

Tema 9. Requisitos Hardware para los Sistemas en Tiempo Real

9.1 Introducción

En este capítulo se pretenden sentar algunas ideas básicas acerca del hardware adecuado para la implementación de sistemas en tiempo real. Se supone que el lector ya tiene conocimientos de Estructura de Ordenadores por lo que en ocasiones se mencionarán ciertos temas sin entrar a dar más explicaciones. Este capítulo se podría estructurar según los siguientes apartados:

- Descripción de las características básicas de los microprocesadores.
- Descripción de las técnicas de interface standard.
- Repaso a las técnicas de comunicación standard.

Como se puede comprender será imposible dar una explicación profunda de estos temas, por lo que simplemente se verá una descripción funcional, haciendo más hincapié en las características más apropiadas para el ser.

9.2 Evolución de la arquitectura de procesadores

Antes de comenzar a ver distintos tipos de arquitecturas de procesadores convendría definir este concepto. Se podría acudir a muchas definiciones, como la citada en[1]: “La arquitectura de un procesador es la interface semántica entre software y hardware”. Dejando aparte las definiciones debe quedar claro que el concepto de arquitectura engloba temas como:

- Juego de instrucciones.
- Modelo de registros.
- Tipos de datos y formatos.
- Modelo de memoria.
- Manejo de excepciones.
- Manejo de interrupciones.

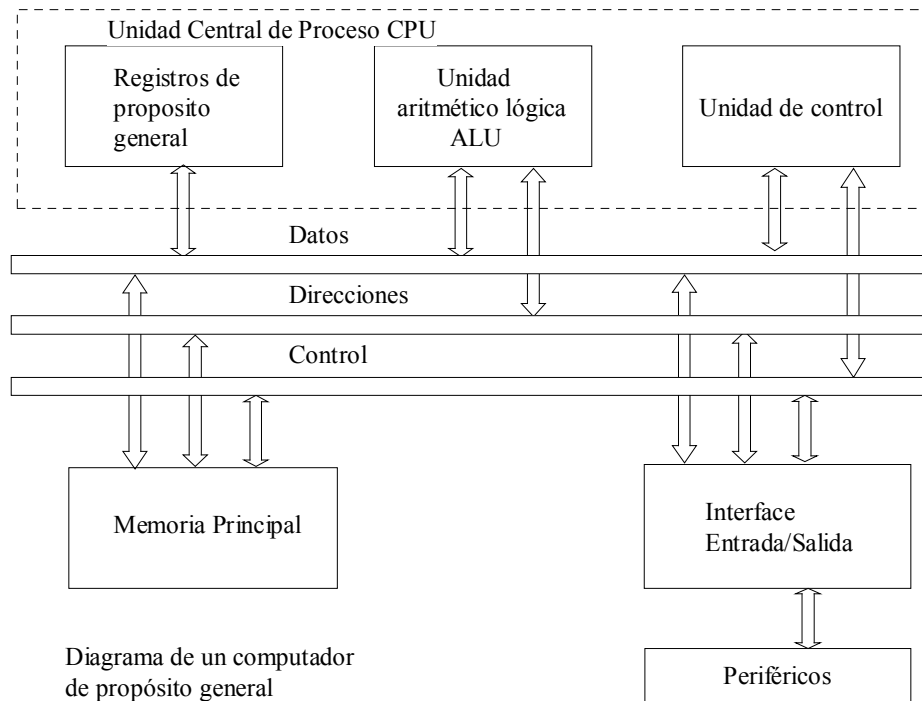
La evolución de la microelectrónica, con las tecnologías VLSI, han permitido aplicar a los procesadores los conceptos que previamente se habían diseñado para los minicomputadores (memoria cache, memoria virtual, manejo de excepciones etc.). Además el incremento de prestaciones de estos microprocesadores ha sido muy veloz, y aún se espera que continúe siéndolo. Es, por tanto, muy factible el desarrollo de str en sistemas basados en microprocesador.

9.2.1 Computador de propósito general

Este computador sigue el esquema clásico definido por Von Neumann, con una unidad central, otra de almacenamiento, y finalmente, una unidad de entrada/salida. Todas estas unidades funcionales se comunican por medio de los buses: de datos, direcciones y de control. Una característica de estos computadores es que son modulares: se proveen los medios de añadir unidades extra, en particular, dispositivos especializados para la entrada/salida. Las capacidades de la unidad básica en términos de potencia de procesamiento, capacidad de almacenamiento, ancho de banda en las operaciones de E/S y estructura de las interrupciones determinan las prestaciones globales del sistema.

De igual importancia en un computador de control resultan los canales de E/S, que permiten la conexión de los dispositivos de medida y actuación con el computador. No hay que olvidar que tales dispositivos no se conectan nunca directamente, sino que se hace por medio de ciertas unidades que sirven de interface.

Un esquema simplificado de los bloques funcionales descritos en el modelo de Von Neumann es el que puede verse en la siguiente figura.



9.2.1.1 Unidad central de procesos CPU

Formada por los registros, la unidad de control y la unidad aritmético lógica (ALU). Constituye el núcleo del computador, sus funciones son de sobra conocidas. Recordaremos las características de una CPU que determinan su potencia de funcionamiento:

- Longitud de palabra.
- Juego de instrucciones.
- Métodos de direccionamiento.
- Número de registros.
- Velocidad del reloj del sistema.
- Estructura de interrupciones.

En los diseños clásicos se hizo hincapié en dotar a la CPU de un juego de instrucciones y direccionamiento amplio y variado. Se posibilitó hacer operaciones en la ALU con operandos depositados en memoria, acceder a los datos con direccionamientos indexados (incluidos autoincrementos/decrementos), acceder a datos de diferentes longitudes, e incluso el hacer varias operaciones con una única instrucción. Se trató de reducir el número de instrucciones precisas para realizar una operación.

9.2.1.2 Almacenamiento

Podríamos dividirlo en rápido, correspondiente a la memoria, y lento, unidades de disco o cinta. El principal es desde luego la memoria, con sus variedades de RAM, ROM, PROM y EPROM. La principal, y con mayor tamaño, de las memorias es la RAM. En ella se pueden albergar tanto datos como instrucciones, pudiendo cambiarse la información contenida en ella tantas veces como se desee de manera instantánea. Por el contrario la misión de las memorias ROM, PROM y EPROM es albergar datos del sistema no volátiles, o funciones predefinidas del computador, que deben permanecer aún cuando se desconecte la alimentación del computador. Estas memorias de solo lectura, que pasan muchas veces desapercibidas para los usuarios convencionales de los computadores, son de gran importancia a la hora de diseñar sistemas de control empotrados. En efecto, en ellas se deberá depositar un programa de arranque que ponga en marcha el sistema. Generalmente albergarán las instrucciones, mientras que el uso de la memoria RAM se destina a los datos. Como las memorias de solo lectura son de acceso más lento que las

RAM cabe la posibilidad de, una vez arrancado el sistema, hacer una copia de las instrucciones de la ROM en la memoria RAM.

El uso de dispositivos secundarios será necesario cuando se precise disponer de grandes cantidades de información. La razón de no usar memoria RAM es doble, en primer lugar su carácter volátil y en segundo lugar su coste mucho más elevado que las unidades de discos o cintas. La desventaja que ofrecen estos dispositivos frente a la memoria central es su velocidad de acceso mucho más baja.

Cuando se lee o escriben datos en un disco se realizan transferencias de información entre la memoria central y esa unidad de almacenamiento. Si este trabajo de transferencia recayera sobre la CPU obtendríamos un bajo rendimiento en su capacidad de procesamiento. Por esta razón se emplean controladores para el acceso directo a memoria (DMA). Esta unidad puede tomar el control de los buses de datos y direcciones aprovechando instantes en que la CPU no los utiliza.

9.2.1.3 Entrada/Salida

La interfaz de E/S es una de las áreas más complejas, y donde más se puede influir desde el punto de vista del diseñador del sistema. Buena parte de esta complejidad proviene de la amplia variedad de dispositivos que se pueden conectar. Esta variedad se manifiesta tanto en las velocidades de transmisión como en su forma de hacerlo. Así se pueden encontrar transmisiones series (síncronas o asíncronas), paralelas, conversiones analógico digital y digital analógico, conversiones a ancho de pulsos, transmisiones vía red e incluso por radio. Además, todos estos dispositivos pueden funcionar incluso con diferentes niveles de tensión, y trabajar por muestreo (polling) o por interrupción.

9.2.1.4 Estructura del bus

Los buses son los elementos que enlazan los componentes de un computador. La caracterización de un bus se puede enfocar desde tres puntos de estudio.

- Estructura mecánica.
- Estructura eléctrica.
- Estudio funcional del bus.

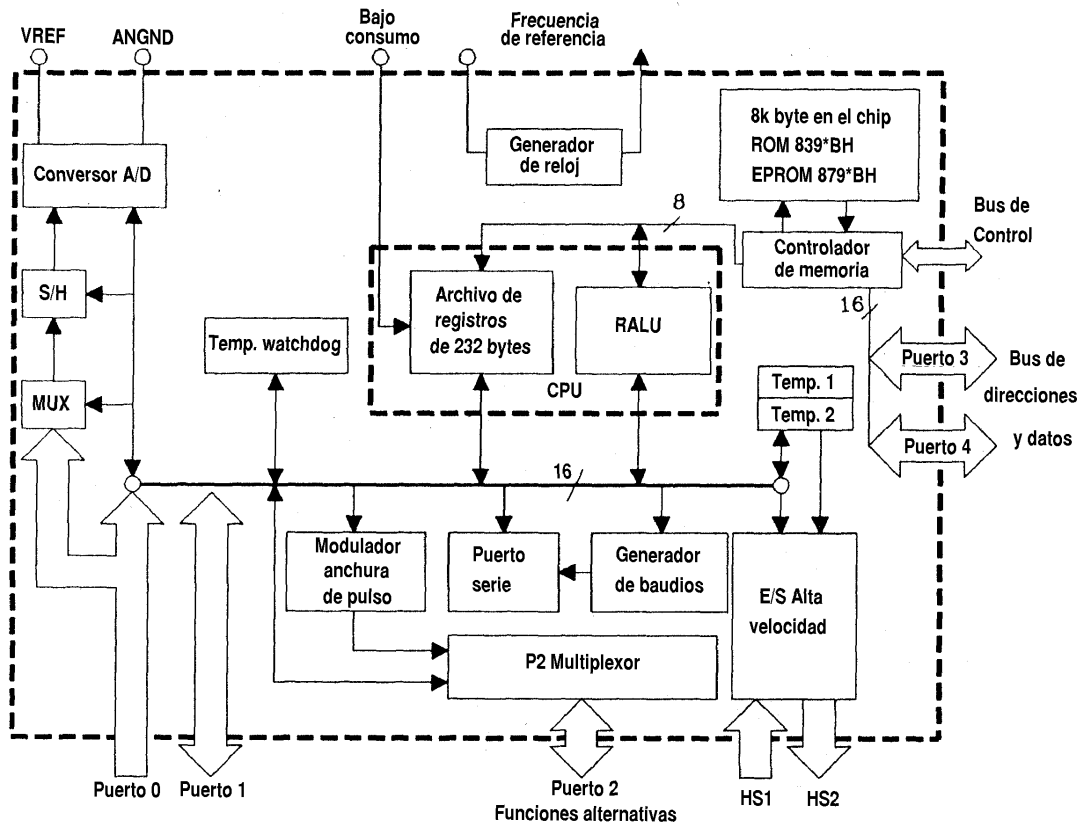
Como sabemos, el bus se subdivide en tres buses, datos, direcciones y control. Aunque en general, a la hora de pensar en las prestaciones de un equipo la principal atención recae en la velocidad del procesador o en la cantidad de memoria de la que dispone, el bus es quizás un elemento aún más determinante. En muchas ocasiones constituyen el verdadero cuello de botella del sistema: Un bus incorrecto o anticuado no permite desarrollar al máximo las potencialidades de los nuevos procesadores. Basta pensar, como ejemplo, en los Pentium, con longitudes de palabra de 64 bits, que por razones de coste y compatibilidad siguen trabajando con buses externos (en la placa madre) de 32 bits. Sin salirnos del entorno de los micros de Intel ya conocemos varias especificaciones diferentes de buses que tratan de mejorar, o reemplazar, a la estructura del bus AT. Así conocemos los buses locales (Vesa o PCI), que son una interfaz con el bus ISA, extensiones de éste como el bus EISA, o el Microchannel (que incorporan los PS de IBM) que es un bus totalmente diferente y con mejores prestaciones que el bus AT.

Además el bus define una interface normalizada por medio de la cual pueden interconectarse dispositivos de diversos fabricantes.

Más adelante se dedicará todo un apartado al estudio de los buses.

9.2.2 Microcontroladores.

Muchos fabricantes de circuitos proporcionan, en un único chip, todos los elementos necesarios para el control. Este tipo de dispositivo se conoce como microcontrolador. En ellos aparecen, habitualmente, espacio en memoria RAM y memoria EPROM, una CPU, un oscilador, temporizadores hardware, puertos de comunicación serie, controladores de interrupciones, puertos de entrada/salida digital, conversores A/D y D/A e incluso señaladores de alarmas (watch-dog timer). Como la memoria con la que cuentan suele ser bastante limitada suelen disponer de una extensión del bus para poder conectarse con memorias externas, fuera del propio chip. En la siguiente figura se puede apreciar el diagrama de uno de estos microcontroladores.



Tradicionalmente este tipo de dispositivos se ha venido programando en ensamblador, que por supuesto es distinto para cada fabricante. Incluso en modelos de un mismo fabricante es posible encontrar incompatibilidad entre los ensambladores de los diferentes dispositivos. Desde hace pocos años, y siguiendo las tendencias que hacen énfasis en el software, se han desarrollado compiladores para lenguajes de alto nivel, generalmente C. Esto ha facilitado la programación y acortado los tiempos de diseño y desarrollo.

La forma habitual de programarlos es emplear un circuito de desarrollo, que se conecta a través del puerto de un PC o una estación de trabajo. El programador edita, compila y enlaza el programa desde su estación de trabajo, luego transfiere el código ejecutable al microcontrolador mediante herramientas software proporcionadas por el fabricante y después prueba el funcionamiento del programa.

Cuando se ha llegado a un funcionamiento correcto el código del programa, junto con un programa de arranque, se graban en la memoria EPROM del microcontrolador. La memoria RAM quedará disponible para los datos, en caso de ser insuficiente habrá que suministrar más conectándola con el bus externo.

Estos dispositivos están pensados para disponer, a priori, de todos los elementos necesarios para el control. De esta manera la tarjeta donde se instale el chip solo necesitará disponer de las líneas de alimentación, tierra y poco más. Son por tanto dispositivos muy adecuados para los sistemas empotrados, pero que no ofrecen alta capacidad de procesamiento. En efecto, carecen de unidades aritméticas en coma flotante, su juego de registros suele ser bastante limitado, así como su capacidad de direccionamiento. No suelen disponer de sistemas operativos y desde luego no están pensados para implementar multitarea. Tampoco están pensados para trabajar en red, sus posibilidades de comunicación se limitan a la utilización de los puertos serie. Todas estas carencias no deben hacernos olvidar que, en muchas ocasiones, un microcontrolador puede ser más que suficiente, y que en la vida cotidiana prestan servicio en muchos aparatos de uso cotidiano (lavadoras, hornos, automóviles etc.).

9.2.3 Procesadores especializados

Este tipo de procesadores se ha desarrollado con dos propósitos principales:

- Aplicaciones críticas en cuanto a su seguridad.
- Incremento de la velocidad de computación.

A la hora de diseñar un programa para una aplicación crítica interesa poder verificar formalmente, usando pruebas matemáticas [2], que la lógica del procesador es correcta. Para ello, cuanto más sencillo sea el juego de instrucciones más facilidades de verificación se tendrán.

En cuanto al incremento en la velocidad de computación hay varias vías disponibles, algunas de ellas orientadas a la solución de ciertos tipos de programas.

9.2.3.1 Procesadores RISC (Reduced Instruction Set Computer)

Este tipo de procesadores se basan, como punto de partida fundamental, en que la velocidad se puede incrementar no solo reduciendo el ciclo del reloj, reducción que viene dada por los adelantos en microelectrónica, sino que fundamentalmente se puede lograr reduciendo el número de ciclos de reloj necesarios para ejecutar una instrucción. Con un juego de instrucciones sencillo y reducido se puede tener un microcódigo fácil y rápido de decodificar. Los procesadores RISC han desechado las complejas instrucciones de los CISC (Complex Instruction Set Computer), que estadísticamente se encontró que apenas eran utilizadas por los compiladores. También han desechado realizar operaciones con datos contenidos en memoria orientando las operaciones hacia el modelo de carga/almacenamiento. Es decir, solo se va a memoria para leer un dato y llevarlo a un registro o viceversa.

Suelen ser procesadores de 32 bits, que dividen la palabra de instrucción en campos fijos (posible por la simplicidad en el juego de instrucciones y de direccionamientos). El espacio en silicio liberado al prescindir de microcódigo y de ciertos operadores hardware especializados es utilizado en ampliar el número de registros y proporcionar una memoria caché interna.

Un avance significativo en el aumento de prestaciones de los procesadores en general ha sido el pipeline. La ejecución completa de una instrucción puede dividirse en etapas: carga, decodificación y búsqueda de datos, ejecución y almacenamiento del resultado. Las etapas pueden realizarse en pseudoparalelismo, es decir, mientras se está en la fase de almacenamiento de la instrucción K también se realiza la ejecución de la K+1, la decodificación de la K+2 y la carga de la instrucción K+3. En [4] se puede encontrar un análisis más detallado de las arquitecturas RISC en comparación con las CISC.

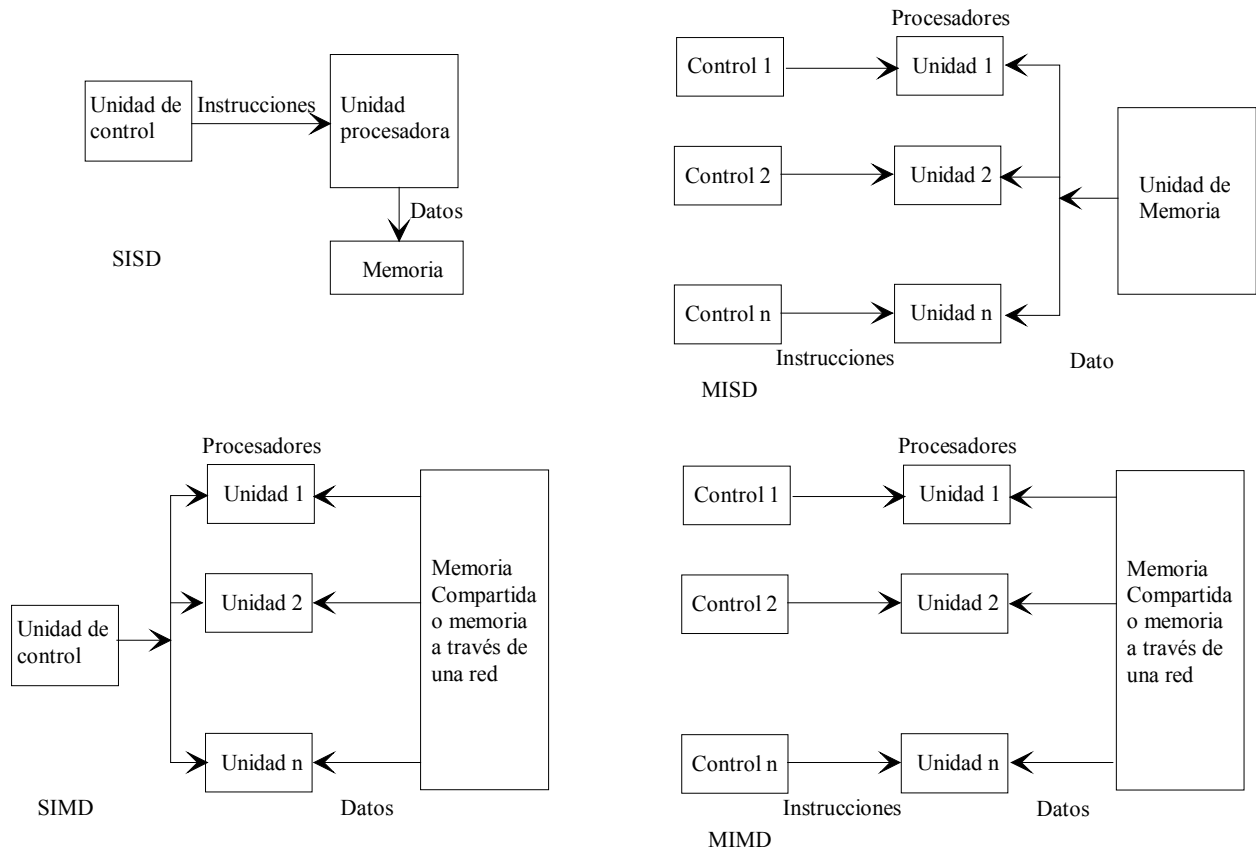
Aunque en el mundo de las estaciones de trabajo la tendencia actual es el uso de estos procesadores, en el campo del control aún no están asentados. Las ideas de la filosofía RISC pueden ser llevadas al control de procesos, pero arrojando a lo que sería el núcleo del procesador con otros elementos, tales como unidades en coma flotante, sistemas multiprocesos, gestión de memoria etc. Ya existen procesadores en este sentido que incorporan nuevas unidades al núcleo central RISC.

9.2.3.2 Computadores paralelos.

Si es posible ejecutar varias instrucciones a la vez, o traer varios datos simultáneamente de la memoria, podrán aumentarse las prestaciones del sistema en cuanto a su velocidad. La arquitectura tradicional utiliza un único camino para datos e instrucciones, su acrónimo es SISD (Single Instrucion stream Single Data stream). En cuanto a los procesadores que implementan paralelismo hay varias categorías:

- SIMD Single Instrucion stream, Multiple Data stream
- MISD Multiple Instrucion stream, Single Data stream
- MIMD Multiple Instrucion stream, Multiple Data stream

Todos ellos, junto con la estructura clásica se ilustran en la siguiente figura.



9.2.3.2.1 Los transputers

En 1978 INMOS Limited produce un nuevo microprocesador que salió a la luz a mediados de los 80 conocido como "transputer". El concepto de transputer es en muchos sentidos nuevo; Aunque la mayor parte de las ideas que utiliza son convencionales. Ideal para aplicaciones que procesen en paralelo, y para aplicaciones que necesiten mucha capacidad de proceso; es el único procesador diseñado especialmente para usar en grandes sistemas distribuidos.

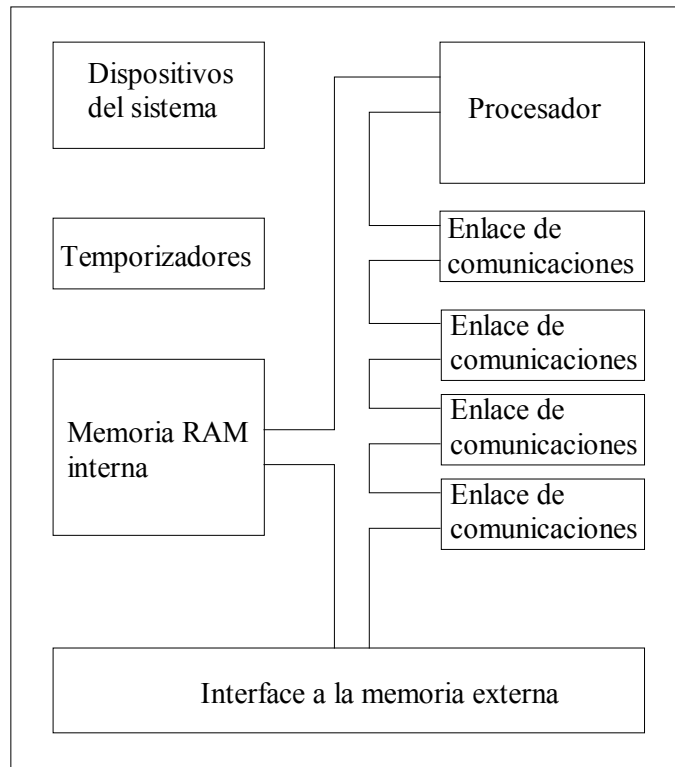
Un transputer consiste en un microprocesador rápido, varios puertos de comunicaciones, una memoria rápida, un interface con memoria externa, timers, relojes y planificador todo en un chip. Algunos tienen también un coprocesador de punto flotante. Los puertos de comunicaciones se llaman "LINKS", y cada transputer suele tener de 2 a 4 links en serie bidireccionales a 20 Mhz / s. Hay transputers de 16 bits, y otros de 32 bits; esto se refiere al ancho de los buses y registros, y se conoce con el nombre de "longitud de palabra" del transputer.

Uno de los objetivos en el diseño de transputers fue hacer el multiproceso muy fácil, y como resultado, tenemos disponibles supercomputadores baratos, contruidos con grandes arrays de transputers. Otras aplicaciones tal como radar y gráficos animados usan grandes arrays de transputers para conseguir una computación muy rápida.

Las diferentes partes del transputer operan concurrentemente. Por ejemplo los enlaces (links) pueden transferir datos a otros procesadores en la red a la vez que el procesador en coma flotante está haciendo cálculos. El retraso de interrupción (interrupt latency) del transputer puede ser menor de 600 ns.

Los transputers pueden emparejarse directamente con otros a través de sus links, se pueden realizar topologías completas sin hardware externo. La construcción del transputer permite al usuario definir su programa como una pequeña colección de procesos comunicantes. En el transputer los cambios de contexto (paso de estar ejecutando un proceso a ejecutar otro distinto) son menores de 1 microsegundo, por ello la sobrecarga de la planificación de un número grande de procesos es baja.

En la siguiente figura se puede apreciar el diagrama de bloques de un transputer; normalmente se dispondrá de varios de ellos formando estructuras enlazadas.



Chip transputer

La sincronización y comunicación entre procesos que van en paralelo se hace a través de canales (channels), que proveen conexiones síncronas punto a punto. La comunicación con procesos en otros transputers es completamente transparente, pues se hace vía links usando en mismo canal y las mismas instrucciones.

TRAM (TRANsputer Modules)- son los módulos hardware estándar de INMOS, pueden contener un transputer con memoria y 16 pines activos, o lógica para un propósito concreto.

CHANNEL- ruta de comunicación entre 2 procesos. Si ambos corren en el mismo transputer se llaman “soft channel” o “internal channel”. Si van en distintos transputers se llaman “hard channel” o “external channel”. Son implementados por el procesador copiando los datos del espacio de datos de un proceso a otro. Un channel es como una comunicación unidireccional, un proceso envía y otro recibe. Si queremos comunicación bidireccional necesitaremos dos channels entre los dos procesos, uno en cada dirección. Para comunicar primero hay que esperar a que el otro esté ready; esto puede provocar una latencia significativa dependiendo de la carga de procesos en el procesador.

No es recomendable que los transputers tengan memoria compartida. La forma correcta de compartir valores es enviar vía link, es más fácil y seguro. De hecho los links pueden usarse como puertos DMA serie que darán acceso directo a la memoria del transputer.

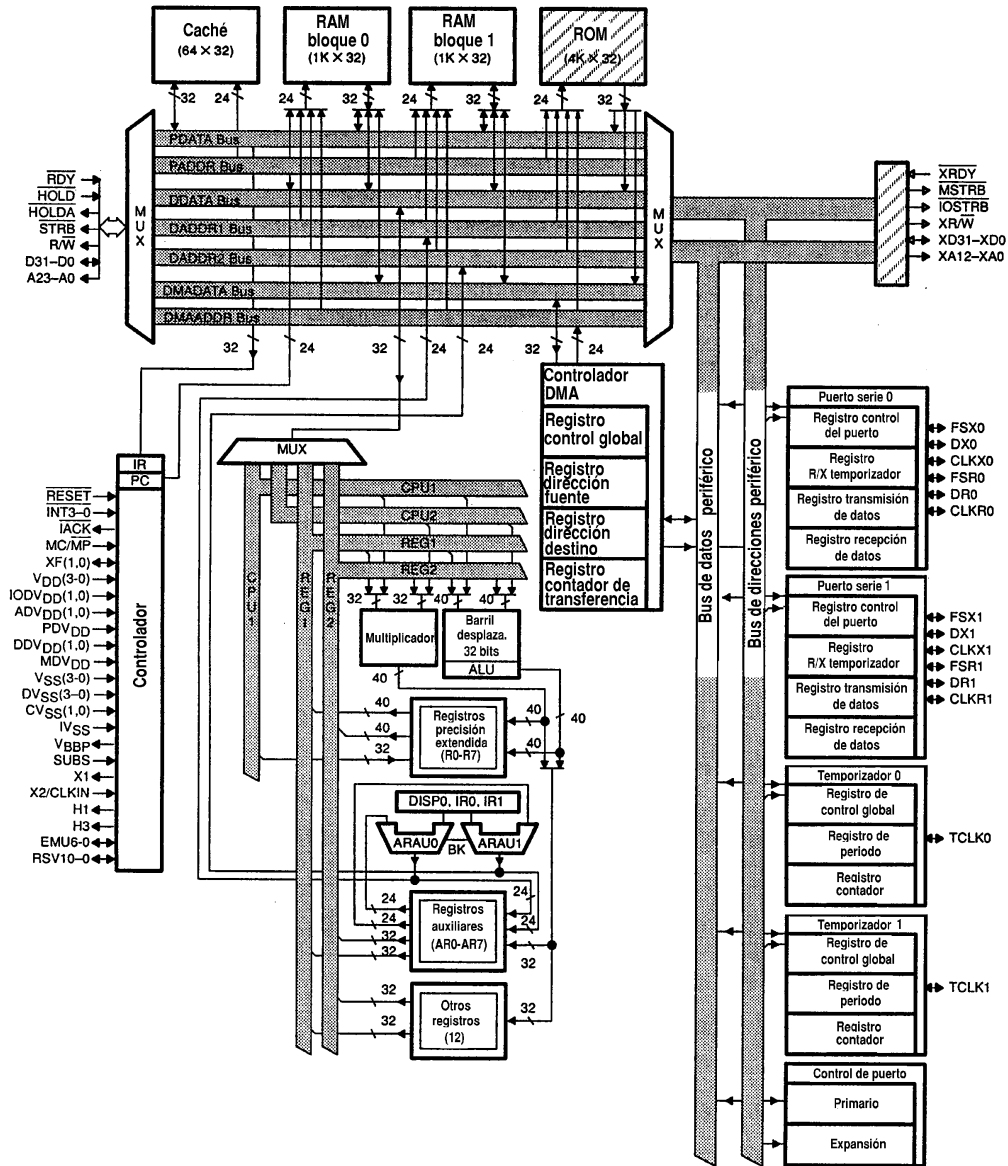
La programación de transputers se realiza en occam, un lenguaje especial diseñado para ellos (mezcla de C y ensamblador). Occam se basa en la suposición de que la aplicación que se va a implementar pueda ser modelada como un conjunto de procesos (acciones) que se comunican con las demás vía canales.

9.2.3.3 Procesadores digitales de señal (DSP)

En aplicaciones tales como el procesamiento de voz, telecomunicaciones, radar, visión artificial y en general, en cualquiera donde se requiera procesamiento digital de señal, pueden obtenerse grandes prestaciones con los DSP's. Se trata de procesadores específicos, orientados en su diseño al trabajo con filtros digitales. Un filtro digital es un algoritmo que toma una secuencia de entrada, la procesa, y la convierte en una secuencia de salida Su algoritmo fundamental viene dado por la ecuación:

$$y(n) = \sum_{k=0}^N h_k * x_{n-k}$$

La arquitectura de este tipo de procesador está orientada a solucionar problemas donde intervienen este tipo de algoritmos, o también los que realizan FFT.



Estructura del procesador digital de señal TMS320C30 de Texas Instrument

Los nuevos DSP's son procesadores de 32 bits, normalmente más en las unidades aritméticas, que implementan operaciones en coma flotante (sumas y productos) mediante rápidos operadores hardware. Para obtener una mayor velocidad de ejecución recurren a la arquitectura Harvard, donde se dispone de buses paralelos para datos y para instrucciones. Al disponer de varios operadores hardware, y de varios caminos de datos, es posible realizar operaciones en paralelo.

Este tipo de procesadores suelen explotar las ventajas del pipeline y algunos de los puntos de la filosofía RISC (instrucciones de longitud fija que se ejecutan en un único ciclo etc.). Incorporan dentro del propio chip memoria RAM, EPROM e incluso memoria caché. También disponen de hardware adicional como temporizadores, puertos serie, controladores de DMA y entradas salidas digitales. A diferencia de los microcontroladores no disponen de convertidores A/D D/A ni watch_dogs, por tanto, no son tan autónomos como estos últimos para realizar tareas de control.

Las últimas tendencias en DSP se encaminan a la interconexión de DSP's de la misma familia. Los constructores disponen patillas para permitir establecer comunicaciones entre ellos según protocolos del fabricante.

Hasta ahora su programación era dificultosa, se realizaba en el ensamblador correspondiente. Ahora ya es fácil que incorporen compiladores de C, e incluso ADA. También se ha desarrollado el sistema operativo SPOX, que corre en plataformas hardware de distintos fabricantes, para facilitar las labores de diseño y la portabilidad de los desarrollos.

9.2.3.4 Redes de computadores

Por último, cabe también la opción de establecer un control distribuido a través de computadores conectados mediante una red local (LAN Local Area Network). En muchas ocasiones hemos utilizado una red, pero básicamente como servidor de archivos. De esta forma un nodo servidor, que ofrece los servicios del sistema operativo, es el que almacena la mayor parte del software, cada estación de trabajo utiliza el disco del servidor, a través de la red, como si fuera un disco local. Con este método de trabajo cada estación de trabajo emplea su propia CPU y memoria, compartiendo solo los archivos.

Las redes permiten funcionamientos más avanzados; así, un control puede dividirse en distintas tareas, que pueden ejecutarse en computadores diferentes que se hayan interconectados a través de la red. Cada una de las tareas, en diferentes CPU's, puede comunicarse con las restantes. Un mecanismo de comunicaciones muy extendido (en UNIX y en Windows) son los sockets.

Hay sistemas operativos, y compiladores, que pueden manejar una red de manera transparente al usuario. La tarea se ejecutaría en distintos procesadores pero el usuario no tendría que preocuparse de ello, él programaría como si dispusiera de un único procesador.

9.3 El reloj de tiempo real

Este dispositivo es fundamental en los sistemas controlados en tiempo real. En la mayor parte de las ocasiones se trata de un oscilador, que genera una onda de frecuencia fija, asociado a un contador.

Una forma de reloj, muy rudimentaria, se basa en la frecuencia de la red eléctrica (50 o 60 Hz), pero lo más frecuente es el uso de un temporizador (timer) hardware. Con este tipo de temporizadores se tiene un oscilador de cuarzo, que genera una onda cuadrada de frecuencia fija, cada flanco de subida (o de bajada) de esa onda decrementa el valor almacenado en un registro contador asociado al oscilador. Cuando el contador llega a cero, se produce una interrupción y el contador se vuelve a recargar automáticamente.

Habitualmente los contadores son programables, se puede cargar en ellos un valor comprendido entre 0 y el máximo que admite. El valor máximo vendrá determinado por la longitud en bits del registro contador (un tamaño clásico es 16 bits). En ocasiones se puede seleccionar la entrada a estos contadores, pudiendo escoger un reloj externo como fuente de pulsos. De esta forma no solo sirven para realizar temporizaciones sino que, además, incluyen la función de contar eventos.

Es frecuente que cada interrupción provoque el incremento de un segundo contador, así, en los PC's que disponen de reloj de tiempo real, cada 55 ms se incrementa un contador. El valor de este contador (que suele ser de 64 bits) sirve para mantener la hora y la fecha, es la base para el suministro de servicios horarios. El inconveniente es que al desconectar la alimentación el valor almacenado en ese registro se pierde, es, por tanto, necesario suministrar alimentación mediante una pequeña batería que garantice el funcionamiento de este reloj aún cuando el sistema esté desconectado.

En un computador orientado al control suele disponerse de más de un temporizador, de esta forma pueden, por ejemplo, establecerse diferentes períodos de muestreo en una planta multivariable. O usar uno de los temporizadores como reloj de tiempo real mientras que el otro funciona como temporizador programable. Otra alternativa es la generación de trenes de pulsos de distintos períodos y duty cycle.

Circuitos como el 8253 permiten una flexibilidad en sus modos de funcionamiento con tres contadores, que pueden generar interrupciones o no, recargarse al llegar a 0, mandar un tren de pulsos o solo uno etc.

9.4 Convertidores Analógicos/Digitales (CAD) y Digitales/Analógicos (CDA)

Dado que el mundo real maneja señales continuas, señales analógicas, mientras que el computador utiliza valores discretos, es necesario establecer una interface entre ambos mundos. Esta interfaz viene dada por los CAD y CDA.

Hay varias posibilidades para la representación de un número de forma binaria. Una forma sería la adopción del binario puro (straight binary). Así, para representar el valor 6 V con 10 bits (0 a 1023), que representan un rango de escala entre 0 y 10 v se haría la siguiente operación: $6 \cdot 1024/10$ y se trunca o redondea a un entero.

Esta forma de representación es especialmente apropiada para valores de tensión unipolares (entre 0 y X). Cuando los valores son bipolares, con signo, se adoptan otros formatos de representación, como por ejemplo el offset binary. Si el rango de un convertidor es -5 a 5v y dispone de 10 bits, ahora el 0 representa -5 y 1023 representa v. Sea una señal de 3 v, su representación se obtiene con $(3 - (-5)) \cdot 1024/10$ y truncando o redondeando a entero.

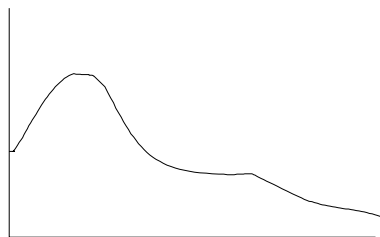
Otra forma común de representación es el complemento a 2. En algunos conversores es posible escoger entre una forma de representación u otra, en la mayor parte de las ocasiones viene ya impuesta

9.4.1 Convertidores Analógicos/Digitales (CAD)

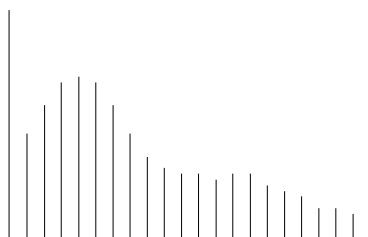
La conversión de una señal continua a discreta implica dos procesos, muestreo y cuantificación. Muchos CAD incluyen un circuito conocido como sample-and-hold. La misión de este circuito es retener un valor continuo para que luego sea pasado a un valor discreto. Su misión es mantener un valor constante mientras que por algún método se va obteniendo el valor digital equivalente (cuantificación) de la señal continua.

Para comenzar con el proceso de digitalización se envía una breve señal, de 1 μ s, al sampler. Este circuito toma la señal que está a su entrada y la traslada a su salida manteniéndola constante. Esto consume un tiempo de unos 0.2 a 0.3 μ s. A partir de este momento comienza el proceso de cuantificación, que puede consumir desde unos pocos microsegundos a varios milisegundos.

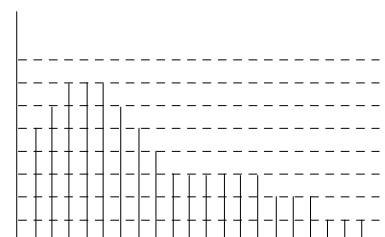
La señal retenida por el sampler-and-hold es pasada a un valor discreto. Ahora se produce un cambio en la naturaleza de la señal. No solamente se pasa de tener una señal continua en el tiempo a una señal discreta, sino que también se pasa de una señal, que puede tomar cualquier valor, a otra donde solo es posible un conjunto de valores. Toda conversión de una señal continua a discreta trae, por tanto, un proceso de aproximación.



Señal continua y de amplitud continua



Señal discreta y de amplitud continua



Señal discreta y de amplitud discreta

Para, a partir de la señal continua, obtener la digital, existen varias técnicas. De todas ellas quizás la más frecuente sea la de aproximaciones sucesivas. Supongamos que se tiene un convertidor de 10 bits (0 a 1023) para transformar señales comprendidas entre 0 y 10 V. Si la señal de entrada por ejemplo es de 6 V, esta señal de entrada se compara con un valor de test de la mitad de la escala del convertidor, en este caso 5 V. Como la señal es mayor que la señal de test el bit más significativo se pone a 1 (1XXXXXXXXX).

Ahora se añade a la señal de test una correspondiente a un cuarto de la escala (2.5 V), la nueva señal de test será 7.5 V. Se realiza de nuevo la comparación, ahora resulta que la señal de entrada es menor, esto implica un 0 en el siguiente bit (10XXXXXXXXX). Ahora se resta un octavo de la escala (1.25 v) a la señal de test que pasa a 6.25 V y se prosigue con el proceso.

Este método de transformación está muy extendido pero es lento; existen métodos que ofrecen mayores prestaciones pero a un mayor coste.

En la teoría del tratamiento digital de señal, procedente de un muestreo y, por tanto, con una conversión A/D, se suele obviar este aspecto. De esta manera se simplifica la teoría matemática, pudiendo aplicarse la correspondiente a los sistemas lineales. En muchas ocasiones el error de cuantificación cometido no es significativo y puede asumirse esta forma de trabajo. Cabe añadir que el mero hecho de trabajar con un computador digital introduce errores de cuantificación al realizar operaciones aritméticas y al representar los números reales, errores que habitualmente se desprecian, ya que suelen ser bastante menores que los asumidos por la conversión A/D. En [4] se pueden encontrar más información acerca de los problemas de la cuantificación.

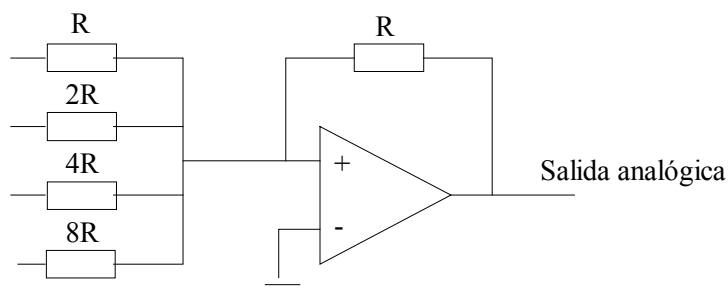
Las tarjetas comerciales existentes para la CAD suelen disponer de varios canales de entrada para poder convertir distintas señales. En lugar de disponer de un CAD para cada canal de entrada se suele tener un único CAD multiplexado para todas las entradas. De nuevo las razones hay que buscarlas en el coste. Es posible, desde luego, desarrollar aplicaciones a medida, diseñando nuestra propia placa, escogiendo los conversores sin necesidad de adquirir una tarjeta comercial, que suele ir conectada al bus de un computador.

A la hora de escoger una tarjeta u otra para realizar la conversión se deberían tener en cuenta los siguientes aspectos:

- Número de entradas analógicas, habitualmente más de 8.
- Si son multiplexadas o no (lo más frecuente es que lo sean).
- Tiempos de conversión, lo que determina la frecuencia máxima de muestreo.
- Precisión (dada por el número de bits del convertidor lo normal oscila entre 8 y 14).
- Cual es el nivel de corriente de las entradas (puede ir de unos nA a varios mA).
- Rango de tensiones. Pueden ser bipolares -5 a 5 v o también unipolares 0 a 10 v.
- Entradas diferenciales o no diferenciales (single ended).

9.4.2 Convertidores Digitales/Analógicos (CDA)

Realizan el proceso contrario de los anteriores. Su funcionamiento es más simple que el de los CAD por lo que es frecuente disponer de un convertidor para cada canal de salida, aunque el número de canales de salida suele ser más limitado (entre 1 y 4) en las tarjetas comerciales. Básicamente funcionan sobre la base de un registro de salida y un sumador. En la siguiente figura se ve un esquema para un convertidor de 4 bits.

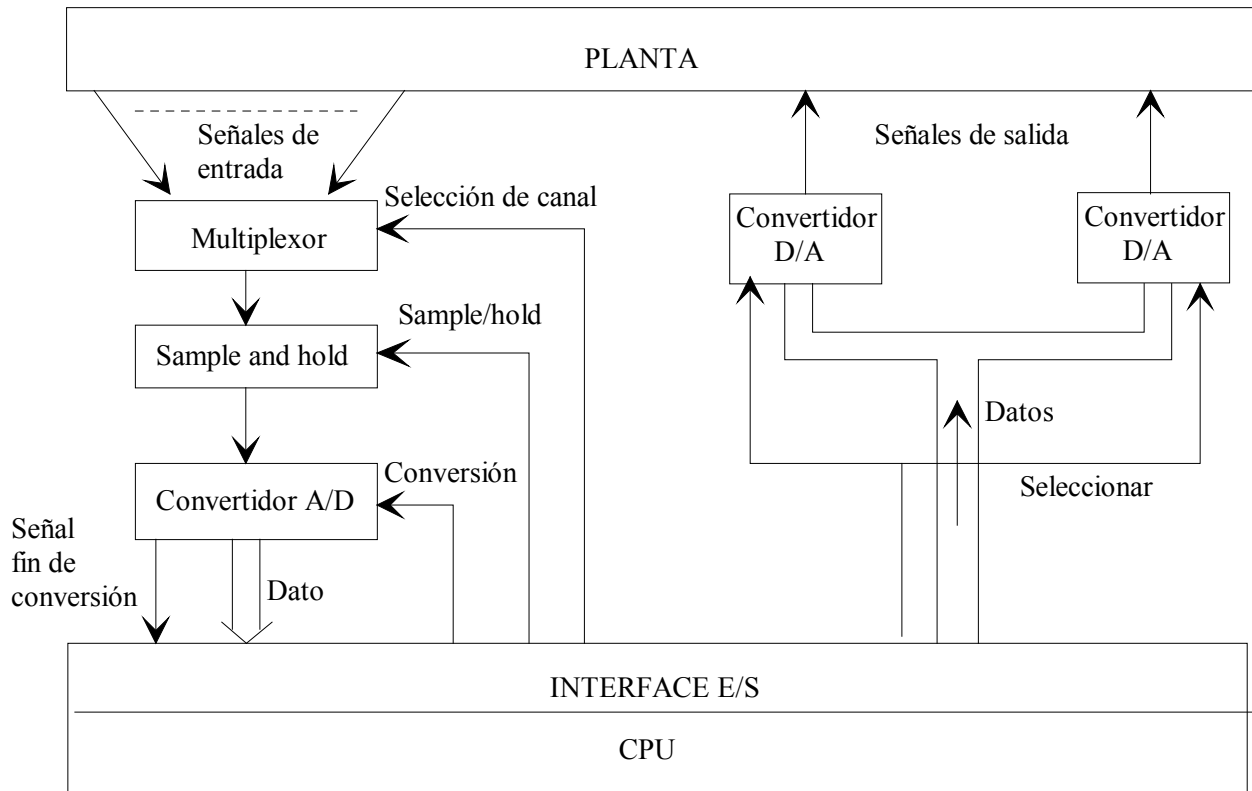


El CDA actúa como un latch, manteniendo el valor que la ha sido introducido hasta que el registro de salida se cargue con un nuevo valor.

En cuanto a elementos principales a destacar en una tarjeta para la conversión D/A se pueden enumerar:

- Número de canales.
- Resolución.
- Niveles de tensión en la salida.
- Velocidad de conversión.

En las tarjetas comerciales suele disponerse de las funciones A/D y D/A conjuntamente, en ocasiones también se dispone de elementos de E/S digital. Ahora puede apreciarse un esquema de bloques de una tarjeta A/D D/A.



Esquema de tarjeta convertora A/D y D/A

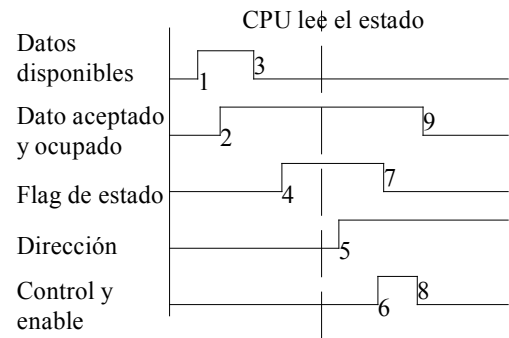
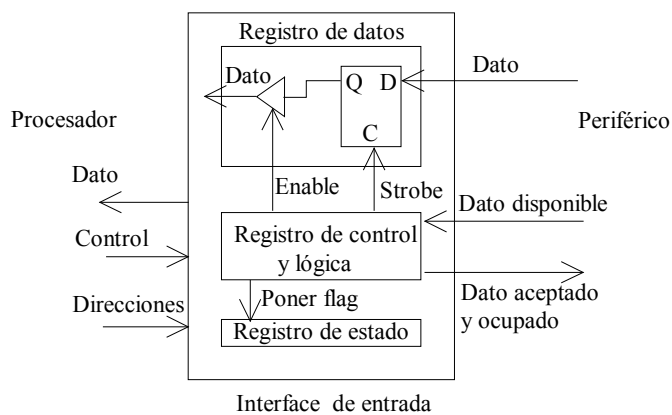
La forma usual de trabajar con una de estas tarjetas desde un computador es escribir en sus registros de control, que estarán mapeados dentro del espacio de direcciones de E/S. Con estos registros se pueden especificar los canales a utilizar o el modo de conversión deseada (diferencial o no, modo de representación de la información), cuando realizar las conversiones, etc. Por lo general, en una tarea de control, se realizará una conversión cada período de muestreo, por tanto, el inicio de la conversión suele estar asociado a una rutina de interrupción ligada a un timer. En dicha rutina, y de acuerdo a las necesidades del control, se seleccionan el canal o canales de entrada. Una vez finalizada la conversión, cuando el dato/s está disponible en un registro de datos de entrada de la tarjeta, aparece una señal que indica tal hecho. En algunas ocasiones esta señal se maneja por muestreo (polling) examinando un bit de un registro de estado de la tarjeta, pero lo más frecuente es que esta señal produzca una interrupción hardware. De cualquiera de las dos formas a partir de este momento puede ejecutarse el algoritmo de control. Una vez finalizado éste, cuando las salidas estén dispuestas, solo resta colocar los valores en los registros de salida para que se proceda a su conversión.

9.5 Interfaces Digitales

En muchas ocasiones no es necesario hacer una conversión de señal continua a discreta o viceversa. Por ejemplo, en el caso de sensores o accionadores todo o nada podemos actuar con ellos como si de señales digitales se trataran. Será necesario, en primer lugar, una adaptación de los niveles de tensión de funcionamiento del captador o actuador a los niveles manejados por el computador (0 a 5 v). Acto seguido se está en disposición de conectarse al puerto digital de E/S.

Los puertos digitales agrupan sus bits de E/S en forma de registros, donde cada componente es un buffer triestado. Este buffer mantiene un valor, que solo deposita en el bus cuando es activado por compuerta, en caso contrario queda en estado de alta impedancia. Este esquema no es válido en el caso de que los datos externos estén disponibles solo durante un breve lapso de tiempo; en tal caso es necesario usar una señal de control desde el dispositivo externo para almacenar el valor en un registro contenido en la interfaz. En este caso la señal de entrada es "latched" (retenida) en un registro interno de la interface por una señal de "strobe" (enclavamiento). Un strobe es un pulso aplicado al reloj de entrada del latch (como en los registros tipo D). La transición de ese pulso (flanco de subida o de bajada) obliga al latch a tomar el valor que se encuentra en su entrada. En la siguiente figura puede verse

una interfaz de entrada junto con un esquema del “handshaking” (secuencia usada para la petición, permiso y transferencia).



Handshaking de un interfaz de entrada digital

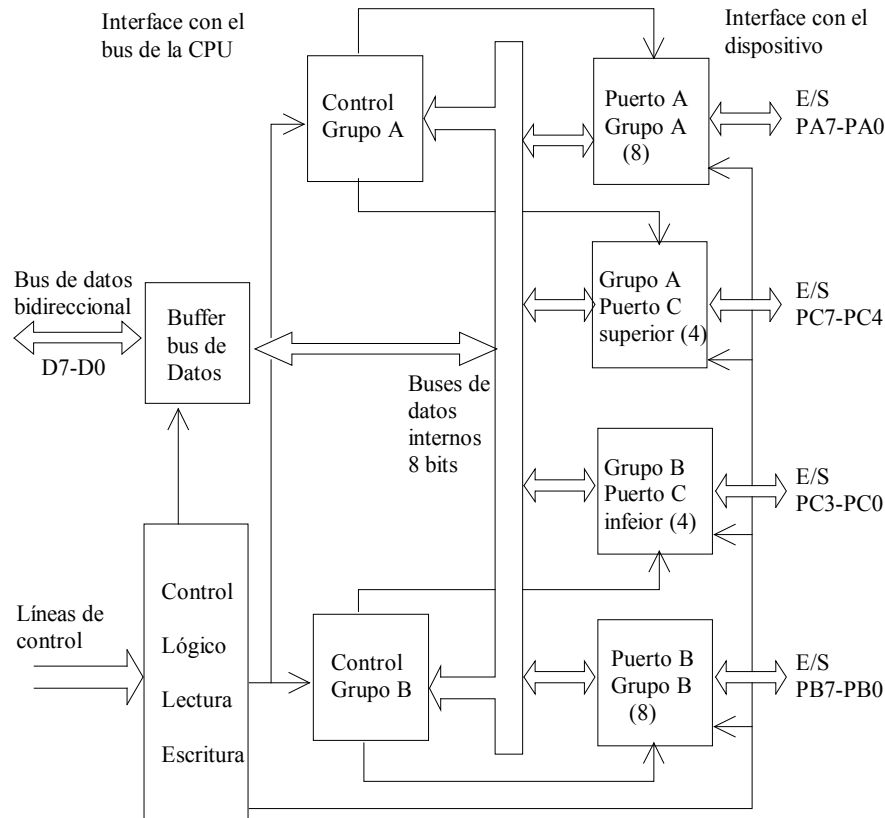
El proceso comienza con el periférico indicando al computador que dispone de un dato (1). La interfaz responde con dato aceptado y ocupado (2), que no bajará hasta que el procesador lea el dato de la interfaz. El dato es enclavado en la interfaz cuando el periférico baja la línea de dato disponible (3). Aproximadamente al mismo tiempo se envía un bit de “ready” (preparado) al registro de estado, para indicar a la CPU que todo está dispuesto. (4). Para conocer este evento la CPU debe estar en un proceso de “polling” (muestreo) de ese registro de estado; en el esquema adjunto no se muestra este tiempo. Cuando detecta el flag a uno lee el registro de datos (5 y 6), lo cual resetea el flag de estado (7) y finalmente resetea la línea de dato aceptado y ocupado (9) después de que la CPU haya retenido el dato (8). La bajada de la línea 9 informa al periférico de que la interfaz está dispuesta para aceptar nuevos datos.

Las interfaces de salida son más sencillas, constan de un registro de datos donde la CPU puede depositar datos en el momento que considere oportuno. Los datos permanecen en el registro hasta que la CPU decida cambiarlos.

Estas interfaces que se han descrito son solo algunas de las varias posibles. Hay ocasiones donde al disponer en la interfaz de entrada de un dato disponible se activa una interrupción hardware; de esta forma no es preciso realizar un polling sobre los bits del registro de estado para detectar si hay dato disponible.

Al igual que las tarjetas conversoras, también es posible encontrar tarjetas con chips que incorporen tanto las funciones de entrada como de salida. Incluso las tarjetas comerciales para la conversión A/D y D/A suelen incorporar algún puerto de E/S digital.

Para finalizar con este apartado se muestra el chip que monta el puerto paralelo de un PC, el PPI 8255 de Intel; dicho chip también se encuentra en muchos otros dispositivos. A grandes rasgos consta de tres puertos de E/S de 8 bits, denominados como A, B y C. Las 8 líneas de C se dividen en dos grupos de 4, que pueden trabajar combinadamente con los puertos A y B. La programación del 8255 es muy flexible y admite 3 modos de funcionamiento. Con estos modos se puede determinar que líneas funcionarán como entradas y cuales lo harán como salidas, así como las que servirán como señales de control con el periférico para establecer un diálogo. Para obtener mayor información acerca del funcionamiento de este chip puede consultarse [5].



Estructura del puerto programable 8255 de Intel

9.6 Técnicas de transferencia de datos

A la hora de la transferencia de datos son varios los factores que se pueden considerar:

- Longitud de la palabra de datos.
- Protocolo de comunicaciones entre los dispositivos.
- Sincronización para iniciar las operaciones.
- Velocidad de la transferencia.

Estos conceptos ya deben ser conocidos en las asignaturas de Estructura de Computadoras e Ingeniería de Computadores por lo que aquí solo se hará una referencia a ellos.

Atendiendo a la longitud de los datos la transferencia se puede realizar en serie (bit a bit) o en paralelo (lo más normal es que emplee palabras de 8 bits). En el caso de ser en paralelo se empleará una interfaz de E/S digital similar al descrito anteriormente. En el caso serie lo más frecuente es la utilización de circuitos UART (Universal Asynchronous Receiver Transmisor). El funcionamiento de estas UART ya fue visto en Tecnología Electrónica de los Computadores.

La comunicación serie se puede realizar de manera síncrona, con una línea de reloj común entre los dispositivos, o asíncrona, estableciendo señales de listo-acepta si se desea. En este caso, al no haber reloj común, es necesario el uso de un protocolo para que los datos no resulten ambiguos. Será preciso especificar de alguna manera el tipo de caracteres que se enviarán (ASCII u otros), la longitud de los datos, número de bits de stop, si se usa paridad (par o impar) o no, la velocidad con que se transfiere (baudios) etc.

El empleo de la comunicación serie resulta adecuado cuando no se precisan de altas velocidades de transferencia o bien cuando los elementos a comunicar están distanciados varios metros. Hay muchos dispositivos (actuadores) que pueden incorporar puertos series para labores de monitorización, así se puede informar al computador cada pocos segundos (o décimas) del estado del actuador. Los dispositivos más elementales no suelen contar con circuitos UART.

Un punto fundamental en la comunicación entre CPU y dispositivos es como indicar a la CPU la presencia de un dato disponible o la aparición de un evento. Anteriormente se mencionó la técnica del polling para detectar la puesta a uno de un flag; esta técnica es muy deficiente pues hace necesaria una espera ocupada por parte de la CPU. Mucho más conveniente es el empleo de interrupciones hardware. Como es sabido, al producirse una interrupción la CPU finaliza con la instrucción que estaba ejecutando, salva el contexto (el contador de programa y en ocasiones el registro de estado del computador) y ejecuta una rutina asociada a la interrupción que se ha producido. Cuando finaliza con esta rutina de tratamiento vuelve al punto del programa donde había abandonado la ejecución para atender a la interrupción.

El periférico dispone de una línea para la solicitud de interrupciones, otra para recibir una señal de reconocimiento y las conexiones con el bus de datos (en el caso de que sea necesario, lo más frecuente es que sí). En caso de ser necesario un periférico puede identificarse enviando una dirección (o vector) a través del bus de datos.

El número de líneas de interrupción hardware internas de los chips de los procesadores suele ser bastante limitado, por lo que a veces se dispone de circuitos específicos, como el 8259 de Intel, para el manejo de las interrupciones. Tales circuitos suelen disponer de un esquema de prioridades para dar preferencia a un dispositivo o a otro. Los controladores de interrupciones, como sucede con el resto de los dispositivos, se pueden programar desde el espacio de direcciones de E/S que implemente el computador. Dicho espacio de direcciones puede estar mapeado igual que la memoria (caso de los micros de Motorola) o en un espacio paralelo (caso de Intel).

Las interrupciones hardware pueden dividirse en enmascarables (las más frecuentes) y no enmascarables (NMI). Habitualmente los micros incorporan instrucciones para habilitar o deshabilitar todas las interrupciones; esto queda reflejado en alguno de los bits de la palabra de estado. En el caso de interrupciones enmascarables es posible deshabilitar selectivamente la atención a los dispositivos periféricos que no nos interese que sea atendidos por alguna razón en concreto, generalmente porque se está atendiendo a otra rutina de interrupción que puede resultar crítico detener. El sistema debe ser capaz de recordar las interrupciones que estén pendientes de atención.

Para permitir una programación más flexible de las rutinas para el manejo de las interrupciones suele disponerse de esquemas vectorizados. Cada línea de interrupción lleva asociado un vector, que es una posición de memoria en una tabla a la que el hardware accede directamente cuando se detecta que se ha producido una interrupción. En dicha posición se almacena la dirección inicial de la rutina de tratamiento de la interrupción correspondiente. Con este esquema, para cambiar la rutina de tratamiento asociada a una interrupción basta con cambiar la dirección almacenada en el vector correspondiente.

Cuando hay más periféricos que líneas de interrupción disponibles es posible establecer arreglos en forma de "Daisy Chain". En este esquema la línea de solicitud de interrupciones es común para todos los dispositivos, mientras que la de reconocimiento va pasando a través de ellos de forma secuencial. Los periféricos que estén situados más próximos eléctricamente, en el sentido del reconocimiento de la señal de interrupción, son los más prioritarios. Efectivamente, ya que en el caso de dos peticiones de interrupción simultáneas el primero que reciba la señal de reconocimiento se queda con ella y no la transmite a los demás.

El mecanismo de interrupciones es muy apropiado para la sincronización de dispositivos, recordar que el reloj de tiempo real también funciona por interrupción, pero no es apropiado para la transferencia de datos entre dispositivos. Para estos menesteres lo más apropiado resultan los controladores de acceso directo a memoria (DMA). Estos son circuitos que se ocupan de la copia de datos de la memoria de un dispositivo a la memoria principal sin el concurso de la CPU. Uno de estos controladores se programa utilizando 4 registros: dirección origen, dirección destino, bytes a transferir y registro de control de las operaciones.

Los DMA pueden clasificarse según tres modos de funcionamiento:

- Modo ráfaga.
- Modo distribuido.
- Por robo de ciclo.

En el primero de ellos la transferencia se realiza por bloques, pongamos de 256 bytes. Durante este tiempo el bus queda ocupado y la CPU no puede operar con él. Se obtienen altas cuotas de transferencia pero puede no resultar apropiado para aplicaciones en tiempo real, ya que podría interrumpirse una tarea crítica.

En el modo distribuido un mismo controlador maneja la transferencia de varios dispositivos multiplexándose en el tiempo, tomando ocasionalmente ciclos de máquina del control de la CPU para realizar las transferencias. Tiene el mismo inconveniente que en el caso anterior.

Por último en el robo de ciclo el DMA emplea el bus solamente cuando la CPU no lo está utilizando. Este es el mecanismo más lento en cuanto a transferencia entre la memoria y el periférico, pero no entorpece para nada la marcha del programa que está ejecutando la CPU.

Recordar que hay esquemas Harvard (múltiples buses de datos y direcciones) donde es posible tener un bus dedicado exclusivamente para el DMA (tal es el caso del DSP TMS320C30 visto anteriormente).

9.7 Los buses en los sistemas en tiempo real.

El cuello de botella de los sistemas basados en microprocesadores es, a menudo, no la velocidad de procesamiento de la CPU, sino el bus. Hay que enfatizar este punto pues en la publicidad remarca la velocidad de las nuevas CPU's, pero debe recordarse que en muchos casos se continúa con buses e interfaces antiguas por razones de compatibilidad. Por poner un ejemplo, el Pentium es un procesador de 64 bits, que internamente en el chip tiene buses de 64 bits, pero que se monta en placas con buses de 32 bits, y que, además, no son capaces de transmitir la información a la velocidad que demanda el procesador. Por el momento vamos a centrarnos en los buses paralelos.

Los procesadores suelen llevar un bus/es interno/s para conectar las unidades funcionales que se encuentran dentro del chip más un interface para conectarse a un bus externo o bus de sistema. La memoria externa al chip, así como los periféricos, se comunican con el procesador por medio del bus externo.

La forma en que podemos encontrar físicamente estos buses son: pistas impresas en una placa, hilos en un cable o cables planos. Los buses absorben y emiten gran cantidad de interferencias eléctricas, por lo que los elementos que enlazan han de encontrarse próximos. El número de elementos conectables a un bus a priori no está limitado; serán cuestiones como el diseño físico, la potencia de la alimentación y el entorno quienes permitirán o no nuevas conexiones. En cada momento solo puede haber un elemento participante en el bus, el resto están pasivos. Las líneas de los buses se clasifican tradicionalmente en direcciones, datos y control. Antiguamente era posible encontrar líneas multiplexadas que en ocasiones llevaban direcciones, en la primera mitad del ciclo del bus, para luego portar datos. Este tipo de arreglo ya no es frecuente, ahora incluso se tiende a desdoblarse los caminos de datos, uno para datos propiamente dichos y otro para instrucciones.

Una clasificación de los buses los divide en síncronos y asíncronos. En el primero de los casos la temporización se realiza sobre la base de una señal de reloj común, mientras que para el bus asíncrono se establece un diálogo entre dispositivos por medio de señales listo y acepta. El bus síncrono es más sencillo, pero en el caso de adaptar la velocidad de transferencia a la de los dispositivos más lentos que puedan estar conectados al bus se produce una merma en las prestaciones. El bus asíncrono por el contrario permite que cada dispositivo transmita a la velocidad apropiada para él, por lo cual se adapta mejor cuando hay diversidad de periféricos, pero precisa de una circuitería más complicada. En los buses asíncronos hay la posibilidad de que pueda haber varios "maestros" que controlen el bus. Esto conlleva de nuevo una complicación adicional en la circuitería (el esquema básico del PC es una única CPU que ejerce las funciones de maestro del bus).

Un bus de altas prestaciones en la industria es el VME. Se trata de un bus asíncrono de 32 bits con posibilidades, mediante un bus de arbitrio, de poder conectar varios maestros, que pueden ser tarjetas con sus propias CPU. En muchas ocasiones este bus no se implementa sobre una placa madre a la manera de los PC's, se instala sobre un rack al que se conectan luego los dispositivos, CPU/s incluida. Esto lo configura como un sistema abierto de alta flexibilidad y donde el usuario puede conectar los elementos específicos que precise utilizar.

9.8 Redes locales en entornos industriales

Los sistemas o redes de comunicación empleados en entornos industriales se encuentran sometidos a una problemática específica que condiciona enormemente su diseño. El desarrollo del control distribuido en la industria va paralelo al de las comunicaciones digitales. Inicialmente los computadores se emplearon en labores de vigilancia y monitorización, para pasar posteriormente a sustituir a los reguladores analógicos y actualmente a implementar sofisticadas estructuras de control. Pero la posibilidad del fallo del computador (por lo general de gran potencia) obliga a tener un computador adicional interconectado para evitar el bloqueo total del proceso productivo durante su reparación. Veamos un breve resumen introductorio extraído de [6].

El desarrollo de los microprocesadores y los controladores (o autómatas) lógicos programables (PLCs) dio lugar a la aparición del control distribuido. En este tipo de control, un microprocesador o PLC controla varios lazos del sistema realizando un control digital directo de éstos, y se comunica con otros elementos de su nivel y con el nivel superior de control o supervisión. El fallo de un elemento del nivel superior no compromete (a no ser que así lo desee el usuario) el funcionamiento de los lazos de control que gestionan los microprocesadores. De la misma forma, el fallo de uno de estos puede tener una incidencia mínima en el funcionamiento del resto del proceso. Del mismo modo, en los procesos de fabricación aparecen elementos de control discreto (máquinas de control numérico, autómatas programables, robots, etc.) que favorecen la automatización y flexibilización de los procesos productivos. Todos estos elementos permiten su programación y control de forma remota con un adecuado sistema de comunicaciones.

De hecho, un sistema de control distribuido está formado por tres componentes principales:

- 1) Los controladores basados en microprocesadores, con algoritmos de control digital directo, capacidad para hacer lecturas de sensores del sistema o actuar sobre elementos del mismo (microprocesadores, PLCs, máquinas de control numérico, robots, etc.).
- 2) El sistema de interface para el operador, basado generalmente en monitores de vídeo en color que informan al usuario de la situación del sistema. Este elemento sustituye al panel de control original.
- 3) El sistema de comunicación interno entre los controladores, la interface para el operador y otros módulos o computadores que puedan existir.



Niveles jerárquicos en la automatización industrial

El sistema de control distribuido pasa a formar parte del control por computador que puede implantarse de forma general en una factoría. El estado final es una instalación con un sistema de control global de todas las actividades mediante una estructura que incluye sistemas basados en procesador de todos los tipos: microprocesadores, microcomputadores, controladores digitales, autómatas programables, computadores personales, minicomputadores y grandes computadores. Todos ellos conectados mediante canales de comunicación, de distintos tipos si es necesario, y organizados por niveles en una estructura jerarquizada como la que se muestra en la figura adjunta.

Cada uno de los niveles, además de llevar a cabo labores específicas, realiza un tratamiento y filtrado de la información que es transmitida en sentido ascendente o descendente por la pirámide. De esta forma se limitan los flujos de información a los estrictamente necesarios para cada nivel. Además del tráfico en sentido vertical de información, existe un flujo de información horizontal dentro de cada nivel con distintas condiciones en cada uno de ellos. En los niveles inferiores se transmiten un número limitado de datos, que se actualizan con gran velocidad, en bloques de pequeño tamaño. En los niveles superiores el volumen de información transmitido es mayor y los bloques de datos son de gran tamaño. Asimismo, los programas que tratan estos datos, que por otro lado mantienen más tiempo su actualidad, son más sofisticados.

El nivel inferior realiza el control digital directo de los bucles de regulación (en sistemas de control de procesos continuos o discretos) o el control de los elementos de fabricación (en sistemas de fabricación flexible). Se adquieren datos de los sensores, se actúa en función de algoritmos de control y consignas seleccionadas por el nivel superior, se ejecutan programas de mecanización o manipulación, se activan alarmas y se transmiten los mensajes e informaciones oportunas al nivel superior.

El siguiente nivel es de supervisión en el ámbito de célula de fabricación o de control. Elabora la información procedente del nivel inferior y se informa al operario de la situación de las variables y de las alarmas. Corrige algoritmos de control, consignas y programas.

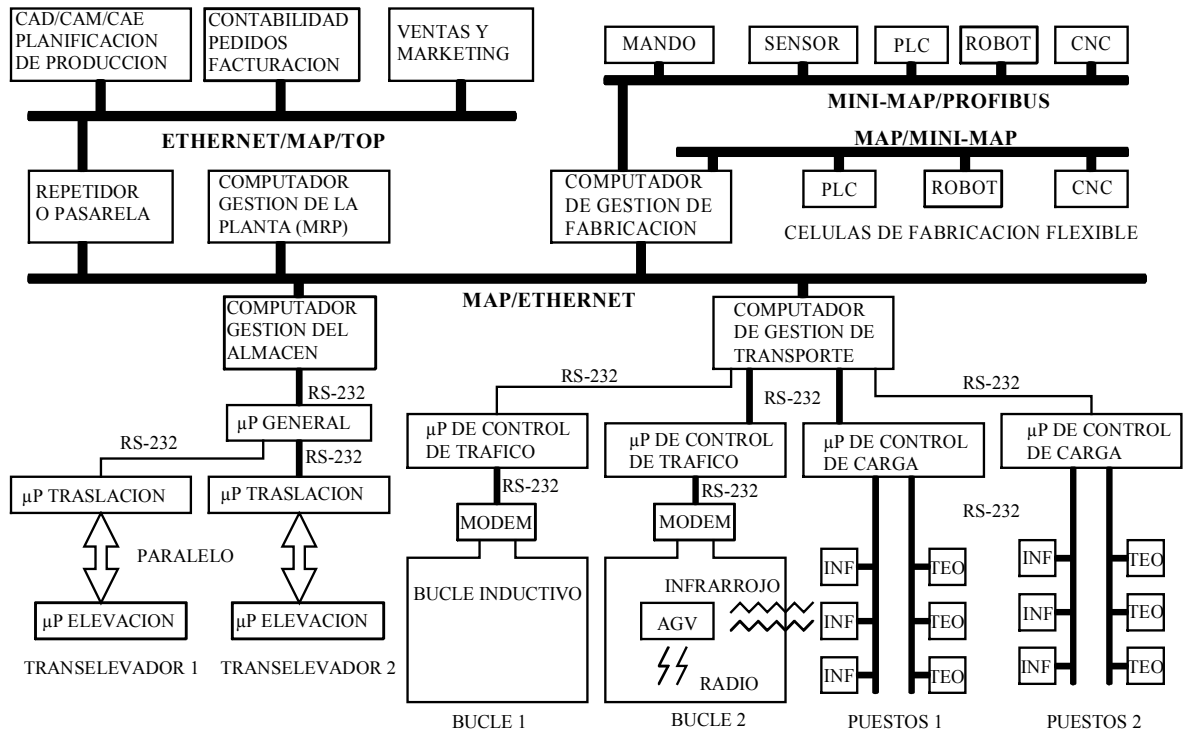
El tercer nivel lleva a cabo labores de coordinación de la planta. Controla y organiza toda el área de producción tratando de optimizar balances de materias y energía. Con el fin de alcanzar esa optimización, establece las condiciones de operación de cada proceso del área y las envía a cada control supervisor para que éstos las adapten y distribuyan entre los controles directos. La programación de este elemento es fundamental para la obtención del Just In Time (JIT), de manera que la coordinación de todas las tareas permita la optimización tanto de tiempos de fabricación del producto como de los períodos de almacenamiento de materiales y stocks.

En el siguiente nivel se realiza la planificación de la producción del conjunto de la factoría. Se encuentran en él también los elementos de oficina técnica que mediante herramientas CAD/CAM/CAE permiten el diseño de productos y la elaboración automática de programas para los elementos de fabricación. En el nivel superior se establecen la política de producción y de fabricación de la empresa en función de los recursos y costes del mercado. En él se incluyen labores de contabilidad y gestión empresarial.

La consecución de la implementación completa de todos estos niveles da lugar a la aparición del CIM (Computer Integrated Manufacturing). El principal inconveniente para el logro de esta integración se encuentra en los problemas que presenta la intercomunicación de los elementos de la base de la pirámide.

9.8.1 Las comunicaciones en el entorno industrial

Si se desea llegar a la configuración de un sistema distribuido jerarquizado tal como se describía en el apartado anterior se deberán utilizar para las comunicaciones, redes de capacidad graduada para cada uno de los niveles, ya que no será rentable, ni viable utilizar una red de mucha capacidad para realizar todas las tareas de comunicación de todos los dispositivos de la organización. Un ejemplo de configuración del sistema de comunicaciones podría ser el representado en la siguiente figura correspondiente al diseño de una planta piloto de fabricación flexible.



COMUNICACIONES EN LA PLANTA PILOTO

En dicho diseño se plantean las posibles soluciones, en lo que se refiere al sistema de comunicaciones para la integración de todos los elementos. Se utilizan redes de gran capacidad para los niveles superiores, con el fin de absorber grandes volúmenes de datos. En los niveles inferiores se aconsejan sistemas de comunicación de bajo coste, adecuados para transferencias rápidas de pequeños volúmenes de datos.

9.8.2 Comunicaciones en tiempo real

El tiempo de respuesta requerido por una red viene determinado por las características de respuesta temporal de la planta o elemento a controlar. De la misma forma, el tiempo de respuesta de una planta o elemento en condiciones anormales, determinará el tiempo máximo de comunicación necesario para un mensaje de alarma.

En los niveles inferiores de la pirámide, el sistema de comunicación deberá ser capaz de proporcionar un tiempo de respuesta máximo, compatible con las exigencias de tiempo real del proceso. En las redes locales utilizadas para la automatización de oficinas, este parámetro tiene mucho menos interés, pues no existe riesgo de una disfunción grave para el receptor de la información.

A la hora del diseño de un sistema de comunicaciones en tiempo real, se plantea la discusión entre los partidarios de los mecanismos probabilistas de acceso por contención (del tipo CSMA/CD como Ethernet) y los de los mecanismos deterministas de acceso por paso de testigo. El segundo de los métodos puede garantizar un tiempo de respuesta máximo y acotado superiormente mientras que un sistema de acceso múltiple como el primero, con posibles colisiones, es necesariamente de tipo probabilístico. Sin embargo, se podría obtener una probabilidad suficientemente pequeña de superación del valor máximo aceptable. Este valor de probabilidad podría ser tan pequeño como el de que, por ejemplo, se pudiera producir un error en la transmisión. Obsérvese, que incluso en el caso de un mecanismo por paso de testigo, la posibilidad de que se produzcan errores durante la transmisión convierten también a este método en probabilístico.

Otro tema importante en los sistemas de control industrial es el de la fiabilidad de los datos. Generalmente los mecanismos de detección de errores de los protocolos de comunicación detectan y recuperan los errores producidos en una transmisión. Pero los mecanismos de detección no son capaces de detectar todos los errores posibles y pueden producirse errores no detectados y, por lo tanto, no recuperados ni comunicados al operador. En las comunicaciones industriales se exigen tasas de errores no detectados del orden de un error cada 20 años, equivalente a una probabilidad de error de 10^{-12} para un mensaje menor de 100 bytes en una red de 1 Mbps de velocidad, al 100% de funcionamiento. Normalmente los códigos de detección de errores por redundancia cíclica son capaces de satisfacer estas tasas.

Por otro lado, un fallo permanente o intermitente en un dispositivo conectado a la red no debe causar un fallo global. Por lo tanto, el dispositivo que falla debe dejar de comunicarse.

9.8.3 Tendencias actuales

Es pues el tema de la comunicación donde se plantean los problemas que aun no tienen una solución definitiva en el control distribuido de los procesos industriales. Las principales deficiencias son:

- Los protocolos de alto nivel son de gran complejidad y elevado coste, lo que encarece excesivamente los elementos que realizan el control directo de las variables del proceso
- Las líneas de comunicación, como el clásico RS-232-C o las líneas 4-20 mA, habituales en los pequeños equipos (microprocesadores, autómatas, etc.), son demasiado elementales para el desarrollo de un sistema íntegro de control distribuido en tiempo real.

Actualmente los productos existentes en el mercado de las redes locales de comunicación, tanto para computadoras como para autómatas programables y máquinas de control numérico, aportan ya algunas soluciones para facilitar la comunicación en el control distribuido de un proceso productivo.

Las tendencias actuales más sobresalientes son:

- La realización de redes locales de alto nivel, como la propugnada por General Motors con el desarrollo de un protocolo de comunicaciones de alto nivel para redes locales (MAP-TOP), que permitan el control en tiempo real del proceso y la conexión de todos los elementos inteligentes de la planta a la red.
- El desarrollo de redes de comunicación de bajo coste para microprocesadores, autómatas programables, máquinas de control numérico y otros elementos de campo, comunicadas mediante pasarelas con las redes locales de niveles superiores.

La red MAP tiene una estructura compleja y sus interfaces son costosas, por lo cual, se definió una versión simplificada y más económica, MINI-MAP, con el fin de dar servicio a los elementos de la base de la pirámide, ofreciendo un servicio de comunicaciones en tiempo real. Sin embargo, las interfaces y el cableado con cable coaxial, siguen teniendo un coste y complejidad desproporcionados para algunos de los elementos a interconectar.

Como solución a estos problemas, se encuentran los buses de campo cuyas funciones son:

- Mejorar la calidad y cantidad en el flujo de datos.
- Ahorrar coste y peso de cableado e instalación.
- Facilitar la ampliación y reducción del número de elementos de campo de un sistema.
- Minimizar el número de conexiones.
- Reducir errores de instalación.
- Reducir terminales y cajas de conexión.

Los elementos de campo pueden ser elementos de medida de temperatura, presión, caudal, PH, elementos pasivos como accionamientos on/off, pequeños autómatas programables, controladores, terminales manuales de adquisición de datos, dispositivos de entrada/salida, etc.

El principal inconveniente de los buses de campo es la falta de un estándar único. Los fabricantes de sistemas de control distribuido han creado sus propias soluciones ante la falta del mismo. En muchas de estas soluciones, se utiliza como sistema físico de transmisión la interface RS-485 que permite una transmisión multipunto sobre un bus de hasta 1200 m y una velocidad de transmisión máxima de hasta 10 Mbps. Esta línea de transmisión ofrece una gran inmunidad a las interferencias.

Otros elementos, comunes en los sistemas distribuidos que utilizan un gran número de autómatas programables u otros elementos de adquisición de datos y control, son las redes de adquisición de datos distribuidas. Estas redes

consisten en la prolongación del alcance de las líneas de adquisición de datos o salidas de control de un dispositivo, mediante la distribución de los módulos de Entrada/Salida en posiciones remotas y su conexión mediante una línea cableada única al módulo central. Estas estructuras no se pueden considerar como un control distribuido, ya que generalmente el control se realiza únicamente en el módulo central y el bus sólo se utiliza para la adquisición de todas las señales y el envío de las señales de control.

9.8.4 Topología de la red

En las comunicaciones industriales, a la hora de elegir entre una topología de red en bus o en anillo, se opta prácticamente siempre por la primera.

En las topologías en anillo cada estación esta unida físicamente a una anterior y otra posterior. La estación siempre recibe los mensajes de la estación anterior y cuando no están dirigidos hacia ella, la interface de la estación los repite sin modificarlos hacia la estación siguiente. Las redes en anillo aseguran el acceso a la red en un período de tiempo máximo limitado, simplifican los mecanismos de acuse de recibo y proporcionan velocidades de transmisión altas con tasas de errores muy bajas. Generalmente, en las comunicaciones industriales se desecha esta topología por que un fallo de una de las interfaces puede bloquear las comunicaciones de todo el resto. Además, suele ser complicada la incorporación de nuevas estaciones a la red o la ampliación del alcance de esta.

En las topologías en bus todas las estaciones se conectan a un mismo tramo de cable, aunque se pueden crear estructuras en árbol mediante repetidores, y todas escuchan los paquetes transmitidos por el canal de transmisión. En este caso, el fallo de la interface de una estación no afecta, por lo general, al funcionamiento del resto de la red. La inserción de nuevas estaciones es sencilla y se consiguen también altas velocidades de transmisión con tasas de errores muy bajas. Como inconveniente, si se desea un tiempo máximo de acceso al canal de transmisión acotado, es necesario recurrir a un protocolo para el mecanismo de control de acceso al medio más elaborado.

Los protocolos y redes de comunicación de datos que se describen a continuación, se implementan siempre o en la mayoría de los casos, sobre topologías en bus.

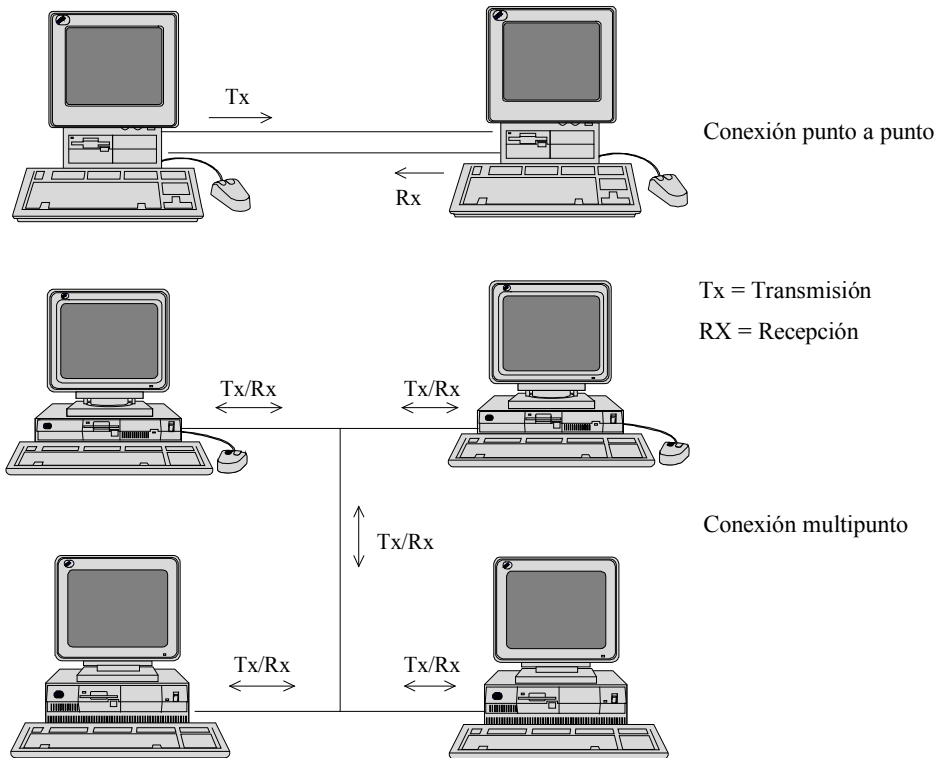
Normalmente, en una red de comunicaciones industriales coexisten varios tipos de redes y protocolos de comunicaciones, que interaccionan entre sí a través de pasarelas (gateways). Se establece una arquitectura jerárquica en la que, en cada nivel, se puede optar por un sistema de comunicaciones diferente en función de las necesidades que exige dicho nivel. Fabricantes de sistemas de control distribuido como Siemens o Allen-Bradley, utilizan en sus aplicaciones comunicaciones mediante Ethernet, MAP, buses de campo, etc. ([SIEMENS 91], [ALLEN 89]) en uno o más de los niveles que componen sus arquitecturas de control.

9.8.5 Buses de campo

Un bus de campo (FieldBus) ha de ser una solución de bajo coste, tiempo real, con transmisión serie de datos y con capacidad para interconectar controladores con todo el espectro de dispositivos de entrada/salida. Según el comité 50 de la ISA (Instrument Society of America), un bus de campo es una conexión serie digital, que permite la transferencia de datos entre elementos primarios de automatización (elementos de campo), empleados en fabricación y procesos, y elementos de automatización y control de más alto nivel.

El concepto elementos de automatización engloba elementos de medida de temperatura, presión, caudal, PH, elementos pasivos como accionamientos on/off, pequeños autómatas programables, controladores, terminales manuales de adquisición de datos, dispositivos de entrada/salida, etc.

El objetivo del bus de campo es llegar a sustituir las conexiones punto a punto entre los elementos de campo y el equipo de control a través del tradicional lazo de corriente de 4-20 mA. Por lo tanto los elementos de campo habrán de volverse más inteligentes para poder dialogar a través del bus con los equipos de control.



Las posibilidades de gestión del bus son maestro-esclavo o multimaestro. En este último caso se pueden utilizar técnicas de acceso aleatorio o por paso de testigo.

MINI-MAP se presentaba en principio como la solución al bus de campo y la interconexión de estos dispositivos, pero se argumentan dos principales inconvenientes a su utilización:

- El precio de una conexión MINI-MAP para un elemento muy simple resulta desproporcionado.
- El nivel de aplicación de MAP es excesivamente complejo para las necesidades de estos elementos.

Se está tratando de desarrollar un estándar de bus de campo que cubra las necesidades de todo el espectro de usuarios. Los objetivos de este sistema de comunicación son:

- Bus serie asíncrono
- Funcionamiento en tiempo real.
- Prestaciones predecibles.
- Protocolos simples y limitados.
- Mejorar la calidad y cantidad en el flujo de datos.
- Ahorrar coste y peso de cableado e instalación.
- Fácil configuración simplificando la ampliación y reducción del número de elementos de campo de un sistema.
- Minimizar el número de conexiones.
- Reducir errores de instalación.
- Independencia del fabricante. Reducir terminales y cajas de conexión.

Con este fin han aparecido algunos estándares para redes de bajo coste que persiguen estos objetivos. Las líneas de este estándar van hacia un modelo de sólo tres niveles de referencia OSI; Físico(1), Enlace (2) y Aplicación (7) con las siguientes características.

1. Nivel Físico

Es el nivel que se encuentra más desarrollado hasta el momento. Se ha pensado también en utilizar, a parte del hilo de cobre, transmisión por radio y fibra óptica. Sus características principales son:

- Bus multipunto de hasta 32 dispositivos.
- Modo H1 para proceso continuo de baja velocidad: 31,25 Kb/s y cable de par trenzado. Hasta 1900 m y 4 repetidores.
- Modo H2 para proceso discreto de alta velocidad: 1 Mb/s y cable de par trenzado apantallado. Hasta 750 m y 4 repetidores.
- Codificación Manchester Bi-Phase L.

2. Nivel de Enlace

El mayor inconveniente está en la definición del MAC (control de acceso al medio). El grupo de FIP apuesta por un sistema con un único maestro mientras que Profibus propone el paso de testigo de IEEE 802.4. Este último proporciona una gran flexibilidad para cada estación de la red, pero se pueden producir pequeños desfases entre las muestras adquiridas y los valores transmitidos en el bus, sobre todo si el tráfico de datos es elevado. Además su realización es más compleja que la de un MAC centralizado. La solución final puede ser un MAC centralizado con posibilidades restringidas de paso de testigo, incluyendo unos tiempos límite en el retorno de los testigos al arbitro central.

3. Nivel de Aplicación

Se pretende en principio utilizar MMS lo que permitiría una conexión casi inmediata con MAP, aunque es posible que se añadan algunas características adicionales. Ante la falta de un único estándar, distintos fabricantes han desarrollado ya sus propias soluciones.

Bibliografía

- [1] Real Time Systems Engineering and Applications M.Schiebe, S. Pferrer Ed. Kluwer 92 ISBN 0-7923-9196
- [2] Real Time Computer Control Stuart Bennet Ed. Prentice Hall 94ISBN 0-13-764176-1
- [3] Real Time Microcomputer System Design: an Introduction Peter D. Lawrence, Konrad .Mauch Ed. Mc graw Hill 87 ISBN 0-07-100561-7
- [4] Diseño de un Sistema de Control en Tiempo Real basado en Procesamiento Digital de Señal. Aplicación al Control Dinámico de Motores Asíncronos. José A. Cancelas Tesis Doctoral Universidad de Oviedo
- [5] Solucionario del PC,XT,AT. R.Jourdain Ed. Anaya
- [6] Redes Locales en Entornos Industriales (Resumen) José A. Sirgo Lección de oposición Universidad de Oviedo

TEMA 8. REQUISITOS HARDWARE PARA LOS SISTEMAS EN TIEMPO REAL

- 8.1. INTRODUCCIÓN
- 8.2. EVOLUCIÓN DE LA ARQUITECTURA DE PROCESADORES
 - 8.2.1. *Computador de propósito general*
 - 8.2.1.1. Unidad central de procesos CPU
 - 8.2.1.2. Almacenamiento
 - 8.2.1.3. Entrada/Salida
 - 8.2.1.4. Estructura del bus
 - 8.2.2. *Microcontroladores.*
 - 8.2.3. *Procesadores especializados*
 - 8.2.3.1. Procesadores RISC (Reduced Instruction Set Computer)
 - 8.2.3.2. Computadores paralelos.
 - 6.4.4.2.1 Los transputers
 - 8.2.3.3. Procesadores digitales de señal (DSP)
 - 8.2.3.4. Redes de computadores
- 8.3. EL RELOJ DE TIEMPO REAL
- 8.4. CONVERTIDORES ANALÓGICOS/DIGITALES (CAD) Y DIGITALES/ANALÓGICOS (CDA)
 - 8.4.1. *Convertidores Analógicos/Digitales (CAD)*
 - 8.4.2. *Convertidores Digitales/Analógicos (CDA)*
- 8.5. INTERFACES DIGITALES
- 8.6. TÉCNICAS DE TRANSFERENCIA DE DATOS
- 8.7. LOS BUSES EN LOS SISTEMAS EN TIEMPO REAL.
- 8.8. REDES LOCALES EN ENTORNOS INDUSTRIALES
 - 8.8.1. *Las comunicaciones en el entorno industrial*
 - 8.8.2. *Comunicaciones en tiempo real*
 - 8.8.3. *Tendencias actuales*
 - 8.8.4. *Topología de la red*
 - 8.8.5. *Buses de campo*