispositivo.**

**La desventaja de este sistema es el consumo de tiempo, siendo a tal efecto mucho más efectivo el encadenamiento margarita, que provee, de hecho, una votación por hardware. Un ejemplo de esta configuración es dada en la figura IV.7, donde está aplicado al arbitraje de un multibus.**

**La línea de pedido de interrupciones es compartida por todos los módulos de E/S. Cuando la UCP detecta una interrupción, envía una señal de reconocimiento que se propaga en serie por todos los módulos hasta que detecta el causante. El módulo contesta disponiendo una palabra en las líneas de datos, esta palabra es considerada como un vector y es el identificador del dispositivo, que es utilizado por la UCP para apuntar a la rutina de servicio adecuada al periférico. Esta técnica también es denominada "interrupción vectorizada".**

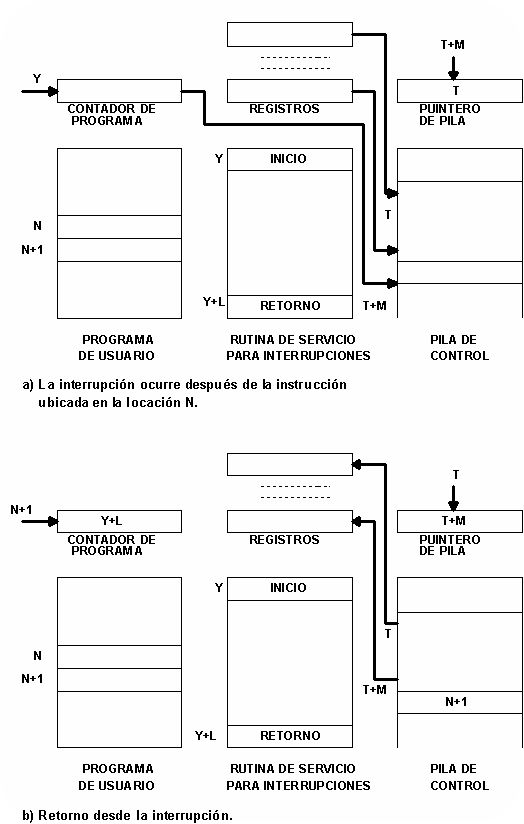
**La otra técnica que hace uso de interrupciones vectorizadas, es el arbitraje del bus, donde un módulo de E/S primero gana el control del bus, antes de que pueda alcanzar la línea de interrupción requerida. En consecuencia, solo un módulo por vez puede tener la línea, y cuando la UCP detecta el pedido, responde por la línea de reconocimiento, entonces el módulo puede entregar su vector en las líneas de datos.**

**Las técnicas indicadas sirven para reconocer al módulo que realizó el requerimiento, pero también pueden proveer la forma de asignar prioridades cuando se superponen los pedidos. En un sistema con múltiples líneas, la UCP tomará primero la de mayor prioridad. Con votación por software, el orden en el cual son consultados los módulos puede indicar su prioridad, y el orden en el cual están dispuestos los módulos en el encadenamiento margarita actúa también como asignación de prioridad. Finalmente, en el arbitraje del bus se puede emplear un esquema de prioridades, tal como se indicó anteriormente.**

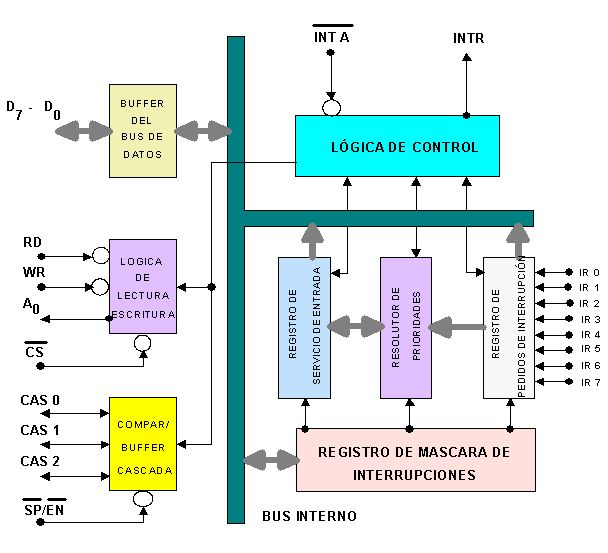
**IV.5.3 - CONTROLADOR DE INTERRUPCIONES INTEL 8259A:**

**Este controlador provee una sola línea para el requerimiento de interrupción (INTR) y una sola para su reconocimiento (INTA). A fin de hacer que el procesador 8086 pueda manejar una amplia variedad de dispositivos, y posea estructuras de priorización, se lo configura con un arbitrador de interrupciones externo de este tipo. En la figura IV.8 se tiene un diagrama de bloques del mismo.**

**En la figura IV.9, se muestra el uso del circuito para conectar hasta ocho módulos. Si es necesario conectar una mayor cantidad, se debe emplear una disposición en cascada del mismo, que puede manejar hasta 64 módulos.**



**Figura IV.7 - Cambios en la memoria y en los registros debidos a una interrupción.**



**Figura IV.8 - Manejador de interrupciones Intel 8259A.**

**El circuito solo tiene por misión el manejo de las interrupciones, para ello, acepta pedidos desde los módulos agregados, determina cual es el de mayor prioridad, y luego lo indica a la UCP mediante la línea INTR. El procesador acepta la interrupción, dando aviso mediante la línea INTA.**

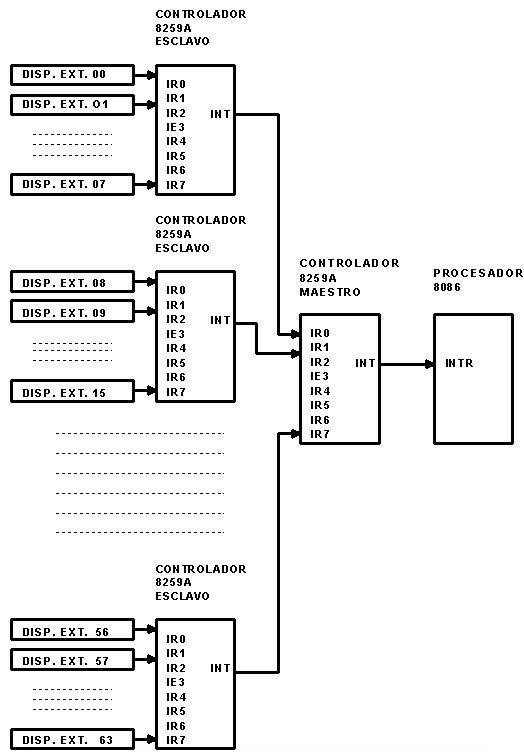
**El controlador coloca el vector de información adecuado sobre las líneas de datos, y la UCP puede proceder a ejecutar el proceso de interrupción, comunicándose directamente con el módulo de E/S para leer o escribir datos.**

**El controlador 8259A es programable, por lo que el procesador puede determinar el esquema de prioridades aplicable, de entre los que siguen:**

**1 - Totalmente anidado: Los pedidos de interrupción son ordenados entre 0 (IR0) y 7 (IR7).**

**2 - Rotativo: En algunas aplicaciones algunos de los dispositivos son de igual prioridad, en este caso, luego de haber servido a un periférico, se le otorga la menor prioridad dentro del grupo.**

**3 - Mascara especial: esto permite a la UCP inhibir selectivamente las interrupciones generadas por algunos dispositivos.**



**Figura IV.9 - Uso del controlador de interrupciones Intel 8259A**

**IV.5.4 - EL INTERFACE PERIFÉRICO PROGRAMABLE (PPI) INTEL 8255A:**

**Como ejemplo de módulo de entrada/salida, utilizado para E/S programadas y manejadas por interrupciones, podemos considerar al 8255A de Intel, que es módulo de propósitos generales, diseñado para ser utilizado con la UCP 8086, integrado en un solo chip. En la figura IV.10 tenemos el diagrama de bloques del mismo.**

**El costado derecho del diagrama es la interfase externa, que posee 24 líneas de E/S programables desde el registro de control del 8086.**

**Las 24 líneas son divididas en tres grupos (A, B y C) de un byte, cada uno de los cuales puede funcionar como puerto de 8 bits. Además, el grupo C puede ser subdividido en dos de 4 bits (CA y C B), los que pueden ser utilizados conjuntamente con los puertos A y B llevando señales de control y de estados.**

**En el costado izquierdo, tenemos la interfase interna con el bus del 8086, e incluye un bus de datos bidireccional (D0 a D7), utilizado para la transferencia de datos e información al registro de control. Las dos líneas de dirección especifican uno de los tres puertos de E/S o el registro de control. Se produce una transferencia cuando se habilita la línea CHIP SELECT conjuntamente con alguna de las señales de LECTURA o ESCRITURA. La línea de RESET es utilizada para inicializar el módulo.**

**El registro de control es cargado por el procesador para controlar el modo de operación y definir las señales. En el modo de operación "0", los tres grupos de 8 líneas externas funcionan como tres puertos de E/S. Cada uno de ellos puede ser definido como de entrada o de salida. De otra forma, el grupo A y el grupo B funcionan como puertos de E/S y las líneas del grupo C como líneas de control para Ay B.**

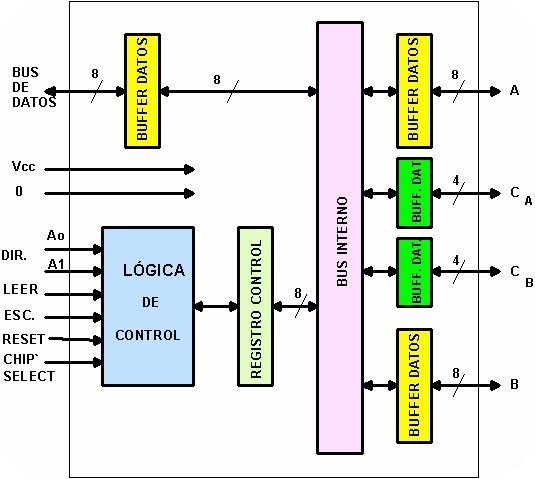
**Las señales de control tienen dos propósitos definidos, el "handshaking" (que en este caso puede interpretarse como coordinación) y para el pedido de interrupciones. La coordinación está basada en un simple mecanismo de tiempo, una línea de control se usa por el emisor como línea para el aviso de "DATOS PREPARADOS", a fin de indicar que los datos están presentes en las líneas de datos de E/S. La otra línea sirve al receptor para que emita la señal de reconocimiento, indicando que los datos han sido leídos y que las líneas de datos pueden ser desactivadas. Otra línea puede ser asignada a la solicitud de interrupciones y vuelta atrás al bus del sistema.**

**Dado que el 8255A es programable mediante el registro de control, puede ser usado para controlar una amplia variedad de periféricos simples. La figura IV.11 ilustra su uso como controlador de teclado y display. El teclado provee 8 bits a la entrada, dos de los cuales, SHIFT y CONTROL, tienen un significado especial para el programa de manejo del teclado. De cualquier forma, su interpretación es transparente al 8255A, el cual simplemente acepta los 8 bits de datos y los presenta en el bus de datos del sistema. Se agregan dos líneas para el control de la coordinación con el teclado.**

**La presentación también es enlazada por un puerto de 8 bits de datos, dos de los cuales tienen también un significado especial que es transparente al circuito, y en adición se agregan dos líneas de coordinación, que proveen un control adicional.**

**IV.6 - ACCESO DIRECTO A MEMORIA:**

**Los modelos de transferencia vistos, siempre hacen uso del procesador para pasar datos de la memoria a un dispositivo externo, o viceversa, con lo cual se le hace perder tiempo en lo que debería ser su fuerte, la elaboración de datos. El acceso directo a memoria tiende a evitar este desperdicio, tratando de que el procesado no pierda tiempo en estas transacciones.**



**Figura IV.10 - Interfase para periféricos programable Intel 8255A (PPI).**

**Una operación de Acceso Directo a Memoria (DMA) es inicializada por la UCP, luego de lo cual sigue con su trabajo, con lo que se puede transferir una gran cantidad de información sin afectar seriamente su performance.**

**IV.6.1 - FUNCIONAMIENTO DEL DMA:**

**Este método de acceso involucra un módulo adicional en el bus del sistema, módulo que es capaz de mimetizar a la UCP, y tomar el control para la realización de las transacciones.**

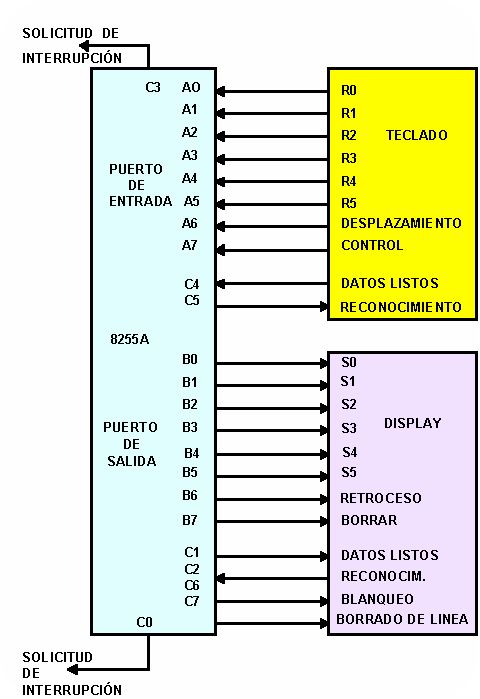
**El sistema trabaja de la siguiente manera, cuando la UCP desea leer o escribir un bloque de datos, envía un comando al módulo DMA, conteniendo la siguiente información:**

**- Que tipo de operación (lectura o escritura) se debe realizar.**

**- La dirección del dispositivo.**

**- La locación de memoria de arranque para la lectura o la escritura.**

**- La cantidad de palabras que deben ser escritas o leídas.**



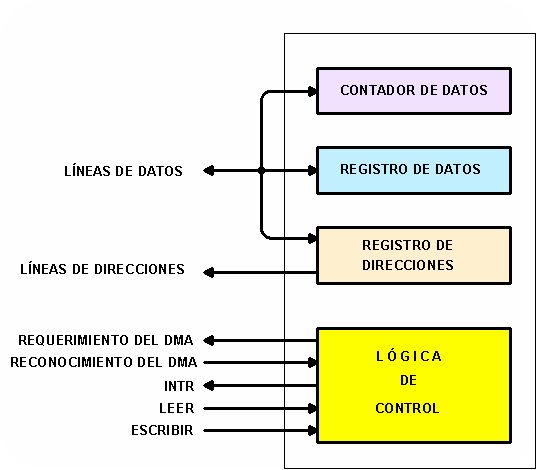
**Figura IV.11 - Interfase de teclado y display con 8255A.**

**La UCP continua con otra tarea, delegando la operación de E/S en el módulo DMA, y éste es el que la tomará a su cargo. Dicho DMA transfiere el bloque de datos palabra por palabra, directamente tomándola o entregándola a la memoria, sin intervención de la UCP.**

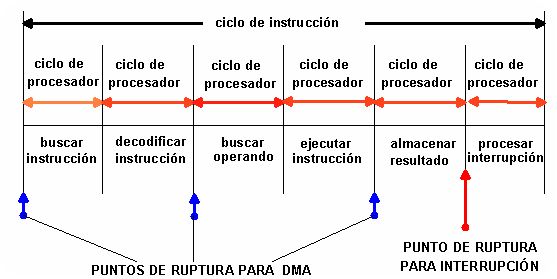
**Una vez concluida la transferencia, el módulo DMA advierte de ello a la UCP mediante una señal de interrupción. En consecuencia el procesador solo toma parte en el comienzo y la finalización del proceso. Lo que ya había indicado en la figura IV.6.c.**

**El DMA necesita adquirir el control del bus para transferir los datos desde y hacia la memoria, por lo cual debe usar el bus solamente cuando la UCP no lo hace, o debe forzar a esta para que suspenda temporalmente la operación. Esta última técnica es más común, y se la conoce como de "robo de ciclo", dado que el DMA accede furtivamente a un ciclo de bus.**

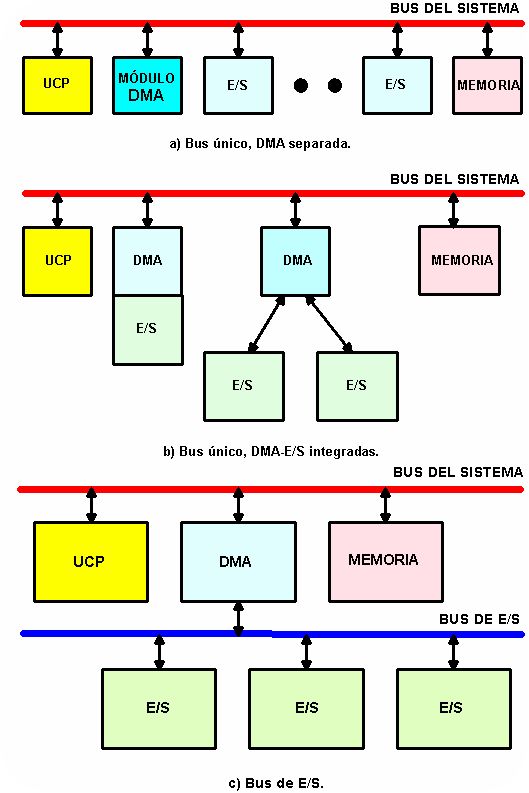
**En la figura IV.13, tenemos indicado en que lugar del ciclo de instrucción puede suspenderse la operación de la UCP. En cada caso esta es suspendida justo antes de que necesite usar el bus, el DMA no interrumpe la operación de la UCP, esta no debe salvar el contexto del programa que está ejecutando, ni hacer ninguna otra cosa, solo se le hace hacer una pausa de un ciclo de bus. Si bien esto retarda la operación, su efecto total en caso de tener que realizar muchas transacciones, es de mucho mejor rendimiento que las otras dos formas, por programa o por interrupción.**



**Figura IV.12 - Diagrama de bloques de una DMA típica.**



**Figura IV.13 - Puntos de ruptura para la DMA y para las interrupciones.**



**Figura IV.14 - Configuraciones posibles con DMA.**

**El mecanismo del DMA puede ser configurado en una amplia variedad de maneras, algunas de las cuales se indican en la figura IV.14.**

**En el primer ejemplo, todos los módulos comparten el mismo bus del sistema. El módulo DMA actúa como un subordinado de la UCP, utilizando una E/S programada para el intercambio de datos entre la memoria y un módulo de E/S, a través del módulo DMA. Esta configuración, que puede ser muy económica, es claramente ineficiente, dado que tal como en la E/S programada, cada transferencia de una palabra, insume dos ciclos de bus.**

**El tiempo, medido en ciclos de bus, puede ser reducido sustancialmente si se integran las funciones de la DMA con las de E/S. Tal como se indica en la figura IV.14.a, ello ocurre cuando se intercala el módulo DMA entre los módulos de E/S y el bus del sistema.**

**Aún más se mejora el sistema, si se dispone un bus de E/S, conectado al bus del sistema mediante un módulo de DMA, tal como se indica en la figura IV.14.c.**

**IV.7 - CANALES DE ENTRADA Y SALIDA Y PROCESADORES:**

**Según ya comentáramos anteriormente, a medida que las computadoras evolucionaron, también lo hicieron las unidades de entrada y salida, en forma tal de ir integrando funciones, hasta convertirse en verdaderos canales de entrada/salida.**

**IV.7.1 - EVOLUCION DE LAS FUNCIONES DE ENTRADA Y SALIDA:**

**Los pasos evolutivos pueden resumirse en la siguiente forma:**

**1 - La UCP directamente controla los periféricos, lo que es notable en dispositivos controlados a microprocesador.**

**2 - Se agrega un controlador o un módulo de E/S. La UCP utiliza un sistema de E/S programada, sin interrupciones. Con este paso la UCP se independiza de los detalles específicos de la interfase del dispositivo externo.**

**3 - Se usa la misma configuración anterior, pero se agregan las interrupciones. La UCP no gasta tiempo en esperar la realización de una operación de E/S, lo cual mejora la eficiencia.**

**4 - El módulo de E/S alcanza directamente la memoria mediante un módulo de DMA, por tanto puede mover un bloque de datos desde o hacia la memoria sin que la UCP intervenga, excepto en el comienzo y final de la transferencia.**

**5 - El módulo de E/S es mejorado para convertirse el mismo en un procesador, con un conjunto específico de instrucciones. La UCP indica al procesador de E/S que ejecute un programa de E/S que está almacenado en la memoria. El procesador los busca y ejecuta sin intervención de la UCP. Ésta puede indicar una secuencia de actividades, y ser interrumpida solamente cuando todo ha sido realizado.**

**6 - Los módulos de E/S tienen también una memoria local, por lo que ya no necesitan buscar su programa de actividades en memoria central, por lo que de hecho estos módulos se convierten en computadoras destinadas a procesar las transacciones de la máquina principal. Esto hace que aumente notablemente la cantidad de dispositivos que pueden conectarse a la máquina. Un uso común de esta arquitectura, lo podemos encontrar en los sistemas de comunicaciones entre terminales interactivas, donde los procesadores de E/S toman a su cargo la mayoría de las tareas involucradas en el control de las terminales.**

**IV.7.2 - CARACTERÍSTICAS DE LOS CANALES DE E/S:**

**El canal de E/S representa una extensión del concepto DMA. Un canal tiene la habilidad de ejecutar instrucciones de E/S, con control total sobre las mismas.**

**En un sistema de computación donde se utilizan estos dispositivos, la UCP no ejecuta instrucciones de E/S, sino que ellas son almacenadas en la unidad de memoria principal, para ser ejecutadas por un procesador de propósitos particulares, dedicado al canal de E/S. La UCP inicia la operación instruyendo al canal para que ejecute el programa almacenado en la memoria, programa que contendrá todas las instrucciones necesarias referentes a los periféricos, sus prioridades, lugar donde deben dejarse los datos, etc.**

**Hay dos tipos de canales de E/S de aplicación masiva, los que se ilustran en la figura IV.15, y son los que actúan mediante selector o mediante multiplexor. El primero puede controlar la transacción con un solo periférico por vez, mientras que el segundo puede actuar sobre varios al mismo tiempo. En este último caso, el multiplexor se encarga de intercalar la información correspondiente a cada dispositivo, en forma tal que cada uno recibe o entrega la información correspondiente.**

**IV.8 - LA INTERFACE EXTERNA:**

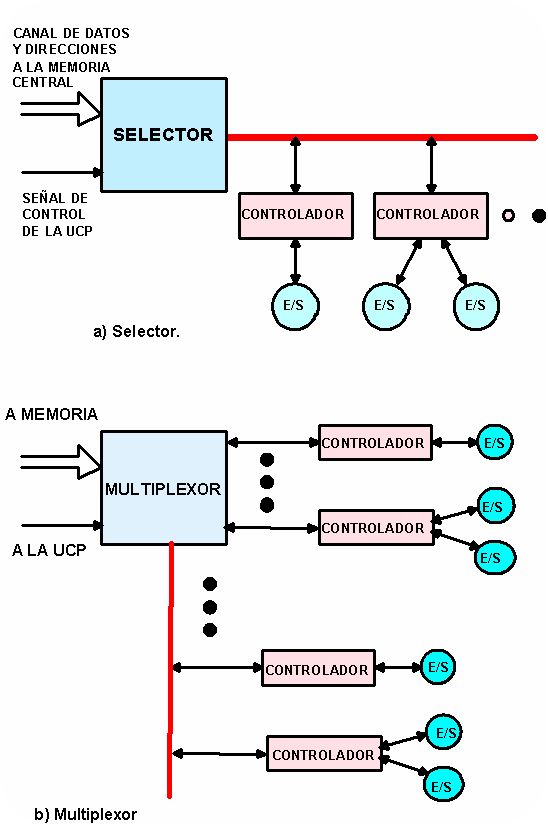
**Como su denominación lo indica, la interfase externa es la que tiene en cuenta la interconexión entre el módulo de E/S y el periférico.**

**IV.8.1 - TIPOS DE INTERFASE:**

**La interfase con un periférico debe ser "hecha a medida" de acuerdo con la naturaleza y la operación del dispositivo, la principal característica es su tipo de conexión, que puede ser *serie* o *paralelo*, ambas de acuerdo al esquema mostrado en la figura IV.16. En el caso a, se tiene la conexión paralelo, en la cual se transfiere una cierta cantidad de bits en forma simultánea, siendo conveniente que la cantidad sea igual a la longitud de palabra utilizada.**

**En la interfase serie, la transferencia se realiza bit a bit, por tanto el módulo debe hacer la conversión correspondiente.**

**El primer tipo se utiliza en sistemas de alta velocidad, tales como discos o cintas, en cambio el segundo se reserva para aquellos periféricos de menor velocidad, tal como impresoras y terminales.**



**Figura IV.15 - Arquitectura de canal para E/S.**

**En ambos casos, el módulo de E/S debe establecer una "conversación" con los periféricos, en términos generales, el diálogo para una operación de escritura es el siguiente:**

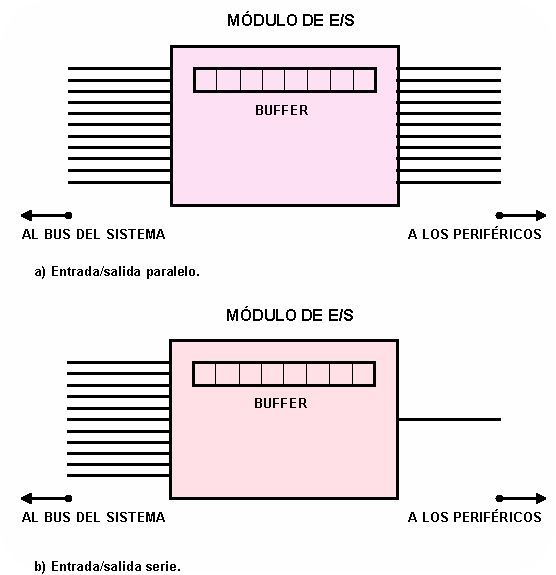
**1 - El módulo de E/S envía una señal de control solicitando autorización para mandar datos.**

**2 - El periférico reconoce el pedido.**

**3 - El módulo de E/S transfiere los datos (una palabra o bloque de palabras, de acuerdo al periférico).**

**4 - El periférico reconoce la aceptación de los datos.**

**Una operación de lectura, se produce de manera similar.**



**Figura IV.16 - Entradas/salidas paralelo y serie.**

**La clave de la operación del módulo de E/S es un registro que puede almacenar datos y entregarlos a la velocidad del periférico, y en la forma requerida por él, en serie o paralelo.**

**IV.8.2 - CONFIGURACIONES PUNTO A PUNTO Y CONFIGURACIONES MULTIPUNTO:**

**La conexión entre un módulo de E/S y un periférico puede acontecer en cualquiera de dos formas, punto a punto o multipunto. En el primer caso, la interfase provee líneas dedicadas entre el módulo de E/S y el dispositivo externo. En sistemas pequeños (Computadoras Personales, estaciones de trabajo).**

**Los enlaces dedicados típicos son los correspondientes al teclado, la impresora, y el módem externo. El ejemplo característico de tales interfases es la especificación EIA-232, también conocida como RS-232.**

**Al aumentar la importancia, aparecen las interfases externas multipunto, utilizadas para soportar dispositivos de almacenamiento masivo (discos y cintas) y los periféricos multimedia (CD-ROM, video, audio). Estas interfases son efectivamente buses externos, y exhiben el mismo tipo de lógica de los buses.**

**TEMA B: BUSES DE ENTRADA Y SALIDA**

**BUSES DE E/S:**

**Un bus es un medio de transmisión compartido que interconecta dos o más dispositivos y permite que se establezca entre ellos una correcta comunicación.**

**Un bus se compone de diferentes conductores eléctricos denominados líneas.**

**Se pueden encontrar diferentes tipos de líneas en un bus:**

* + **SEGÚN SU FUNCIÓN: líneas de datos, líneas de direcciones o líneas d e control.**
  + **SEGÚN SU USO: líneas dedicadas o líneas multiplexadas.**
  + **SEGÚN SUS CARACTERÍSTICAS ELÉCTRICAS: líneas unidireccionales o bidireccionales.**

**Los buses de e/s suelen estar estandarizados para que los distintos dispositivos de diferentes fabricantes puedan conectarse correctamente en todas las arquitecturas.**

**Los estándares especifican una serie de requisitos eléctricos, mecánicos y de protocolos de comunicación que deben cumplir los buses y las líneas que los componen.**

**Los parámetros de caracterización más importantes de un bus se suelen separar en diferentes niveles:**

* + **NIVEL FÍSICO.**
  + **LÍNEAS.**
  + **MODO DE OPERACIÓN.**
* **NIVEL FÍSICO:** 
  + **CAPACIDAD DE CONEXIÓN (NÚMERO MÁXIMO DE DISPOSITIVOS QUE PUEDEN CONECTARSE AL BUS)**
  + **LONGITUD MÁXIMA DEL BUS**
  + **SOPORTE FÍSICO**
  + **NIVELES DE TENSIÓN**
  + **FRECUENCIA DE FUNCIONAMIENTO, TIPO DE CONECTORES, ETC.**
* **LÍNEAS:** 
  + **ANCHO DEL BUS (NÚMERO TOTAL DE LÍNEAS)**
  + **ANCHO DE DATOS**
  + **LÍNEAS UNIDIRECCIONALES O BIDIRECCIONALES, ETC.**
* **MODO DE OPERACIÓN:**
  + **PROTOCOLOS DE TRANSFERENCIA (que determinen el tipo de comunicación que se establece entre los dispositivos conectados al bus)**
  + **PROTOCOLOS DE SINCRONIZACIÓN (que determinen el inicio y el fin de cada transferencia de información)**
  + **PROTOCOLOS DE ARBITRAJE (que controlen el acceso al bus cuando más de un dispositivo puede actuar como maestro).**

**JERARQUÍAS DE BUSES:**

**Si en una computadora conectamos todos los dispositivos a un único bus, aparecen dos problemas: primero, una disminución global en el rendimiento del sistema por los retardos de propagación de las señales debidos a la longitude del bus, a los tiempos de espera para conseguir la utilización del bus y a la adaptación para dar servicio a los dispositivos más lentos; Segundo, incompatibilidad de los distintos dispositivos con el bus, ya que cada fabricante diseña sus propias interfaces optimizadas para la función que cumplen.**

**La solución a estos problemas es utilizar una jerarquía de buses en lugar de un único bus. Para el diseño de la jerarquía de buses se debe tener en cuenta la ubicación de los diferentes niveles de la jerarquía de la memoria, la ubicación de los controladores e interfaces de E/S, la necesidad de algunos dispositivos de accader directamente a la memoria principal y las diferentes necesidades en cuanto a ancho de banda de los dispositivos.**

**Normalmente se utilizan las jerarquías de buses en lugar de un único bus, que incorporan los siguientes tipos de buses:**

* + **BUS DEL SISTEMA Y BUS DE MEMORIA.**
  + **BUSES DE EXPANSIÓN.**

**BUS DEL SISTEMA Y BUS DE MEMORIA:**

**Son los que conectan el procesador con el resto del sistema y la memoria principal con el controlador de memoria respectivamente. Se trata de buses rápidos y cortos, propietarios (no estandarizados) y optimizados para arquitecturas y diseños específicos.**

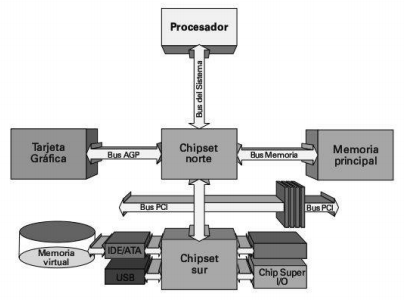
**Esta optimización es posible ya que a estos buses se conectan un número fijo de dispositivos de prestaciones conocidas.**

**BUSES DE EXPANSIÓN:**

**Se trata de buses más largos y lentos, abiertos (estandarizados), accesibles por el usuario y a los que se conectan un número indeterminado de dispositivos de prestaciones desconocidas y muy diferentes entre sí.**

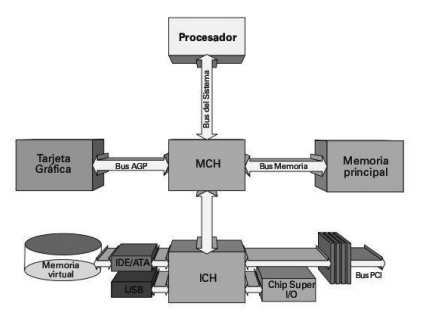
**EVOLUCIÓN DE LA JERARQUÍA DE BUSES EN LAS COMPUTADORAS PERSONALES**

**Hasta hace unos años, la jerarquía de buses en las placas base de las computadoras personales era la siguiente:**



**El procesador se comunica con el controlador de memoria , integrado en el chipset norte, mediante el bus del sistema. Al chipset norte se conecta la memoria principal (mediante el bus de memoria) y también la tarjeta gráfica (mediante alguna de las versiones de AGP, un bus de expansión estándar que se utiliza sólo para la conexión de tarjetas gráficas). El resto de dispositivos de E/S se conectan al sistema a través del bus PCI (bus de expansión tradicional) o de otros buses e interfaces más específicos, como puede ser el IDE/ATA para el disco duro. La conexión entre el chipset sur, que centraliza todos los flujos de información de E/S, y el chipset norte , se realiza mediante el bus PCI.**

**Este diseño de jerarquía comenzó a presentar cuellos de botella, especialmente en la conexión entre chipsets, ya que por ejemplo, la tarjeta de red suele estar conectada en una de las ranuras PCI, por lo se realizaron los siguientes cambios:**



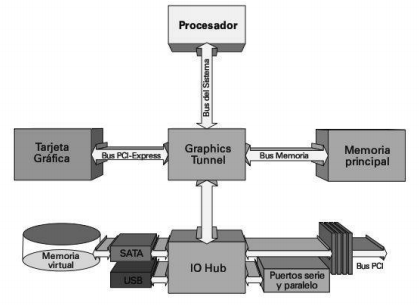
**El bus PCI se conecta directamente al nuevo chipset sur, denominado ICH (IO Controller Hub). La conexión entre el ICH y el MCH ( Memory Controller Hub) se realiza mediante un bus dedicado de ancho de banda bastante mayor que PCI.**

**No se interfiere con los dispositivos conectados al bus PCI, ni con los dispositivos que se conectan directamente al ICH (USB e IDE/ ATA) y no tienen que pasar por el bus PCI.**

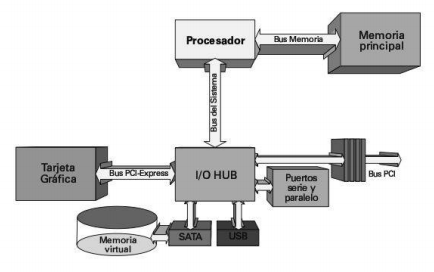
**En las últimas modificaciones, al integrar los últimos modelos de procesadores el controlador de memoria , la conexión de la memoria principal se hace directamente al procesador en lugar de al MCH. El bus estándar para la conexión de tarjetas gráficas también se ha modificado , siendo en los diseños más actuales PCI express en lugar de AGP.**

**Se pueden encontrar tres alternativas para el diseño de jerarquía de buses en PCs:**

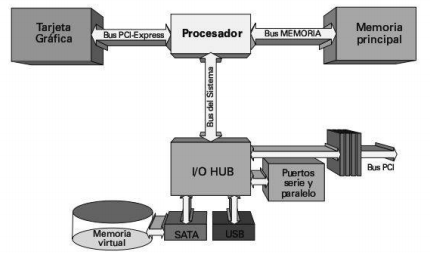
**Alternativa N°1:**



**Alternativa N°2:**



**Alternativa N°3:**



**ANCHO DE BANDA DEL BUS**

**El ancho de banda (bandwidth) del bus, se refiere a la cantidad de información que es capaz de transferir por unidad de tiempo**

**BW = ancho de datos . f . n° transferencias/ciclo**

**ancho de datos: n° de líneas de datos del bus**

**f: frecuencia de funcionamiento de esas líneas**

**n° transferencias/ciclo: transferencias de información que realiza el bus por cada ciclo de reloj.**

**OPTIMIZACIÓN DE UN BUS**

**Cualquier optimización que pretenda mejorar el ancho de banda de un bus debe ir orientada a incrementar alguno de estos tres aspectos:**

* + **EL ANCHO DE DATOS**
  + **LA FRECUENCIA DE OPERACIÓN DEL BUS**
  + **EL NÚMERO DE TRANSFERENCIAS POR CICLO.**

***Ancho de datos y frecuencia de operación:* El ancho de datos está limitado por las interferencias que se producen entre unas líneas y otras, especialmente a frecuencias de operación altas. Es difícil encontrar buses con un ancho de datos por encima d e 64 o 128 bits.**

**La tendencia de optimización actual es la utilización de buses con un ancho cada vez menor, casi serie en lugar de paralelo, y que utilicen *señalización diferencial* para soportar frecuencias de funcionamiento mayores.**

**SEÑALIZACIÓN DIFERENCIAL**

**Los buses tradicionales dedican un conductor para cada bit de información que se desea transmitir por el bus.**

**La *señalización diferencial* utiliza dos conductores para cada bit de información que se desea transmitir por el bus.**

**Por ejemplo, se desea optimizar un bus con un ancho de datos de 32 bits , que realiza una transferencia por ciclo y cuya frecuencia de funcionamiento es de 333 MHz.**

**BW = 4B . 333 MHz .1 = 1.3 GB/s**

**Si se aumenta el ancho de datos como si se aumenta la frecuencia de funcionamiento, comienzan a aparecer interferencias entre las líneas del bus.**

**Cada una de las líneas lleva una tensión de entre 0 y 5 V. Los valores entre 0 y 1.5 V se interpretan como 0 lógico , y los valores entre 3.5 y 5 V se interpretan como 1 lógico (el resto de valores son indeterminados ) .**

**Supongamos que en el bus original se produce una interferencia que resta 0.5 V a todas las líneas del bus.**

**Entonces:**

**3.8 V - 0.5 V = 3.3 V - En el Rx se reconoce estado Indefinido**

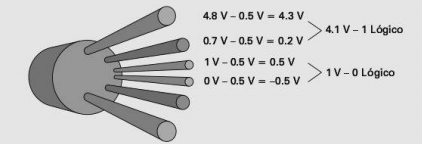
**4.0 V - 0.5 V = 3.5 V - En el Rx se reconoce estado 1 Lógico**

**0 V - 0.5 V = -0 .5 V - En el Rx se reconoce estado Indefinido**

**3.5 V - 0.5 V = 3.0 V - En el Rx se reconoce estado Indefinido**

**En cambio si se toma la diferencia entre dos líneas como se muestra en la siguiente figura:**

**Por el primer par de conductores en el Rx se reconocerá un 1 Lógico y por el segundo par de conductores en el Rx se reconocerá un 0 Lógico, a pesar de tener todos los conductores una interferencia que resta 0.5 V.**



**Entonces, las 32 líneas que se empleaban en el bus original para transmitir 32 bits, ahora se emplean para transmitir 16 bits.**

**Gracias a la mayor robustez frente al ruido, la frecuencia de funcionamiento del bus puede aumentarse hasta 1.5 GHz.**

**El ancho de banda del bus optimizado es: BW = 2B . 1.5 GHz . 1 = 3 GB /s**

**Se ha pasado de transmitir 1.3 GB/s a transmitir 3 GB/s, con la misma cantidad de líneas, utilizando señalización diferencial y gracias a esta, aumentando la frecuencia de funcionamiento del bus.**

**NÚMERO DE TRANSFERENCIAS POR CICLO:**

**Mejorar en todo lo posible el modo de operación del bus y sus protocolos de transferencia , sincronización y arbitraje para que le diera tiempo a realizar más de una transferencia de información por ciclo.**

**Se aprovechan los flancos de subida y de bajada de las señales de reloj, o se combinan varias señales de reloj desfasadas.**

**Casi siempre se realiza un número de transferencias por ciclo que sea potencia de dos.**

**Accelerated Graphics Port (AGP)**

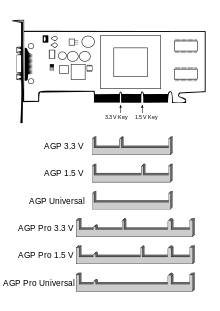
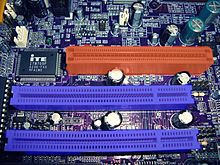
**Las diferentes versiones de este bus fueron mejorando sus prestaciones incrementando simplemente el número de transferencias de información por ciclo, de manera que se pasó del AGP inicial al AGP 2X (dos transferencias por ciclo) , al AGP 4X (cuatro transferencias por ciclo) y por último , al AGP 8X (ocho transferencias por ciclo).**

**BW(AGP) = 4B . 66 MHz . 1 = 264 MB /s**

**BW(AGP2X) = 4B . 66 MHz .2 = 528 MB /s**

**BW(AGP4X) = 4B . 66 MHz . 4 = 1.05 GB/s**

**BW(AGP8X) = 4B . 66 MHz • 8 = 2.1 GB/**



**UTILIZACIÓN DE PROTOCOLOS DE COMUNICACIONES DE ALTO RENDIMIENTO**

**Las técnicas de optimización de buses que se han impuesto en la actualidad se basan en la utilización de protocolos de comunicación de alto rendimiento.**

**Los buses de E/S dentro de una computadora cada vez se parecen más a las redes de comunicaciones y se maneja la misma terminología: conexión punto a punto , paquete de datos, switch.**

**La idea principal es: evitar la señalización de control , y utilizar protocolos de red dentro de las arquitecturas, codificando la información de control junto con los datos, enrutando los paquetes resultantes mediante switches desde el origen hasta el destino. Ejemplos de este tipo de optimización son buses como PCI Express o Hipertransporte (HyperTransport).**

**PCI-Express**

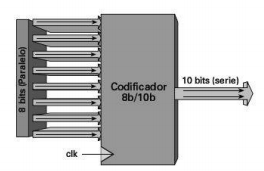
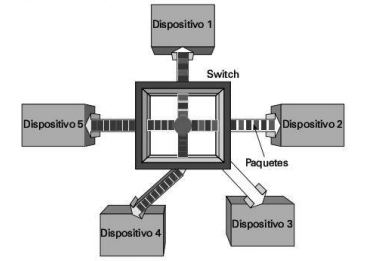
**Tecnología de conexión punto a punto basada en el antiguo bus PCI que permite la mínima latencia de comunicaciones en enlaces entre chips.**

**Está basada en uno o varios switches dentro del sistema que actúan como controladores de todos los dispositivos que utilizan PCI-Express y que se encargan de la gestión de los paquetes que se utilizan para transferir información por el bus.**



**El protocolo de codificación de paquetes que utiliza PCI-Express es síncrono, codificando la señal de reloj junto con los datos de manera que se transmiten 10 bits de información por cada 8 de datos.**

**El estándar soporta cuatro tipos básicos de paquetes, los de transacción con memoria, transacción de E/S, configuración y mensajes (gestión avanzada de interrupciones) .**



**El ancho de datos de este bus puede ser de 1, 2, 4, 8, 12 o 16 bits, con un canal para cada dirección y señalización diferencial ( PCI Express x1, x2, x4 ,x8 ,x12 o x16)**

**Un enlace PCI Express puede ser x1, x2, x4, x8 , x12 o x16 según el número de lanes que lo compongan . Un lane es el conjunto de 4 líneas, dos en un sentido y dos en el contrario (porque se utiliza señalización diferencial y es un bus full-duplex ).**

**Un lane que permite la transmisión de 1 bit en los dos sentidos al mismo tiempo.**

**Los dispositivos PCI-Express negocian con el switch cuántos canales o lanes podrán utilizar para sus comunicaciones.**

**La frecuencia de funcionamiento de este bus es siempre de 2.5 GHz, pero la versión 1.0 hace 0.8 transferencias por ciclo , y la 2.0, justo el doble, 1.6 transferencias por ciclo .**

**BW (PCI Express x 16 1.0) = 2B . 2.5 GHz . 0.8 = 4 GB/s en cada sentido (porque es full-duplex)**

**BW (PCI Express x 16 2.0) = 2B . 2.5 GHz . 1.6 = 8 GB/s en cada sentido (porque es full-duplex)**

**Hipertransporte (HyperTransport)**

**Tecnología de conexión punto a punto que permite la mínima latencia de comunicaciones en enlaces entre chips.**

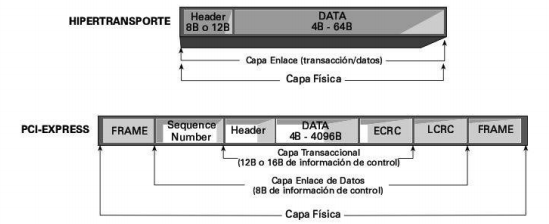
**El estándar está diseñado como un *protocolo de conexión universal*, que no se limita a los buses dentro de una computadora.**

**También está basado en uno o varios switches dentro del sistema que actúan como controladores de todos los dispositivos que utilizan Hipertransporte.**

**Los switches se encargan de la gestión de los paquetes que se utilizan para transferir información por el bus.**

**El protocolo de codificación de paquetes es síncrono , pero mucho más eficiente que el utilizado por PCI Express (se reduce mucho la información de control , se utilizan menos cabeceras).**

**El tamaño de los paquetes es siempre un múltiplo de 4 bytes.**



**El ancho de datos de este bus puede ser de 2, 4, 8, 16 o 32 bits , con un canal para cada dirección y *señalización diferencial*.**

**La frecuencia de funcionamiento varía entre 800 MHz y 3.2 GHz dependiendo de la versión, y siempre se realizan dos transferencias de información por ciclo.**

**Este bus se está utilizando como *bus del sistema* y como *bus de memoria* , no sólo para la conexión de dispositivos periféricos como PCI Express hasta el momento .**

**Un ejemplo típico de ancho de banda para bus del sistema sería:**

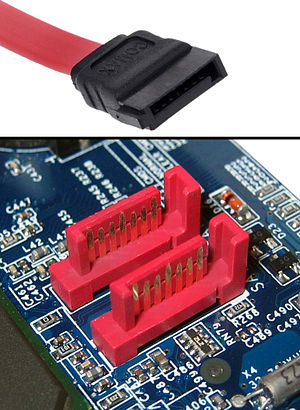
**BW(HT versión 3.1) = 4B . 3.2 GHz . 2 = 25.6 GB/s en cada sentido (porque es full-duplex )**

**Serial ATA**

**Serial ATA o SATA (acrónimo de *Serial Advanced Technology Attachment*) es una interfaz de transferencia de datos entre la placa base y algunos dispositivos de almacenamiento, como puede ser el disco duro o rígido, lectores y regrabadores de CD/DVD/BR, Unidades de Estado Sólido (SSD).**

**La primera generación específica en transferencias de 150 MB por segundo, también conocida por SATA 150 MB/s o Serial ATA-150.**

**Actualmente se comercializan dispositivos SATA II, a 300 MB/s, también conocida como Serial ATA-300 y los SATA III con tasas de transferencias de hasta 600 MB/s.**

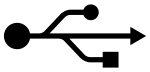


**En la siguiente tabla se muestran funciones de cada uno de los pines del conector SATA:**

|  |  |  |
| --- | --- | --- |
| **Pin 1** | **GND** | **Masa** |
| **Pin 2** | **HT+/DR+** | **Transmisión diferencial +** |
| **Pin 3** | **HT-/DR-** | **Transmisión diferencial -** |
| **Pin 4** | **GND** | **Masa** |
| **Pin 5** | **HR-/DT-** | **Recepción diferencial -** |
| **Pin 6** | **HR-/DT+** | **Recepción diferencial +** |
| **Pin 7** | **GND** | **Masa** |

**De la anterior se ve que utiliza *señalización diferencial* y modo de transmisión *full dúplex***

**Universal Serial Bus (USB)**



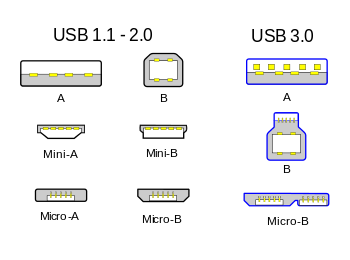
**Existen distintas versiones 1.0, 1.1, 2.0, 3.0 que se diferencian por la velocidad de transferencia de datos. Se denominan de:**

**Baja velocidad (1.0): Tasa de transferencia de hasta 1,5 Mbit/s (188 kB/s).**

**Velocidad completa (1.1): Tasa de transferencia de hasta 12 Mbit/s (1,5 MB/s)**

**Alta velocidad (2.0): Tasa de transferencia de hasta 480 Mbit/s (60 MB/s) pero con una tasa real práctica máxima de 280 Mbit/s (35 MB/s).**

**Superalta velocidad (3.0): Tiene una tasa de transferencia de hasta 4,8 Gbit/s (600 MB/s).**



**En la figura anterior se muestran los distintos tipos de conectores utilizados según la versión.**

**El cable USB 2.0 dispone de cuatro líneas, un par para datos, y otro par de alimentación.**

**La velocidad del bus (USB 3.0) es diez veces más rápida que la del USB 2.0, debido a que han incluido 5 contactos adicionales (visibles en la figura anterior).**

**Las señales del USB se transmiten en un cable de par trenzado cuyos hilos se denominan D+ y D-,utilizan *señalización diferencial* en modo de transmisión *half dúplex* (HDX) excepto el USB 3.0 que utiliza un segundo par de hilos para realizar una comunicación en modo de transmisión *full dúplex* (FDX).**

