

# Connecting Processors, Memory and I/O Devices

## Conexión de Procesadores, Memoria y Dispositivos de Entrada y Salida

Anderson G. Roa, Cristian J. Martínez Alvarez  
*Universidad Industrial de Santander*

*Abstract—the following document is focused on describe the different media and protocols that can be used in the communication between the different interfaces and devices with the computer. (Abstract)*

*Keywords—interfaces; net; I/O (input / output); devices; bridge (key words)*

*Resumen—El siguiente documento está enfocado en describir los diferentes medios y protocolos que pueden ser usados en la comunicación entre las diferentes interfaces y dispositivos con el computador.*

*Keywords—interfaces; net; I/O (input / output); devices; bridge (key words)*

posibilidad de adoptar periféricos diversos y de cualquier fabricante, siempre que sean ‘compatibles’. Pero para que estos periféricos puedan conectarse con distintas maquinas era necesario estandarizar el tipo de conexión que hay entre los periféricos y las maquinas, gracias a este desarrollo computacional se abrió la posibilidad de formar diferentes configuraciones en una máquina, bajando costos de las máquinas de propósito general y a la vez se potencia la capacidad de computadoras de propósitos específicos en la industria mundial.

### I. INTRODUCCION

En un sistema computacional, los diversos subsistemas deben tener interfaces entre si que se comunican a través de protocolos de transferencia que permiten enviar y recibir información comúnmente a través de buses. Antiguamente los ordenadores utilizaban una topología de bus único, denominado bus del sistema o backplane para conectar procesador, memoria y módulos de E/S, pero cuando el número de dispositivos conectados a un bu aumenta disminuye su rendimiento, por ello para evitar la caída de dicho rendimiento, el sistema de buses se jerarquiza, apareciendo dos buses más: el bus local, y el bus de E/S. El primero es de longitud pequeña, de alta velocidad y adaptado a la arquitectura particular del sistema para maximizar el ancho de banda entre el procesador y la cache, aislando el tráfico procesador-cache del resto de transferencias del sistema. y el bus de E/S o de expansión reduce el tráfico en el bus del sistema, la existencia de estos buses permite diseñar una amplia gama de controladores de periféricos compatibles.

Debido a esto las computadoras de la plataforma PC puede ser sistemas abiertos que admiten el incremento de funciones y la

### II. ESTADO DEL ARTE

El cambio en las conexiones internas del procesador ha venido de la mano de evoluciones que se han producido en las capacidades de integración de cada momento. El front-side bus, también conocido como FSB es el tipo de bus usado como bus principal en algunos de los antiguos microprocesadores de la marca Intel para comunicarse con el circuito integrado auxiliar, es el que utilizaban procesadores como los Core 2 Duo o los Atom. Dicho bus incluye señales de datos, direcciones y control, así como señales de reloj que sincronizan su funcionamiento. En los nuevos procesadores de Intel desde Nehalem, y desde hace más tiempo que estos primeros en los AMD, se usan otros tipos de buses como el Intel QuickPath Interconnect y el Hyper Transport respectivamente.

Hyper Transport se puede encontrar tanto en equipos de AMD, como de Nvidia, o Apple. AMD decide incluir el controlador de memoria dentro del chip y debido a esto descarga el bus de un gran trasiego de información.

### III. MARCO TEÓRICO

Una computadora está constituida por el procesador, la memoria y un módulo de E/S, que se comunican entre sí. En

efecto, es una red de módulos elementales, y por consiguiente deben existir líneas para interconectar estos módulos. El conjunto de líneas que conectan los diversos módulos se denomina estructura de interconexión. El diseño de esta estructura dependerá de los intercambios que deban producirse entre los módulos. Los que se necesitan son:

**Memoria:** Recibe y envía datos. Recibe direcciones. Recibe señales de control (leer, escribir, temporizar).

**Módulo de E/S:** Recibe señales de control de la computadora. Envía las señales de control a los periféricos. Recibe direcciones de la computadora. Envía señales de interrupción. Recibe y envía datos.

**Procesador:** Lee instrucciones y datos. Escribe datos una vez los ha procesado. Envía señales de control a otras unidades. Recibe señales de interrupción.

La estructura de interconexión debe dar cobertura a los siguientes tipos de transferencias:

- Memoria a procesador y viceversa.
- E/S a procesador y viceversa.
- Memoria a E/S y viceversa.

La estructura de interconexión más común son las de bus y buses múltiples.

#### INTERCONEXIÓN CON BUSES:

Un bus es un camino de comunicación entre dos o más dispositivos ya que se trata de un medio de transmisión compartido. Cualquier señal transmitida por uno de esos dispositivos está disponible para que los otros dispositivos conectados al bus puedan acceder a ella. Un bus está constituido por varios caminos de comunicación capaces de transmitir señales binarias representadas por 1 y por 0. El bus que conecta los componentes principales del computador se denomina bus del sistema.

#### ESTRUCTURA

El bus del sistema es un conjunto de conductores eléctricos paralelos. Estos conductores son líneas de metal grabadas en una tarjeta que se extiende a través de todos los componentes del sistema, cada uno de los cuales se conecta a algunas o a todas las líneas del bus. Está constituido por entre 50 y 100 líneas que se pueden clasificar en tres grupos funcionales: líneas de datos, de direcciones y de control.

Las líneas de datos proporcionan un camino para transmitir datos entre los módulos del sistema. Generalmente consta de 8, 16 o 32 líneas distintas, cuyo número se conoce como anchura.

El número de líneas determina cuántos bits se pueden transferir al mismo tiempo.

Las líneas de dirección se utilizan para designar la fuente o el destino del dato situado en el bus de datos. La anchura del bus de direcciones determina la máxima capacidad de memoria posible en el sistema. Las líneas de direcciones generalmente se utilizan también para direccionar los puertos de E/S.

Las líneas de control se utilizan para controlar el acceso y el uso de las líneas de datos y de direcciones. Las señales de control transmiten tanto órdenes como información de temporización entre los módulos del sistema. Las líneas de control típicas son: E/L en memoria, petición de interrupción, señales de reloj.

En cuanto al funcionamiento del bus, si un módulo desea enviar un dato a otro debe hacer dos cosas: obtener el uso del bus y transferir el dato a través del bus. Si un módulo desea pedir un dato a otro módulo debe obtener el uso del bus y transferir la petición al otro módulo mediante las líneas de control y dirección apropiadas.

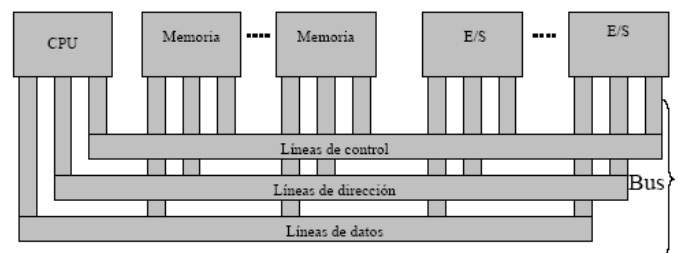


Fig. 1. Interconexión mediante un bus.

#### IV. BUSES: CONEXIÓN DE LOS DISPOSITIVOS DE E/S AL PROCESADOR Y LA MEMORIA

En un sistema computador, cada uno de los diferentes subsistemas debe tener interfaces con el resto. Por ejemplo, la memoria y el procesador necesitan comunicarse, de la misma forma que necesitan comunicarse el procesador y los dispositivos de E/S. Esto se hace habitualmente mediante un bus, este es un canal de comunicación compartido, que utiliza un conjunto de cables para conectar múltiples subsistemas. Las dos ventajas más importantes de la organización en bus son versatilidad y bajo coste. Una vez definido el sistema de conexión, se puede añadir de forma sencilla nuevos dispositivos, y los periféricos pueden pasarse incluso de un sistema a otro, si ambos usan el mismo tipo de bus, efectivos en coste porque el conjunto de cables se comparte de múltiples formas.

La desventaja más importante del bus es que crea un cuello de botella en la comunicación, limitando la máxima productividad de E/S. Cuando la E/S debe pasar por un bus, el ancho de banda

de este bus limita la productividad máxima de la E/S. Por ello es un reto muy importante el diseño de un sistema de bus capaz de satisfacer los requerimientos del procesador, en los sistemas comerciales, en los que la E/S es muy frecuente, y en los supercomputadores, donde la velocidad de la E/S debe ser muy alta porque el rendimiento de los procesadores es alto.



fig. 2. Cable bus.

Una de las razones por las cuales el diseño de un bus es tan difícil es que la velocidad máxima del bus está limitada por factores físicos, como la longitud del bus y el número de dispositivos. Estos factores físicos impiden usar el bus a cualquier velocidad. Existen varias técnicas para mejorar el rendimiento del bus; sin embargo, estas técnicas pueden afectar negativamente a otras métricas de rendimiento. Por ejemplo, para obtener un tiempo de respuesta pequeño para las operaciones de E/S, se debe minimizar el tiempo para realizar el acceso al bus simplificando el camino de comunicación. Por otra parte, para mantener velocidades altas de E/S se debe maximizar el ancho de banda del bus. El ancho de banda del bus puede aumentarse añadiendo más buffers y transmitiendo bloques grandes de datos, lo cual incrementa el tiempo necesario para realizar el acceso al bus. Claramente, estos dos objetivos, acceso rápido al bus y ancho de banda elevado, pueden producir conflictos en los requerimientos de diseño. Finalmente, la necesidad de transferencia muy diferentes, contribuye a complicar aún más el diseño del bus.

Normalmente, un bus tiene un conjunto de señales de control y un conjunto de señales de datos. Las señales de control se usan para señalar peticiones y reconocimientos y para indicar que tipo de información pasa por las líneas de datos. Esta afirmación puede ser datos, ordenes complejas o direcciones. Por ejemplo, si un disco quiere escribir en memoria los datos de un sector, las líneas de datos se usarán para indicar la dirección de memoria en la que se deben escribir los datos del sector, y también para trasladar los datos del sector. Las líneas de control se usarán para indicar el tipo de información presente en las líneas de datos del bus en cada instante de transferencia. Algunos buses tienen dos conjuntos de señales para comunicarse por separado los datos y las direcciones en una sola transmisión. En cualquier caso, las líneas de control se

usan para indicar que es lo que contiene el bus y para realizar el protocolo del bus. Dado que el bus es compartido, se necesita también un protocolo para decidir quién es el siguiente en usar el bus.

Consideremos una transacción de bus típica. Una transacción de bus tiene dos partes: el envío de la dirección y el envío o recepción de los datos. Las transacciones de bus se definen típicamente en función de la operación que se realiza en la memoria. Una transacción de lectura transfiere datos de memoria (o bien al procesador o bien a un dispositivo de E/S), y una transacción de escritura escribe datos en memoria. Una operación de entrada consiste en la entrada de datos desde el dispositivo a la memoria, donde el procesador los habrá escrito. La figura 2 muestra los pasos típicos de operación de salida, en la que se leerán los datos de memoria y se enviarán al dispositivo. Mientras que la figura 3 muestra los pasos de una operación de entrada en la que los datos se leen del dispositivo y se escriben en memoria.

## V. CONEXIÓN DE PERIFÉRICOS AL COMPUTADOR

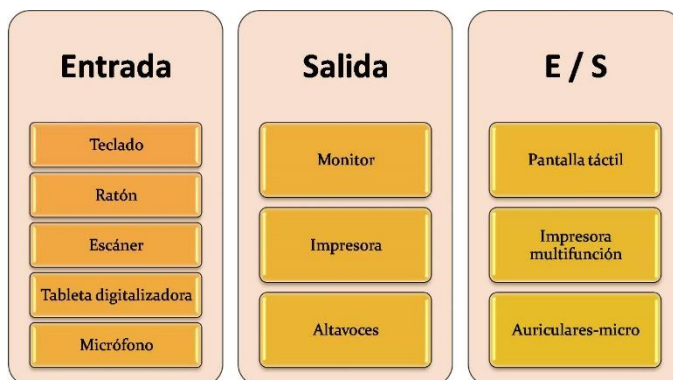


Fig. 3 Dispositivos periféricos.

Los periféricos se conectan a la CPU a través de grupos de hilos que se conoce como buses. En el interior del computador el bus transmite la información de los datos en paralelo. El bus que conecta la CPU con los otros elementos del procesador se conoce como bus local o bus de la CPU. Es un bus muy rápido y conecta la CPU con las tarjetas de la placa base y los controladores de los dispositivos externos. Las conexiones entre los periféricos y los controladores o tarjetas de la placa base se realizan a través de un bus más general llamado bus del sistema. También suele conectar algunas ampliaciones de memoria.

Algunos periféricos requieren un bus especializado que se adapte a su velocidad de transferencia, sus niveles de tensión, la naturaleza de sus señales de control y otros requerimientos. A estos buses se les llama bus de entrada/salida o bus de expansión. Por todo lo visto, los computadores grandes al disponer de varios tipos de buses requieren de dispositivos adaptadores o de interconexión entre buses. Los procesadores

suelen tener unas ranuras de expansión (6 normalmente) sobre la placa base que están conectadas al bus del sistema y que permiten conectar una serie de dispositivos a este bus a través de tarjetas de circuito integrado y que permiten conectar varios dispositivos a la CPU, como por ejemplo tarjetas digitalizadoras de imágenes, aceleradores gráficos con FPGAs, etc.

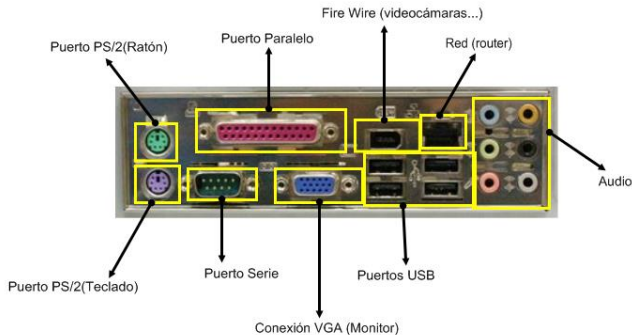


Fig. 4. Conectores.

## VI. JERARQUIA DE BUSES

Si se conecta un gran número de dispositivos al bus, las prestaciones pueden disminuir. Hay dos causas principales:

1. Cuanto más dispositivos se conectan al bus, mayor es el retardo de propagación.
2. El bus puede convertirse en un cuello de botella a medida que las peticiones de transferencia acumuladas se aproximan a la capacidad del bus. Sin embargo, este problema se puede resolver, por ejemplo, incrementando el bus de datos.

Por consiguiente, la mayoría de las computadoras utilizan varios buses, normalmente organizados jerárquicamente.

### Bus del sistema (backplane)

Los ordenadores antiguos utilizaban una topología de bus único, denominado bus del sistema o backplane, para conectar procesador, memoria y los módulos de E/S, tal como la que se muestra en la siguiente figura:

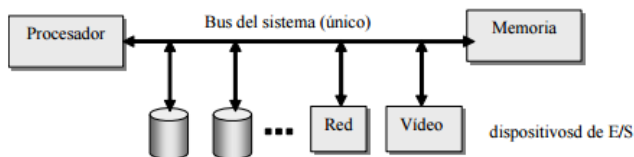


Fig. 5. Bus del sistema ó backplane..

Sin embargo, cuando el número de dispositivos conectados a un bus aumenta disminuye su rendimiento. Esto es debido a dos motivos fundamentales. El primero el aumento del retardo de propagación de las señales debido al aumento de longitud de los conductores que dan soporte al bus. Esto afecta especialmente a la propagación encadenada de la señal de concesión del bus. El segundo el incremento de demanda de acceso que se produce al aumentar el número de dispositivos conectados. Este exceso de dispositivos puede crear un cuello de botella que haga que el rendimiento del sistema se degrade por la espera inútil que se origina cuando tienen que realizar transferencias. Por otra parte, las diferencias en la velocidad de operación de los dispositivos conectados al bus, también repercuten negativamente en su rendimiento.

En efecto, los dispositivos lentos pueden ocasionar retrasos importantes a los rápidos.

Para evitar la caída de rendimiento, el sistema de buses se jerarquiza, apareciendo dos buses más: el bus local, y el bus de E/S.

### Buses locales

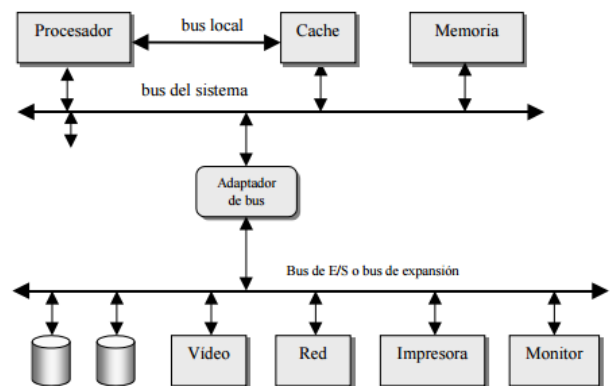


Fig.6. Bus Local .

El bus local es de longitud pequeña, de alta velocidad, y adaptado a la arquitectura particular del sistema para maximizar el ancho de banda entre el procesador y la caché, por eso suele ser un bus propietario. Este bus aísla el tráfico procesador-caché del resto de transferencias del sistema.

### Buses de E/S o de expansión

El bus de E/S o de expansión reduce el tráfico en el bus del sistema, de manera que el procesador puede acceder a memoria en un fallo de caché mientras realiza una operación de entrada/salida.

Los buses de expansión son buses estándar o abiertos (ISA, EISA, PCI, etc.) es decir, independientes del computador y con unas características bien definidas en el correspondiente

documento de normalización. La existencia de estos buses permite diseñar una amplia gama de controladores de periféricos compatibles.

Para conectar los buses del sistema y de expansión se requiere un Adaptador de Bus, dispositivo que permite adaptar las distintas propiedades de ambos buses: velocidad, carácter síncrono o asíncrono, multiplexación, etc.

El proceso de jerarquización se puede complicar más con otras topologías que den cabida a dispositivos de distinta velocidad. De esta forma se equilibra mejor el tráfico de información en sistemas que operan con muchos dispositivos conectados. En la figura siguiente se muestra una topología jerárquica en la que se contempla un bus de alta velocidad del que cuelgan dos buses, uno de expansión para dispositivos rápidos y otro de E/S para dispositivos lentos.

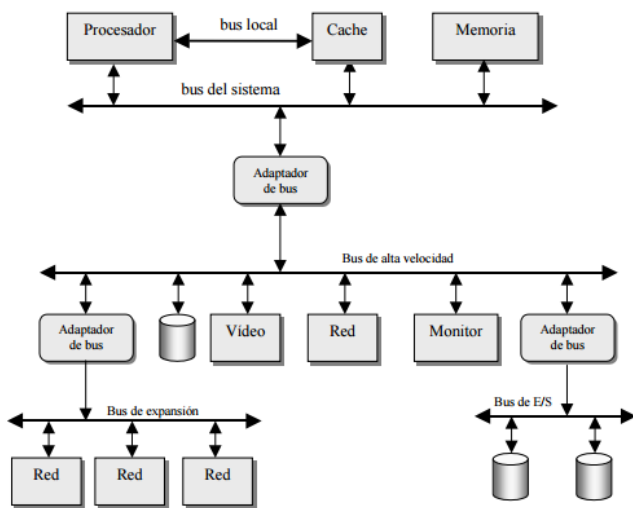


fig.7. Buses de expansión.

## VII. ELEMENTOS DE DISEÑO DE UN BUS

<b>Tipo</b>	Dedicado Multiplexado	<b>Anchura del bus</b>	Dirección Datos
<b>Método de arbitraje</b>	Centralizado Distribuido	<b>Tipo de transferencia de datos</b>	Lectura Escritura
<b>Temporización</b>	Síncrono Asíncrono		Lectura-modificación-escritura Lectura-después de-escritura Bloque

Fig. 8. Elementos de diseño de un bus.

### TIPO

Las líneas de bus se pueden dividir en dos tipos genéricos:

**Dedicadas:** están permanentemente asignadas a una función o a un conjunto de componentes del computador. Un ejemplo típico de dedicación funcional es el uso de líneas separadas para direcciones y datos. **Multiplexadas:** pueden utilizarse para propósitos diferentes.

### METODO DE ARBITRAJE

**Método de arbitraje** En un momento dado puede haber más de un controlador intentando usar el bus, pero sólo uno de ellos puede obtener el control y usarlo para transmitir. Es necesario disponer de un método de arbitraje que determine quién utiliza el bus. En general, los diversos métodos de arbitraje se pueden clasificar en: Centralizados Distribuidos

### TEMPORIZACION

El término temporización hace referencia a la forma en la que se coordinan los eventos en el bus. Con temporización síncrona, la presencia de un evento en el bus está determinada por una señal de reloj. Con temporización asíncrona, la presencia de un evento en el bus es consecuencia y depende de que se produzca un evento previo.

### ANCHURA DEL BUS

La anchura de un bus es el número de líneas distintas que lo componen, afecta tanto a la capacidad como al rendimiento del sistema.

Cuanto más ancho es el bus de datos, mayor es el número de bits que se transmiten a la vez y, por tanto, mayor es el rendimiento del sistema.

Cuanto más ancho es el bus de direcciones, mayor es el rango de posiciones a las que se puede hacer referencia y, por tanto, mayor es la capacidad máxima del sistema.

### TIPOS DE TRANSFERENCIA DE DATOS

Un bus permite varios tipos de transferencias de datos. Todos los buses permiten transferencias de lectura y escritura.

Transferencias de lectura: esclavo → maestro

Transferencias de escritura: maestro → esclavo

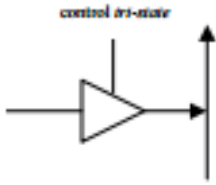
En una escritura, el maestro pone el dato tan pronto se estabiliza la dirección y el esclavo ha podido reconocerla.

En una lectura, el esclavo pone el dato tan pronto como haya reconocido la dirección y disponga del mismo.

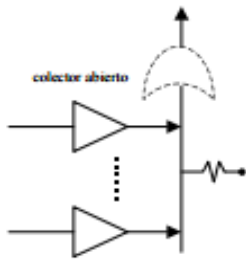
### VIII. DIRECCIONABILIDAD DE BUSES

Las líneas de un bus se pueden clasificar en función de su direccionalidad:

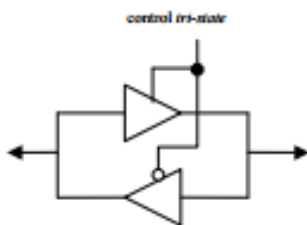
#### 1. Líneas unidireccionales.



- Emisor Múltiple



#### 2. Líneas Bidireccionales.



Desde el punto de vista físico un bus es un conjunto de conductores eléctricos paralelos dispuestos sobre una tarjeta de circuito impreso. Los dispositivos del Sistema se conectan a través de conectores (*slots*) dispuestas a intervalos regulares a lo largo del bus.

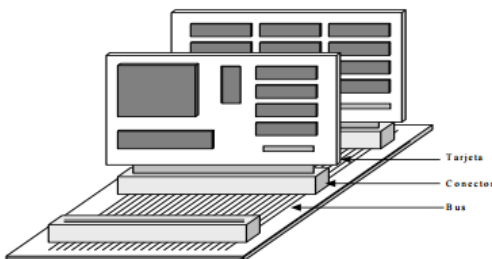


Fig. 9. Buses de Sistema

La disposición anterior corresponde generalmente a los buses del Sistema. Los denominados buses de entrada/salida (E/S) conectan las unidades a través de cables.

Los principales parámetros que caracterizan un bus son:

- Ancho de banda: velocidad de transferencia medida en Mb/s.
- Anchura del bus: número de líneas que lo componen.
- Ancho de datos: número de líneas de datos.
- Capacidad de conexión: número de unidades conectadas al bus.

### IX. PROTOCOLOS DE TRANSFERENCIA

#### 1. Síncronos.

En los buses síncronos existe un reloj que gobierna todas las actividades del bus, las cuales tienen lugar en un número entero de ciclos de reloj. La transferencia propiamente dicha coincide con uno de los flancos del reloj (el de bajada en el ejemplo de la siguiente figura)

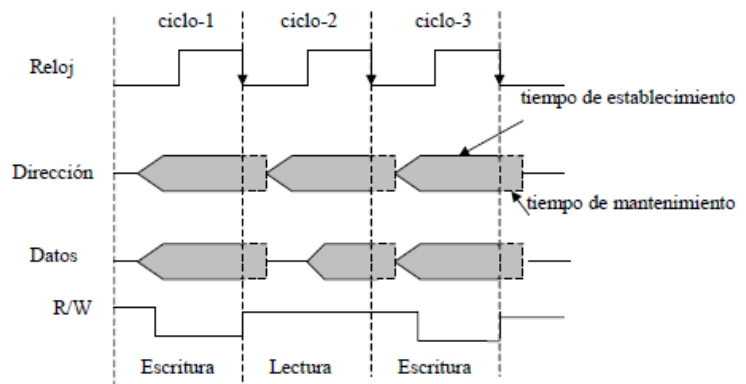


Fig. 10. Diagrama de ciclo de reloj.

En este caso se puso que el tiempo de acceso al *slave* es menor de un ciclo, es decir, en cada ciclo tiene una operación con memoria. Los buses síncronos son rápidos, pero no tienen capacidad para conectar unidades con velocidad de transferencia baja o no conocida a priori.

#### 2. Asíncronos

Los buses asíncronos utilizan un protocolo tipo *handshaking* para comunicarse el *master* con el *slave*. En el siguiente diagrama se presenta el diálogo de señales que tiene lugar durante una transacción de lectura de memoria por parte de la CPU utilizando un protocolo asíncrono (*handshaking*) sobre un bus que multiplexa las direcciones y los datos sobre las mismas líneas (*datos/dirección*).



- 1) La CPU active la señal de *lectura* al tiempo que coloca la dirección de la posición a leer en las líneas *datos/dirección*.
- 2) La Memoria detecta la activación de *lectura*, lee la dirección que hay en *datos/dirección* y activa la señal de *reconocimiento* para indicar que ha detectado la orden de *lectura*.
- 3) La CPU detecta la activación de *reconocimiento* y en respuesta se activa la señal de *lectura* y libera las líneas de *datos/dirección*.
- 4) La memoria detecta que se ha desactivado la señal de *lectura* y desactiva la señal de *reconocimiento* para dar por terminada la orden de lectura.
- 5) Cuando la memoria ha accedido al dato y lo tiene preparado lo pone en *datos/dirección* y activa la señal de *dato disponible*.
- 6) La CPU detecta que *dato disponible* está activo y procede a leer los datos del bus y activar seguidamente la línea de *reconocimiento* para indicar que ya dispone del dato.
- 7) La memoria al detectar la señal de *reconocimiento* desactiva *dato disponible* y libera las líneas de *datos/dirección*.
- 8) Finalmente, la CPU al detectar que se desactiva *dato disponible*, desactiva, a su vez, la señal de *reconocimiento*, indicando que la transmisión ha finalizado.

A partir de este momento se puede iniciar una nueva transacción.

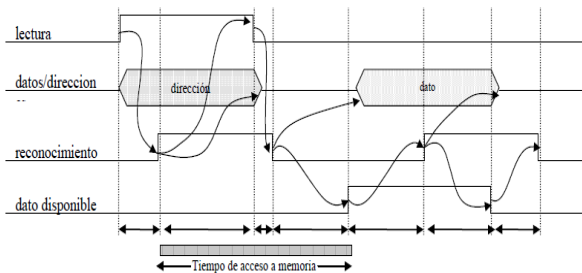


Fig. 11. Diagrama de diálogo de señales.

Un bus asíncrono trabaja igual que un par de máquinas de estados finitos que se comunican de tal forma que uno de los autómatas no avanza hasta que sabe que el otro autómata ha alcanzado un determinado estado, es decir, los dos autómatas están coordinados.

Los buses asíncronos se escalan mejor con los cambios de tecnología y pueden admitir una mayor variedad de velocidades de respuesta en los dispositivos.

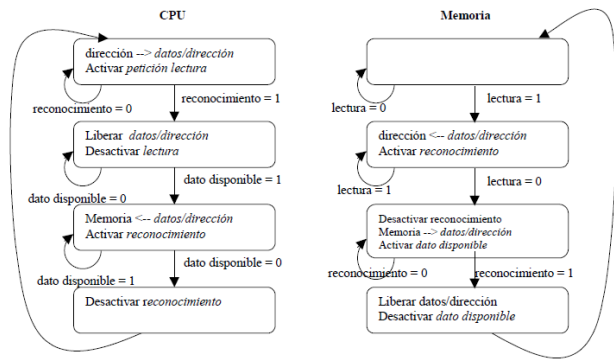


Fig. 12. Autómatas de máquinas de estado finito.

### 3. Semisíncrono

En los protocolos semisíncronos existe, como en los síncronos, un reloj que gobierna las transferencias en el bus. Sin embargo, en este caso existe, además, una señal de *espera* (*wait*) que es activada por el *slave* cuando la transferencia va a durar más de un ciclo de reloj. De esta forma, los dispositivos rápidos operarán como un bus síncrono, mientras que los lentos alargarán la operación el número de ciclos que les sea necesario.

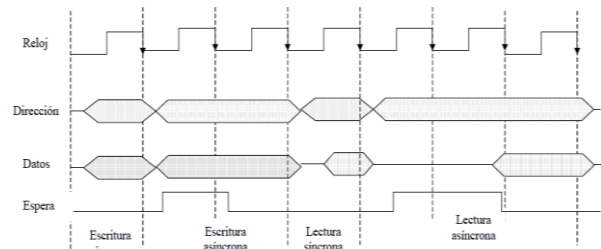


Fig. 13. Diagrama de ciclo de reloj semisíncrono.

### 4. Ciclo partido

En los buses de ciclo partido la operación de lectura se divide en dos transacciones no continuas de acceso al bus. La primera transacción es la de petición de lectura que realiza el *master* sobre el *slave*. Una vez realizada la petición el *master* abandona el bus. Cuando el *slave* dispone del dato leído, inicia un ciclo de bus actuando como *master* para enviar el dato al antiguo *master*, que ahora actúa como *slave*.

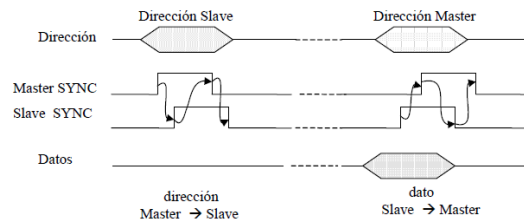


Fig. 14. Diagrama de ciclo partido.

## X. PROTOCOLOS DE ARBITRAJE

La demanda para utilizar el bus en un computador puede provenir de diferentes unidades, no solo de la CPU. Por ejemplo, si el computador tiene E/S por DMA, esté demandará el uso del bus cuando tenga el control de la operación con la memoria. Los procesadores de E/S necesitan acceder al bus no solo para realizar las transferencias de datos por DMA, sino también para leer su programa de canal. Por supuesto, si el sistema es multiprocesador (más de una CPU con memoria compartida) los candidatos al uso del bus aumentan.

Para garantizar que en todo momento sólo una unidad acceda al bus, se utilizan los protocolos de arbitraje. Los protocolos de arbitraje organizan el uso compartido del bus, estableciendo prioridades cuando más de una unidad solicita su utilización y garantizando, sobre todo, que el acceso al bus es realizado por un solo *master*.

Existen dos grupos de protocolos de arbitraje, los *centralizados* y los *distribuidos*. En los primeros existe una unidad de arbitraje, el árbitro del bus, encargado de gestionar de forma centralizada el uso del bus. El árbitro puede ser una unidad físicamente independiente o estar integrado en otra unidad, por ejemplo, la CPU. Por el contrario, en los protocolos distribuidos no existe ninguna unidad especial para la gestión del bus. Esta se realiza de forma distribuida entre las unidades de acceso.

### 1. Protocolo de encadenamiento (*daisy chaining*) de dos señales

Es el protocolo centralizado más sencillo ya que utiliza sólo dos señales encadenadas, una de *petición* del bus y otra de *concesión*.

El *master* que quiere acceder al bus activa la señal de petición (*out*) y los demás *masters* la propagan hasta el árbitro. El árbitro activa la señal de concesión que es propagada por los *masters* que no solicitaron el acceso al bus. El *master* que recibe la señal de concesión y tiene una petición pendiente toma el control del bus. Si un *master* recibe una señal de petición mientras está accediendo al bus, bloquea su propagación al árbitro hasta que finalice la utilización del bus.

La prioridad viene determinada por la proximidad al árbitro.

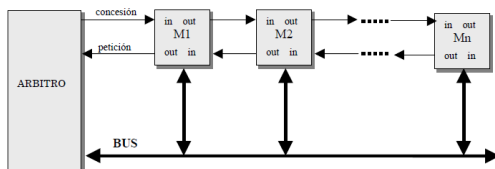


Fig. 15. Diagrama Protocolo de encadenamiento de dos señales.

### 2. Protocolo de encadenamiento (*daisy chaining*) de tres señales

Utiliza una línea más que el protocolo anterior, la línea de ocupación. Además, la línea de petición no es encadenada sino compartida por todos los masters a través de una entrada al árbitro con capacidad de O-cableada.

Cuando un master toma el control del bus activa la línea de ocupación. El árbitro solo activa la línea de concesión cuando recibe una petición y la línea de ocupación está desactivada. Como en el caso anterior, si un master recibe la concesión y no ha solicitado el bus, transmite la señal al siguiente master. Un master toma el control del bus si tiene una petición local pendiente, la línea de ocupación está desactivada y recibe el flanco de subida de la señal de concesión.

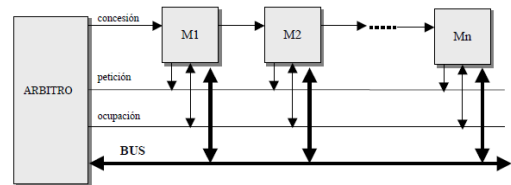


Fig. 16. Diagrama Protocolo de encadenamiento de tres señales.

### 3. Protocolo de encadenamiento (*daisy chaining*) de cuatro señales

Este protocolo permite simultanear el uso del bus por un *master* con el proceso de arbitraje para la selección del *master* siguiente. De esta forma, cuando el primer *master* abandona el bus, no se pierde tiempo en el arbitraje para el siguiente porque ya se ha hecho, pasando directamente el *master* seleccionado a realizar su transacción, al tiempo que se realiza la selección del siguiente *master*. Para ello se añade una cuarta línea al esquema anterior, la línea de *reconocimiento*.

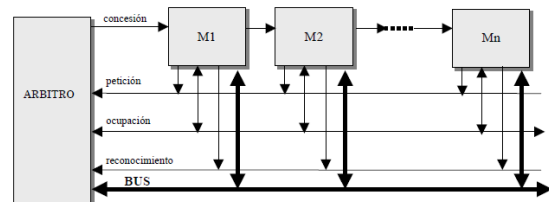


Fig. 17. Diagrama Protocolo de encadenamiento de cuatro señales.

La línea de *reconocimiento* la activa un *master* que solicitó el bus (*activó petición*) y recibió la *concesión* pero la línea de *ocupación* estaba activa (bus ocupado). Cuando el árbitro recibe la activación de *reconocimiento* inhibe su actuación, es decir, deja de atender la señal de petición y genera la de concesión. El *master* queda en espera para ocupar el bus tan pronto lo abandone su actual usuario, que lo hará desactivando la señal de



esto ocurre, el *master* ocupa el bus y desactiva la señal de *reconocimiento*, con lo que el árbitro recupera su actuación, procediendo a un nuevo arbitraje entre los *master* solicitantes, simultáneamente con la operación de transacción en el bus. En la siguiente figura hemos representado el diálogo de señales correspondiente a una ocupación del bus por el *master* M1, seguido por el arbitraje a favor de M2 mientras M1 realiza su transacción, y terminando con la ocupación del bus por M2 cuando M1 finaliza:

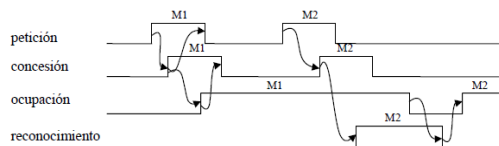


Fig. 18. Diagrama de diálogo de señales.

#### 4. Protocolo con concesión por encuesta (polling)

Este protocolo sustituye la línea encadenada de concesión del bus por un conjunto de líneas que permiten acceder de forma selectiva a la dirección asignada cada master sobre estas líneas.

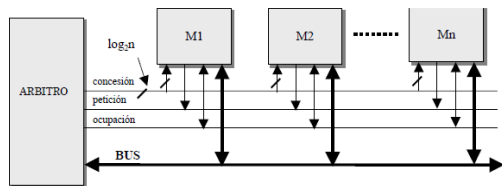


Fig. 19. Diagrama Protocolo con concesión por encuesta.

#### 5. Protocolo con señales independientes

También denominado en estrella, utiliza una línea de concesión específica para cada línea de petición independiente. Esta alternativa tiene la ventaja que el árbitro puede aplicar distintos algoritmos de decisión en caso de peticiones simultáneas (FIFO, prioridad fija, prioridad variable). Además, los retardos de propagación de las señales son pequeños en comparación con las anteriores alternativas.

Tiene la desventaja del número elevado de líneas de arbitraje (una por posible master). El número de master queda limitado al número de líneas existentes.

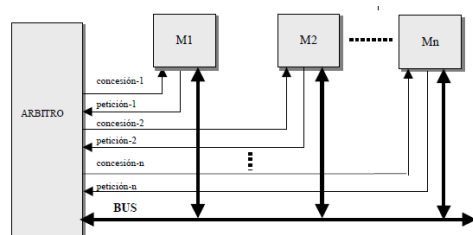


Fig. 20. Diagrama Protocolo con señales independientes.

#### 6. Protocolo distribuido

En estos protocolos la responsabilidad del arbitraje no reside en una unidad independiente, sino que se distribuye por los diferentes masters conectados al bus.

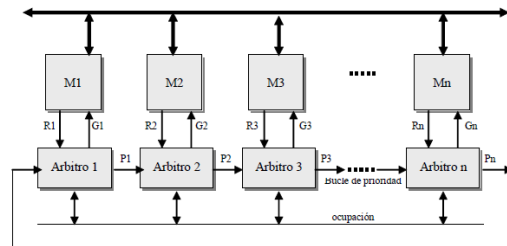


Fig. 21. Diagrama Protocolo distribuido.

- Arbitro-I concede el bus al *master* Mi activando Gi si:
  - Mi ha activado su línea de petición de bus Ri.
  - La línea de ocupación está desactivada.
  - La línea de entrada de prioridad Pi-1 está activada.
- Si el *master* Mino ha activado su línea de petición de bus Ri, el Arbitro-i active la línea de salida de prioridad Pi.

### XI. EJEMPLOS

#### 1. Conexión de periféricos al computador.

Los periféricos se conectan a la CPU a través de buses. En el interior del computador el bus transmite la información de los datos en paralelo.

El bus que conecta la CPU con los otros elementos del procesador se conoce como **bus local** o bus de la CPU. Es un bus muy rápido y conecta la CPU con las tarjetas de la placa base y los controladores de los dispositivos externos.

Las conexiones entre los periféricos y los controladores o tarjetas de la placa base se realizan a través de un bus más general llamado **bus del sistema**. También suele conectar algunas ampliaciones de memoria.

Algunos periféricos requieren un bus especializado que se adapte a su velocidad de transferencia, sus niveles de tensión, la naturaleza de sus señales de control y otros requerimientos. A estos buses se les llama **buses de entrada/salida** o **bus de expansión**.

Todos los buses poseen unas especificaciones normalizadas, como son:

- protocolos de transmisión de datos.
- velocidades y temporización de las transferencias
- anchuras de los sub-buses

- sistema físico de conexión (conectores estandarizados).

Ejemplos de buses normalizados:

- S-100 Bus (IEEE 696)
- CAMAC (o IEEE 583) (“Computer Automated Measurement and Control”)
- GPIB (o IEEE 488) (“General Purpose Interface Bus”)
- Multibus (o IEEE 796)
- ISA Bus (“Industrial Standard Architecture”)
- ISA AT Bus
- MCA (“Micro-Channel Architecture”)
- EISA (“Extendet Industry Satandard Architecture”)
- SCSI (“Small computer System Interface”)
- Futurebus+ (IEEE 896.1 e IEEE 896.2)
- USB o Bus Serie Universal

Ejemplos de buses locales:

- VL-bus, ideado por VESA para varios fabricantes
- PCI (“Peripheral Component Interconnected”), creado para Intel
- Quick-ring, propuesto por Apple
- VME, IEEE 1014 (“Versa Module Europe”)
- NuBus, o IEEE 1196
- M-bus y S-Bus, utilizados por las estaciones de SUN.

2. Ejemplo buses asíncronos.

El dispositivo X posee una entrada /CS (chip select) que lo habilita. Entonces /CS= /DirMem + /AS, donde /DirMem es la combinación de líneas A23 - A0 que refieren a su dirección en el mapa de memoria.

/DTACK: Data ACKnowledge - Reconocimiento de datos  
Línea de entrada, activa a nivel bajo.

Mediante la activación de esta línea, el dispositivo direccionado indica que la transferencia del dato se ha efectuado correctamente.

En un ciclo de lectura, al activarse /DTACK, el uP captura el dato del bus de datos mediante un latch y seguidamente termina el ciclo.

En un ciclo de escritura, la activación de /DTACK indica al uP que el dispositivo externo ha capturado correctamente el dato y acto seguido termina el ciclo.

### **Funcionamiento:**

El control del bus asíncrono se puede explicar de la siguiente forma:

1.- El microprocesador valida la dirección colocada en el bus de direcciones activando la línea /AS (=0).

2.- Se efectúa el ciclo, donde transcurre la transferencia.

3.- El dispositivo direccionado activa /DTACK (=0) indicando el fin de la transferencia, por lo que el uP desactiva /AS (=1) y concluye el ciclo de bus.

Esta forma de comunicación permite ajustar la velocidad de trabajo entre el uP y cada dispositivo conectado al sistema, puesto que estos no tienen que tener la misma velocidad de proceso.

## XII. CONCLUSIONES

- Se puede concluir que las redes y buses de entrada y salida son casi siempre estandarizadas, ya que deben cumplir unas normas estándar para realizar su cumplimiento con los resultados esperados, las cuales se establecen por medio de protocolos.
- Se concluye que las redes y buses de entrada y salida no siempre son síncronas, ya que se pueden ver redes y buses con protocolos de transferencia de manera asíncrona, semisíncronas y de ciclo partido.
- Por medio de todos los protocolos establecidos, se maneja una gran variedad de redes y buses de entrada y salida que permite trabajar con diferentes arquitecturas y mejorar las velocidades y temporización de las transferencias.
- Si se conectan un gran número de dispositivos al bus, las prestaciones pueden disminuir, por tanto, es necesario el uso de varios buses que son organizados jerárquicamente.
- El diseño de un bus es muy difícil debido a que la velocidad máxima del bus se ve limitada debido a factores físicos como la longitud y el número de dispositivos que se conectan.
- Gracias al desarrollo tecnológico del bus de E/S o de expansión surgieron las interconexiones punto a punto lo cual redujo el tráfico en el bus del sistema, la existencia de este bus permitió diseñar una amplia gama de controladores de periféricos compatibles hoy en día, resolviendo el problema de ejecutar muchos cables paralelos a alta velocidad.

## XII. REFERENCIAS

- [1] William Stallings. (2005) Organización y Arquitectura de Computadores – 7º edición, Madrid. pp.77-83.
- [2] Santiago Cristobal Pérez, Higinio Facchini, Daniel M. Argüello. (2015) Arquitectura de computadoras, Argentina. pp.110-111.
- [3] David A. Patterson, John L. Hennessy. Computer Organization and Design, *The hardware / Software interface* pp. 582-586.
- [4] <https://www.ecured.cu/BUS>
- [5] Ampliación de Estructura de Computadores - 3º de Ingeniería Informática Documento PDF.
- [6] <http://www.istr.unican.es/asignaturas/PTR/PTR-TEMA1-3en1.pdf>
- [7] <http://www.upv.es/amiga/275.htm#0>
- [8] <http://www.upv.es/amiga/278.htm>
- [9] M. Young, *The Technical Writer's Handbook*. Mill Valley, CA: University Science, 1989.
- [10] Estructura de Computadores, Facultad de Informática – UCM Documento PDF.
- [11] <http://www.areatecnologia.com/informatica/tipos-de-buses.html>
- [12] <http://computadoras.about.com/od/conoce-procesadores/a/Que-Son-Fsb-Hypertransport-Dmi-Qpi-Y-Fdi.htm>
- [13] Morris Pearse Education 1994 Arquitectura de computadoras – 3º edición, Mexico.
- [14] <http://www.fdi.ucm.es/profesor/jjruz/WEB2/Temas/EC10.pdf>