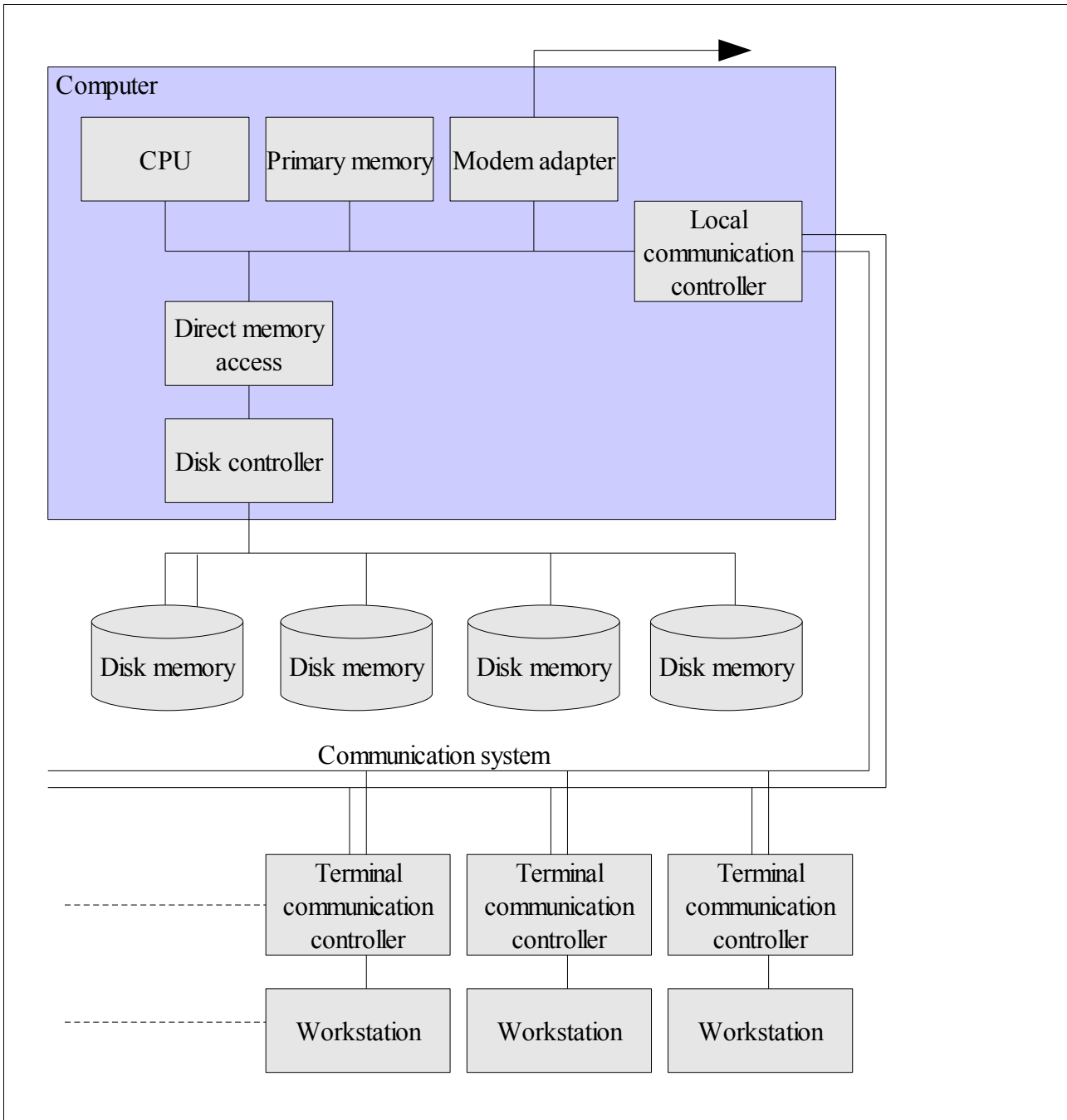


SITEMA

FACIT 6501

CONTENIDO

DESCRIPCIÓN DEL SISTEMA: HARDWARE.....	4 (5)
Introducción.....	4 (5)
Equipo Central.....	6 (7)
Introducción.....	6 (7)
Computadora.....	6 (7)
Sistema I/O.....	8 (9)
Memoria de Discos.....	9 (11)
Sistema de Comunicación.....	11 (13)
DESCRIPCIÓN DEL SISTEMA: SOFTWARE.....	15 (17)
Introducción.....	15 (17)
Programas de Utilidad.....	17 (19)
Sistema Operativo.....	19 (21)
Operaciones I/O.....	21 (25)
Intérprete.....	22 (27)
Generación del Sistema.....	23 (29)
Programas de Utilidad.....	24 (31)
LOGIC-3 RESUMEN.....	26 (33)
Introducción.....	26 (33)
Estructura de los Programas.....	27 (35)
Funciones del Programa.....	28 (37)



Cada sistema incluye una computadora y una o más unidades de discos, así como un sistema de comunicación al que se conectan las estaciones de trabajo por medio de cables normales de dos hilos. La computadora y las unidades de disco son elementos clave del equipo central. Todo el proceso y almacenamiento de datos se realiza en el equipo central. Se pueden conectar el Facit 6501 a un ordenador mediante una línea telefónica utilizando un equipo módem.

DESCRIPCIÓN DEL SISTEMA: HARDWARE

Introducción

El Facit 6501, es un sistema ideal para el tratamiento de la información así como para todo tipo de aplicaciones. La flexibilidad que nos brinda este equipo, se obtiene utilizando un concepto modular sistemático tanto para el Hardware como para el Software.

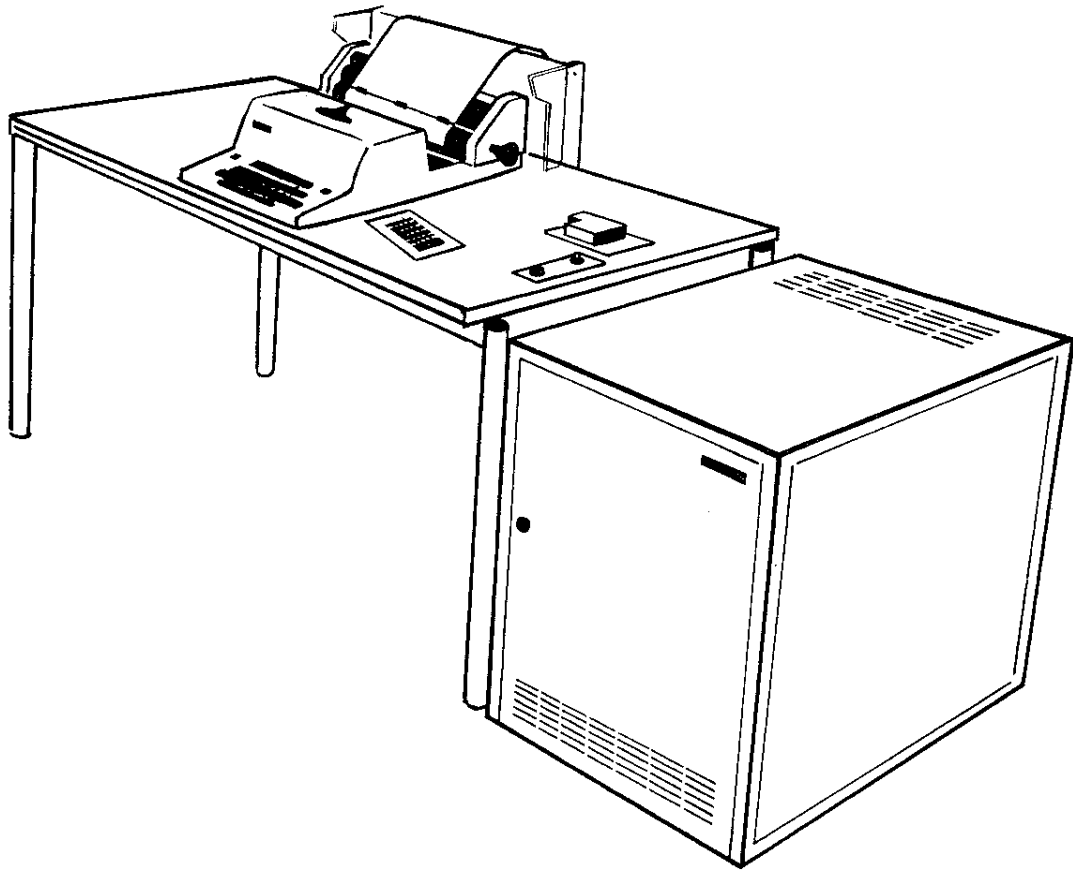
El sistema, está diseñado en base a una computadora con unidades de discos magnéticos y un sistema de comunicación, al cual, pueden conectarse las distintas estaciones de trabajo. Los puestos o lugares de trabajo se conectan utilizando cables telefónicos de dos hilos los cuales pueden tener una extensión de 2.000 metros. Adicionalmente, puede conectarse la computadora a otro sistema mayor, mediante la red general de comunicaciones.

Este principio de diseño general, ha creado los medios para un sistema de tiempo real, en el cual cada estación de trabajo, funcionará como un terminal de diálogo, completamente independiente de las demás estaciones. Cada operador, en su estación de trabajo, está libre para seleccionar el programa que precise para la aplicación en particular y de ninguna forma, es influenciado por otras actividades en el sistema general. Todas las estaciones de trabajo, tienen acceso a las unidades de discos y a la unidad central (acceso múltiple).

Las estaciones de trabajo, están diseñadas para ajustarse a la mayoría de los requisitos del usuario. Una estación de trabajo, está compuesta de un número de módulos adecuado para una área de aplicación. En este contexto, se entiende un "Módulo" como una unidad específica de Hardware, incluida en la estación de trabajo. Por ejemplo: un teclado o una impresora. Las estaciones de trabajo incluidas en la instalación de un sistema Facit 6501, pueden ser diseñadas idénticamente, pero también pueden ser completamente distintas las unas de las otras. No obstante, siempre se incluye una unidad de control de comunicación de terminal (Facit 5124) y una fuente de alimentación (Facit 5025), con la excepción de las estaciones de trabajo, formadas por un display. La unidad de display, está equipada internamente, con la unidad de control de comunicaciones y una fuente interna de alimentación. Se pueden conectar un máximo de 63 módulos al sistema de comunicación.

Las estaciones de trabajo, están claramente orientadas hacia el sistema de comunicación, mediante un interface de Hardware. Esto, facilita la conexión de nuevos tipos de estaciones de trabajo y la ampliación de la capacidad total del sistema para mantenerse al día según los requisitos del usuario.

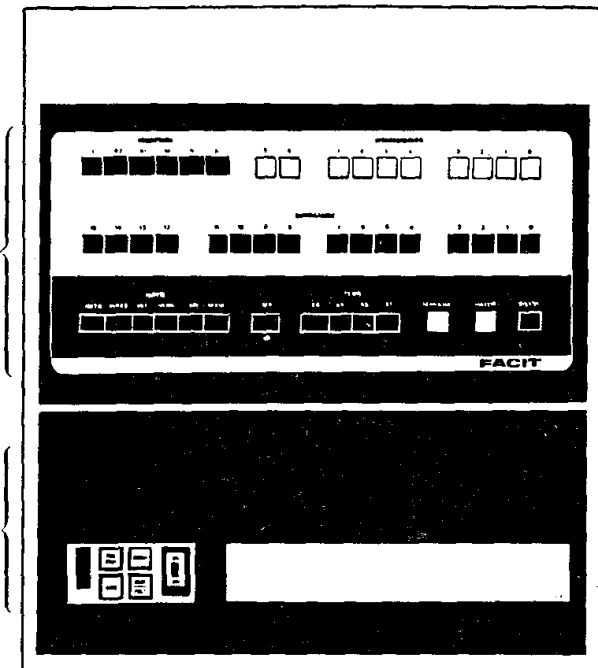
Naturalmente, también se utiliza la modularidad en el diseño de la memoria primaria y de la memoria de discos, permitiendo una ampliación a base de incrementos.



Central equipment with one work station

COMPUTER

DISK MEMORY



Central equipment

Equipo central

Introducción

El equipo central Facit 6501, está diseñado en base a un sistema modular. Esta modularidad, permite dimensionar los recursos de la memoria primaria y de la memoria de discos, según las necesidades del sistema. Es posible, conectar un adaptador para una conexión remota mediante un módem. Se puede ampliar la memoria primaria, con incrementos de 4K a 8K bytes hasta un máximo de 32 K bytes.

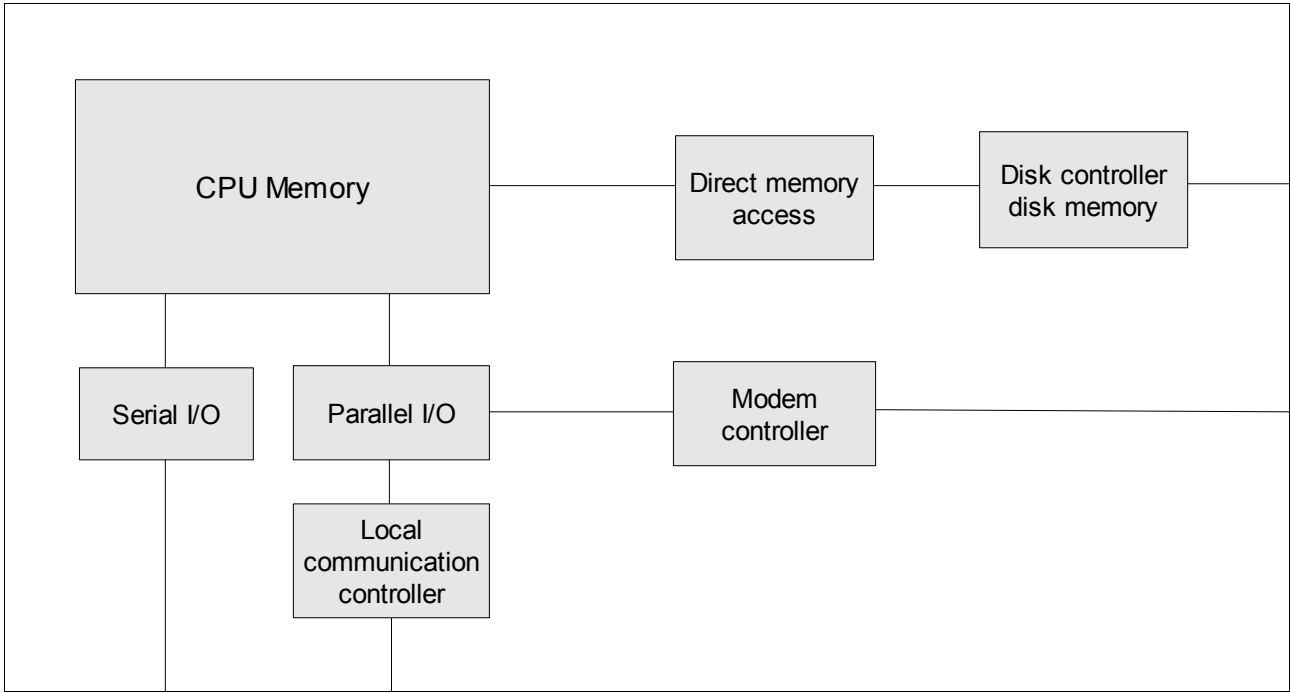
La memoria secundaria, se compone de una unidad de discos que puede ser ampliada hasta un máximo de cuatro. Estas unidades de discos, están equipadas con uno o dos discos. Un disco es intercambiable (principio de cassette). La capacidad de cada disco es de 2,5M bytes. Por lo tanto, se puede aumentar la capacidad desde 2,5M a 20M bytes.

Al equipo central se le puede acoplar un line adapter para la transmisión de datos asíncrona o síncrona.

Computadora

La computadora, está ideada en base a una filosofía universal, orientada al byte, con un diseño modular. Incluye un canal de acceso directo a memoria, un potente sistema I/O y un sistema eficaz de interrupciones de funcionamiento, lo cual, da plena seguridad al sistema. El micro, incluye un juego de 89 instrucciones. Algunas de estas instrucciones, operan con longitud de palabra variable, siendo muy adecuadas para el manejo de conjuntos de datos. Asimismo, existen hasta ocho formas distintas de direccionamiento de memoria, lo cual, facilita considerablemente el sistema de programación.

El procesador está equipado con un bootstrap micro-programado, lo cual, facilita considerablemente la puesta en marcha del sistema. Se incluye como estándar un reloj de tiempo real y un dispositivo automático de puesta en marcha, después de un corte de corriente. La computadora, las unidades de disco y las unidades de control de los mismos, están alojadas en un armario. Todo el equipo está diseñado en base a las técnicas más modernas de fabricación.



Sistema I/O

La computadora tiene tres canales I/O: un canal I/O en paralelo (tráfico de datos controlado por el programa y, mediante la unidad de control I/O), un canal I/O serial (en contacto directo con el procesador) y un canal de acceso directo a la memoria DMA (tráfico de datos controlado por la unidad de control de discos).

■ **Canal I/O para transferencias en paralelo**

Se utiliza este canal para transferencias de datos desde/a las estaciones de trabajo. Bajo el control del microprograma, se transfieren los datos byte a byte, siendo la unidad de control de comunicación local (Facit 5123) la que se encarga de convertirla en serial, para su transferencia por la línea de dos hilos, a las estaciones de trabajo.

La transferencia de datos es realizada en dos fases, la primera de las cuales establece contacto con la unidad receptora. La transferencia real se realiza en la segunda fase, a continuación de la cual, se puede empezar otro ciclo de transferencia.

Además de la unidad de control de comunicaciones Facit 5123, el Facit 6501, puede utilizar un adaptador para la transferencia de datos a través de una red de comunicaciones. También en este caso, se realiza una conversión de paralelo a serial. La transferencia de datos del Facit 5123 a la memoria primaria o a los registros internos, está controlada por seis instrucciones de micro programa. La transferencia de datos entre el módem y la memoria primaria, está controlada por programa utilizando estas seis instrucciones, o bien por una "instrucción concurrente" que permite la transferencia automática de un bloque, sin control de programa. Esto significa que el programa puede funcionar en otras tareas mientras se efectúa la transferencia de datos.

■ **Canal I/O para transferencia en serial**

Sólo se utiliza este canal, para conexión con el sistema, para el servicio de mantenimiento. La transferencia está controlada por microprogramación. Ha sido diseñado para una velocidad de transferencia de 110 baudios, utilizando caracteres de 11 bits. Existen dos micro-instrucciones: Entrada de Bytes Serialmente (IBS) y Salida de Bytes Serialmente (OBS).

■ **Sistema de interrupciones**

Se pueden efectuar interrupciones externas desde el Facit 5123, o bien desde el módem, mediante el canal orientado a bytes. Cada unidad conectada tiene definida una cierta prioridad. Cuando se ha de efectuar una interrupción el sistema verifica si la unidad en cuestión tiene la máxima prioridad. En caso positivo se efectúa la interrupción. Cuando se ha recibido una respuesta positiva a la interrupción, la unidad emite un byte de dirección, el cual es utilizado por la unidad de conexión para la iniciación de un ciclo de transferencia de datos.

■ **Canal DMA (Acceso directo a la Memoria)**

El equipo dispone de un canal de acceso directo a la memoria para una transferencia rápida de datos a/o desde la memoria de discos. Se utiliza dicho canal para la transferencia de datos directamente entre la memoria primaria y la secundaria. Se inicia la transferencia a través del canal DMA con la asistencia del programa, continuando de forma completamente automática y sin estar influenciada por el programa. El canal DMA está diseñado para la transferencia de bloque por bloque, pero puede también utilizarse para transferencia de byte a byte.

■ **Memoria de discos**

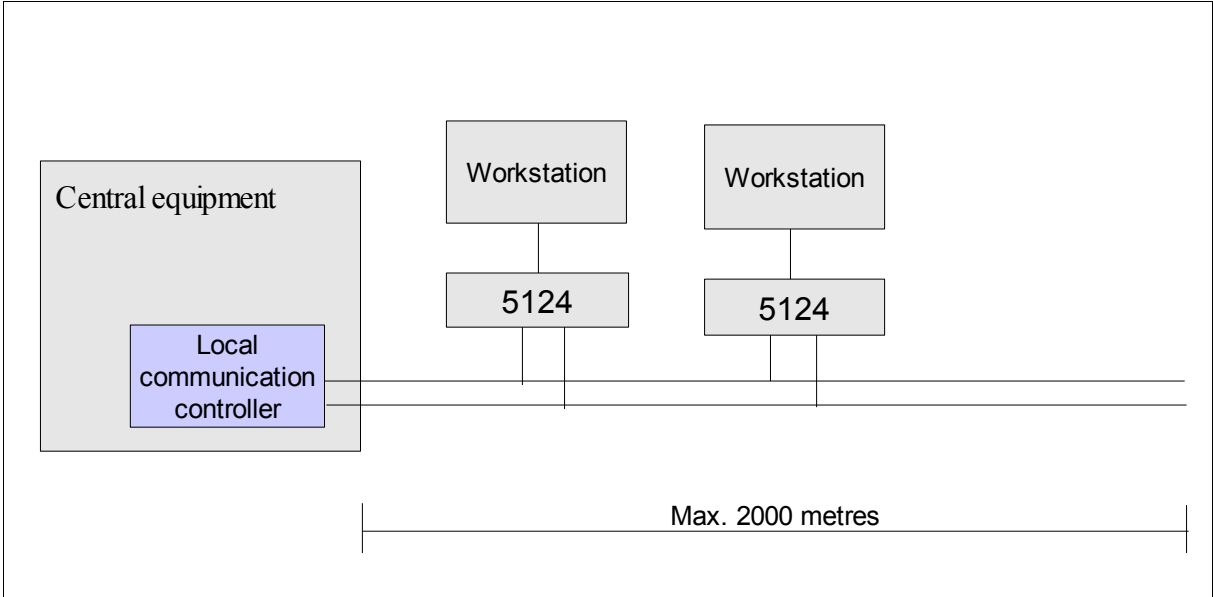
Las unidades de discos están disponibles en modelos de uno o dos discos. Uno de los discos es intercambiable. La capacidad de un disco es de 2,5M bytes. Las unidades de discos están conectadas a la computadora mediante una unidad de control de discos Facit 5129, situado en el bastidor de la computadora. Los datos se almacenan en ambas caras del disco, siendo el cassette compatible con unidad IBM 2315. Los datos son almacenados en bloques de 256 bytes. Cada cara de disco contiene 203 pistas con 24 bloques.

La unidad de control de discos verifica que los datos hayan sido leídos o grabados correctamente antes de que sean transferidos desde /o al canal DMA. Esta verificación se basa en el chequeo de una suma que se calcula y almacena durante la grabación y que se verifica durante la lectura. También se realiza una verificación de la dirección.

La transmisión de datos desde los discos al procesador se efectúa a través del canal DMA. El tiempo de transferencia de un byte es de aproximadamente 5 micro-segundos. Un bloque de 256 bytes es transferido en 1,5ms. En respuesta a una orden se pueden transferir hasta un máximo de 48 bloques.

El tiempo transcurrido entre la emisión de la orden y la transferencia del primer byte, puede ser de 9-100ms, según la posición de las cabezas de grabación/lectura, en el momento en que se emite la orden. La unidad de control de los discos Facit 5129 puede controlar hasta cuatro unidades de discos.

Todos los programas de aplicación y todos los datos registrados y procesados dentro del sistema, son almacenados en los discos. Además, se almacena una copia de la información de los programas ("software" y parámetros del sistema) de la memoria primaria, lo cual permite una carga rápida de la memoria primaria. Cuando se ha de poner en marcha el sistema sólo debe efectuarse la lectura de un programa de control de disco a través del canal serial.



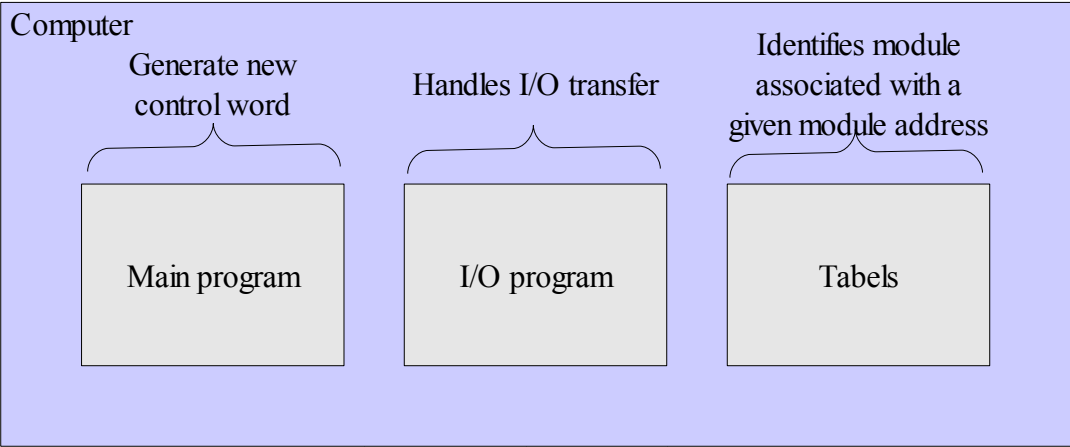
Sistema de comunicación

Para la transmisión de datos entre la computadora y las estaciones de trabajo, se utilizan las siguientes unidades de hardware: Unidad de Comunicación Local Facit 5123 (LCC), Cable de dos hilos y la Unidad de Comunicación Terminal Facit 5124 (TCC). La longitud total del cable de dos hilos pueden llegar a 2.000 metros. La Unidad de Comunicación Local 5123 está construida sobre una placa de circuitos impresos y se halla ubicada en la computadora. La Unidad de Comunicación Terminal 5124 proporciona la conexión entre los módulos de las estaciones de trabajo y el cable de dos hilos.

Las operaciones I/O se realizan independientemente de la ejecución de los programas (programas principales). La LCC llama a la computadora cuando tiene algo para transmitir o desea recibir información pero de no ser así, maneja las comunicaciones con las diferentes estaciones de trabajo, de forma totalmente independiente. El gráfico nos muestra el principio de las operaciones I/O.

Después de la primera llamada de la LCC, la computadora emitirá una *palabra de control* de un máximo de 64 bits. Cada posición corresponde a una posible dirección de módulo. Los parámetros del sistema en la memoria primaria incluyen una tabla reflejando el diseño de las estaciones de trabajo y la dirección de sus distintos módulos. Cuando aparece una inscripción I/O durante la ejecución del programa principal, se ha de establecer contacto con un módulo determinado. De este modo se colocará en "uno" la posición del bit (en la palabra de control) que corresponda a este módulo.

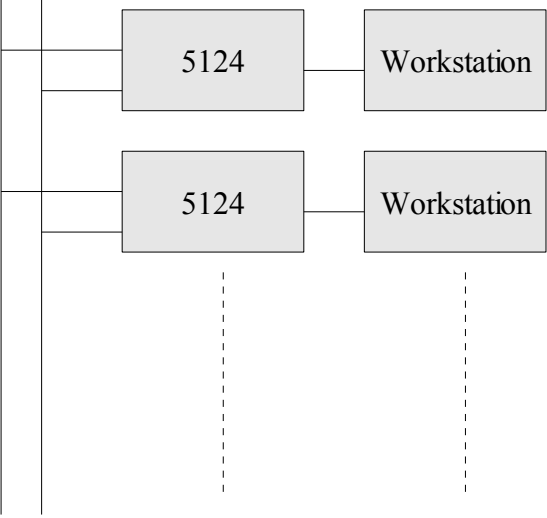
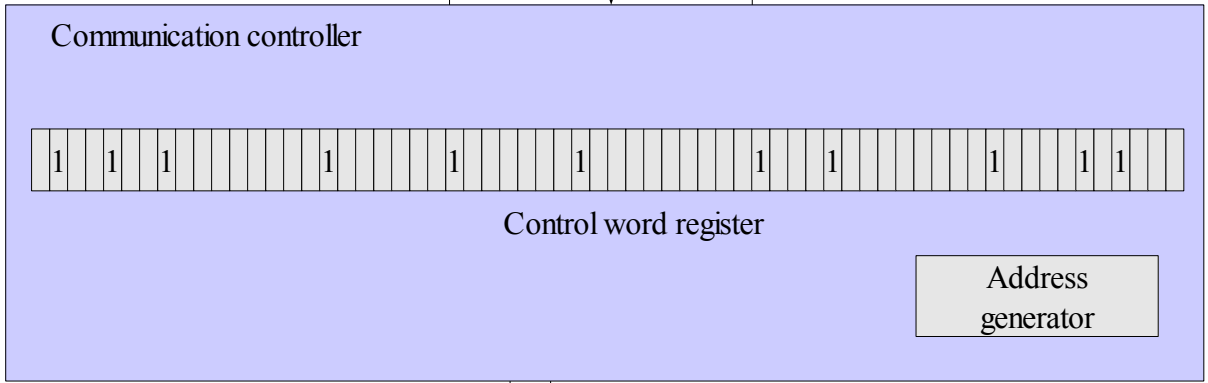
La LCC investiga la "palabra de control", posición por posición y cuando encuentre un bit en situación de "1", se pondrá en contacto con el módulo en cuestión mediante la línea de dos hilos y la TCC. No se realizará otra llamada de la LCC a la computadora, hasta que el módulo haya entregado los datos a la LCC o esté preparado para recibir los datos. A continuación, se le indicará a la computadora cuál de los dos módulos ha establecido contacto y se investigarán los parámetros del sistema para saber si ha de enviar o recibir datos (un módulo que tiene interface tanto de entrada como de salida es considerado por la computadora como dos módulos independientes). Después de concluir la transferencia, la LCC continuará la investigación de la palabra de control.



Module address indicated when a set bit is encountered in the control word

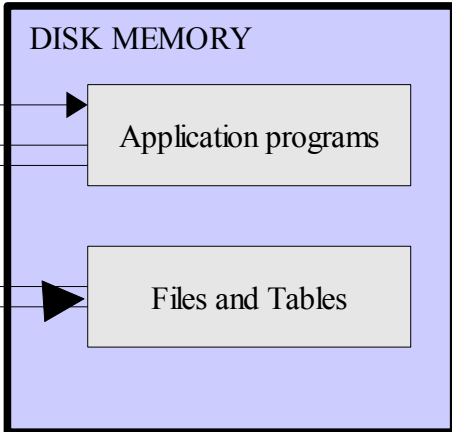
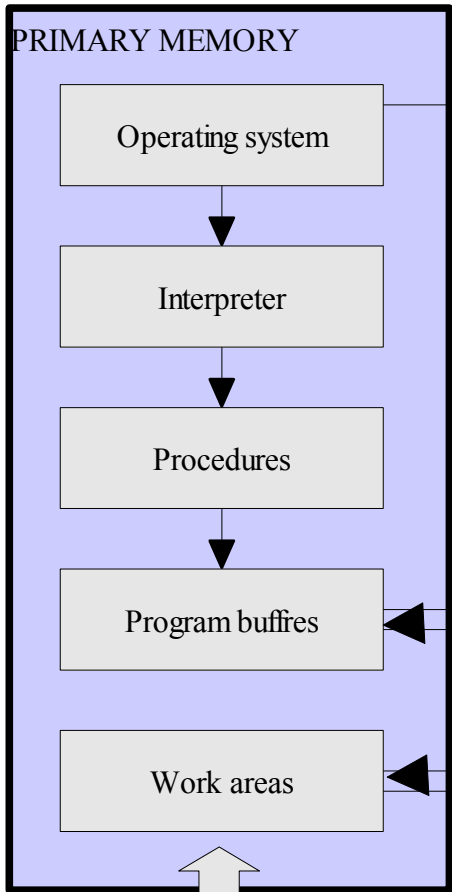
Data IN/OUT

Interrupt when the entire control word has been investigated



Cuando se ha efectuado la investigación completa de la palabra de control, se emitirá otra palabra también de control mediante una llamada de la LCC. Según se va ejecutando el programa podría haber cambiado la necesidad de contacto con varios módulos. Por lo tanto, la nueva palabra de control podría contener nuevos bits en situación de "1". La computadora establecerá contacto con los diferentes módulos durante la subsiguiente exploración de la nueva palabra de control. El tiempo de exploración y de ejecución será tan insignificante que cada operador en su estación trabajará como si dispusiera de la computadora únicamente para su uso.

Tanto la LCC como la TCC, incluyen un modulador y circuitos para el chequeo de la longitud de los caracteres y el chequeo de paridad. Estas unidades manejan los procedimientos de transferencia independientemente, incluyendo una repetición cuando se detecta un error de paridad. Sólo cuando el error es permanente se informará a la computadora de la situación anormal.



DESCRIPCIÓN DEL SISTEMA: SOFTWARE

Introducción

El sistema de programación Facit 6501, está diseñado en su totalidad para un diálogo terminal-computadora. Los programas de aplicación están en memoria de discos y son requeridos en segmentos. Esto significa que los programas de aplicación son totalmente independientes de las limitaciones de capacidad de la memoria primaria.

Se ha desarrollado un lenguaje especial de programación denominado LOGIC-3. Este es un lenguaje orientado a las aplicaciones para una fácil programación de las distintas rutinas que se adapta al resto del sistema, dando como resultado un juego de instrucciones muy potentes. El diseño general que se representa en la página anterior sólo era posible para grandes sistemas, pero debido a la nueva tecnología de hardware y la filosofía de las estaciones de trabajo en cuanto a hardware y software, han hecho posible el diseño del sistema tal como se describe.

El software incluye un sistema operativo, un interpretador de LOGIC-3 y un paquete de procesos.

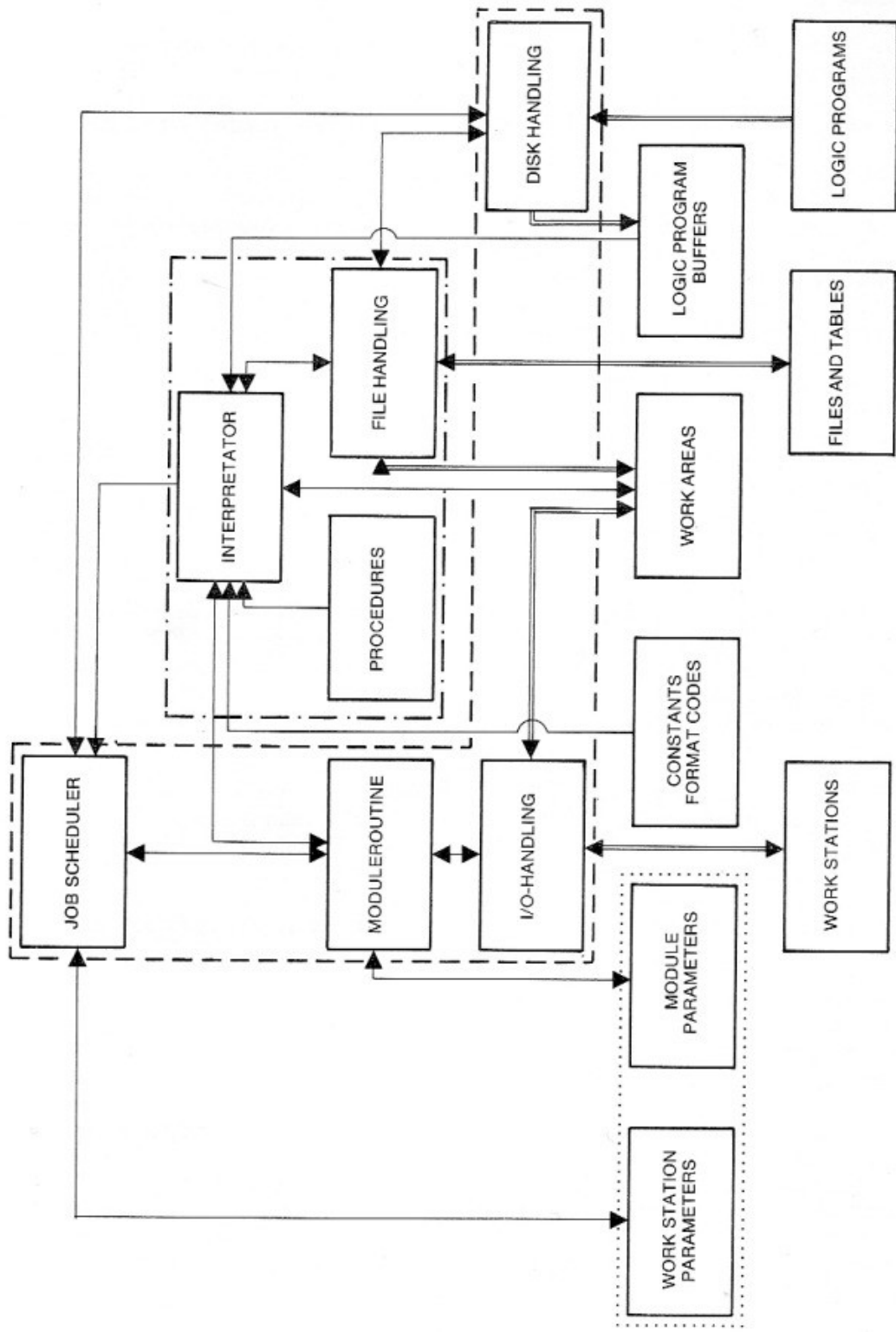
El sistema operativo explora la situación de las estaciones de trabajo y administra la transferencia de datos entre las estaciones y la computadora, y entre la computadora y las unidades de discos. También se encarga de llamar a los programas de aplicación entre los discos y distribuir el tiempo de la CPU entre las diferentes estaciones de trabajo e iniciar la interpretación del LOGIC-3.

Interpretador del LOGIC-3. Este, como su nombre indica, interpreta las instrucciones y tiene acceso a la información de las constantes y parámetros para las distintas estaciones de trabajo. Utilizando esta información llamará a la rutina de proceso que ejecute la instrucción. De esta forma, las instrucciones son interpretadas junto con la ejecución real y no hay necesidad de una compilación de tipo tradicional.

El paquete de procesos consiste en subrutinas para las varias instrucciones de LOGIC-3. Se carga la memoria primaria conjuntamente con el sistema operativo del interpretador. Para hacer que las instrucciones del LOGIC-3 sean potentes algunas de estas subrutinas pueden ser bastante extensas, como por ejemplo, la rutina de SORT-FILE (Clasificación de un fichero). Cada procedimiento es iniciado por el interpretador en conexión con la interpretación de las instrucciones LOGIC-3. Así, cada instrucción LOGIC-3, en el momento de la ejecución, se convierte en un proceso que comprende un conjunto de instrucciones máquina.

Las instrucciones de LOGIC-3 son cargadas en buffers de memoria predeterminados con el funcionamiento del sistema y las instrucciones máquina se ejecutan sin tener que activar una transferencia desde disco. El diseño del sistema de programación permite una fácil reconfiguración de una instalación en cuanto a las estaciones de trabajo, capacidad de memoria, número de programas de aplicación, ficheros y organización de ficheros. Se crea una estructura general de una instalación Facit 6501, mediante la utilización de un programa especial disponible denominado "generación del sistema". Esta generación del sistema requiere ciertos datos de entrada especificando el tipo de información que acabamos de mencionar. Mediante la introducción de los datos necesarios, el programa genera los parámetros de la creación del sistema, significando esto que puede cambiar o ampliarse el sistema fácilmente para ajustarse a los requisitos variables.

SOFTWARE CONNECTION



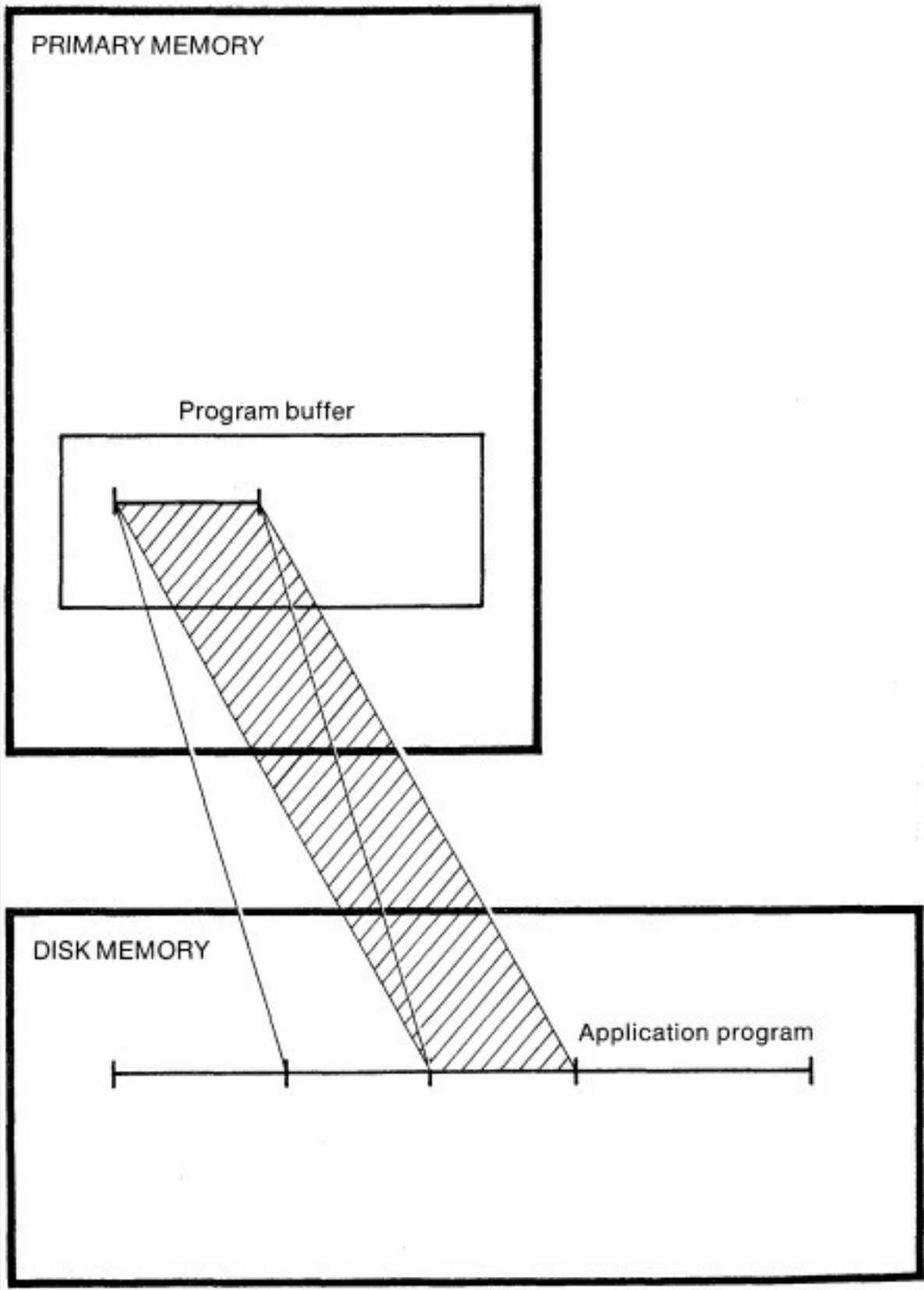
- Operating System
- . - Interpreter
- System parameters

- ← Program connection
- ⇄ Data flow

Programas de utilidad

El software del sistema incluye un número de programas de utilidad para el desarrollo, verificación y activación de programas y sistemas de programa. Están disponibles los siguientes programas de utilidad:

MONITOR	Carga de programas del sistema desde los discos.
LOGED	Programa de edición para instrucciones LOGIC-3.
TRIC	Ensamblador de las instrucciones LOGIC-3 a código máquina.
FG	Programa de generación de ficheros y tablas.
TOS	Sistema operativo de test; por ejemplo, un sistema operativo especial para la verificación de los programas en LOGIC-3.
SG	Programa para la generación del sistema.
DCY	Programa para efectuar copias de discos.



El programa, que está en la memoria de discos, se transfiere a la memoria primaria en segmentos que se adaptan al buffer de programa.

Sistema operativo

El sistema operativo realiza las siguientes funciones:

- Exploración continua de la situación de las estaciones de trabajo (mediante el sistema de comunicación) y administración de los conjuntos de caracteres para las operaciones I/O.
- Llamada al disco para los programas de aplicación y segmentación de éstos llevándolos a buffers predeterminados.
- Distribución del tiempo CPU a las distintas estaciones de trabajo.
- Inicia al interpretador para la ejecución de instrucciones LOGIC-3.
- Administra la transferencia de datos en conexión con los accesos de la memoria de discos.
- Interpreta los mensajes del operador e inicia las acciones debidamente.
- Controla los fallos de corriente y maneja las rutinas de puesta en marcha.
- Situación de recursos y ejecución de programas lógicos.

El sistema operativo incluye un programador de trabajo que controla la ejecución de instrucciones de las distintas estaciones. Debido a que la ejecución es considerablemente más rápida que, por ejemplo, las operaciones del teclado o impresión, varios operadores de las estaciones de trabajo pueden activar rutinas individuales simultáneamente. Por lo tanto, el sistema operativo controla la ejecución de varios programas individualmente. Se ejecuta una o varias instrucciones de cada estación de trabajo a la vez y, después de lo cual, la siguiente estación de trabajo está servida en la misma forma. Debido a que la ejecución es tan rápida a cada operador le parecerá que su programa es el único que se está ejecutando. No existirá retraso alguno causado por la ejecución de otros programas.

El sistema operativo es utilizado conjuntamente por todas las estaciones de trabajo. Cuando le presta servicio a una estación de trabajo determinada, investigará si el operador ha emitido alguna orden: por ejemplo, para la carga de otro programa. Si no hay orden el sistema de funcionamiento pasará a la próxima estación de trabajo, etc. etc.

De esta forma el sistema operativo le prestará en un momento determinado servicio a una estación de trabajo, posteriormente a otra y así sucesivamente irá investigando entre todas las estaciones. Para conseguir lo que ha de realizar el programa en las distintas estaciones de trabajo, por ejemplo, ejecutar instrucciones lógicas, buscar programas lógicos, etc., se le designa a cada estación de trabajo un campo de datos especiales que controla el sistema operativo. Este campo de datos contiene los parámetros de las diferentes estaciones de trabajo. Además, existe una área y un buffer de programa lógico para cada estación de trabajo.

Los parámetros de las estaciones de trabajo, contienen información sobre la localización del programa lógico requerido, una dirección de la instrucción lógica actual, e información sobre lo que debe hacerse a continuación. Cuando se ejecuta una instrucción lógica el sistema operativo llamará al intérprete, al cual se le proporciona acceso a los parámetros que identifican la estación de trabajo en cuestión y la dirección de la instrucción lógica actual, así como la dirección de la instrucción lógica que se debe ejecutar.

El intérprete ejecuta la orden ocasionada por esta instrucción. Una vez realizado señala que desea tomar la próxima instrucción. El sistema operativo investiga si esta instrucción está cargada en el buffer de programa de la estación de trabajo o bien está disponible en la memoria de discos. Si la instrucción está en el disco el sistema operativo ordenará la

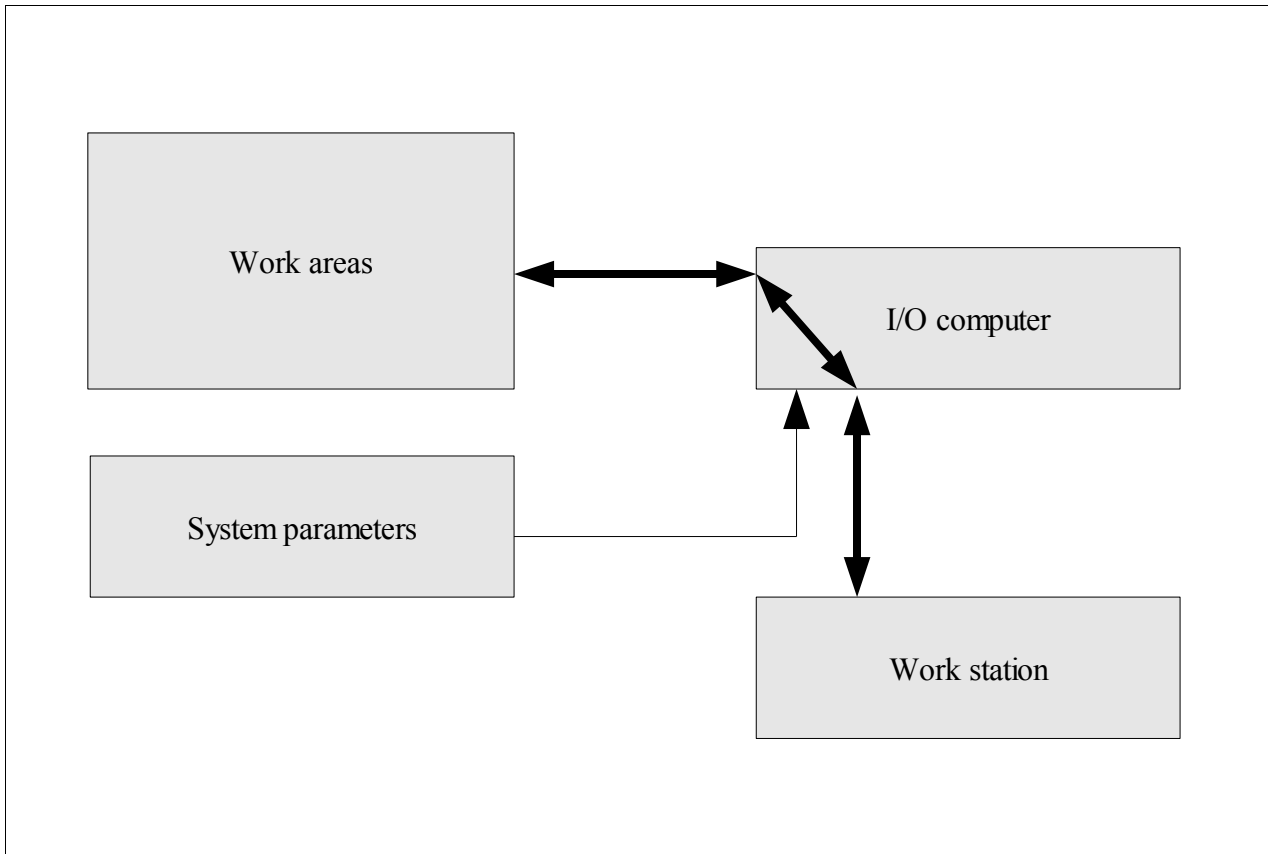
recogida del adecuado segmento del programa en el disco y lo cargará en el buffer.

Por lo tanto, en la práctica, el tamaño de un programa lógico no tiene límite, ya que se almacena en el disco y sólo se recogen los segmentos relativos de uno en uno. Cuando se ha de ejecutar una instrucción fuera del segmento cargado en este momento, dicha parte del programa es segmentada y recogida del disco y llevada al buffer de programa lógico en la memoria primaria.

El intérprete continúa ejecutando instrucciones hasta que ocurre una de las siguientes situaciones:

- Se han ejecutado cinco instrucciones.
- La instrucción a ejecutar no está cargada en el buffer de programa.
- Se está ejecutando una instrucción de disco.
- Se está ejecutando una instrucción que requiere un estado de espera para operaciones I/O.
- El programa está concluido.

Cuando ocurre una de estas situaciones, el sistema operativo empezará a prestar servicio a la siguiente estación de trabajo.



Operaciones I/O

Para poder comprender el significado del principio del funcionamiento de entrada/salida (I/O), imaginemos que todos los caracteres son transferidos por una computadora especial, digamos la computadora I/O, que funciona con las mismas posibilidades de memoria que las de la computadora real.

La computadora I/O entrará en acción cuando se introduce un carácter o cuando un dispositivo de salida está preparado para recibir un carácter. Recibe sus datos de control de campos especiales, uno por cada unidad I/O. Estos campos así como los parámetros de módulo informan a la computadora I/O de cuál área de trabajo deberá recibir el carácter o proporcionarlo para la salida, a qué estación de trabajo está conectada la unidad y cuándo se debe concluir la transferencia.

Cuando un programa lógico requiere una transferencia, el intérprete informa al sistema operativo de esto. En consecuencia, el sistema operativo colocará la información necesaria en los parámetros de módulo y pondrá la computadora I/O en funcionamiento. Esta utilizará la información para conocer de qué módulo, al cual ha solicitado la I/O y qué área es la que ha de recibir o proporcionar datos.

La computadora I/O verifica que se haya concluido la operación I/O y avisa a la estación de trabajo, mediante los parámetros de módulo y los parámetros de la estación de trabajo. El sistema operativo recibe información indicando que la transferencia ha sido concluida y desconecta la computadora I/O. Durante todas las operaciones I/O, la computadora detiene temporalmente todas las demás actividades. Cuando haya sido desconectada la computadora I/O, se renovará la ejecución de instrucciones lógicas.

Intérprete

El intérprete es una unión entre el sistema operativo y el "paquete" de procesos. El sistema operativo llama al intérprete cuando se ha de ejecutar una instrucción lógica y simultáneamente se le proporciona una llegada a los parámetros de acceso para la estación de trabajo; por ejemplo, las direcciones de la instrucción lógica, los parámetros de la instrucción de trabajo, definición de las áreas y parámetros de módulo. Una serie de parámetros del sistema son comunes a todas las estaciones de trabajo. Incluyen la dirección de constantes, formatos y definiciones de ficheros y tablas.

El intérprete analiza la instrucción lógica de acuerdo con cierto patrón, según el tipo de instrucción en cuestión. El resultado de este análisis es, con la ayuda de los parámetros del sistema, la construcción de las direcciones y demás información necesaria para la ejecución de la instrucción. A continuación se llama a la rutina de proceso que debe ejecutar la instrucción. Una vez ejecutado el proceso el intérprete calculará la dirección de la subsiguiente instrucción a ejecutarse para la estación de trabajo en particular y, a continuación, devolverá el control al sistema operativo.

Cuando se ejecuta una instrucción se realiza un número de verificaciones como puede ser la verificación de que no se haya excedido ningún límite de área. Si se detecta un error se recibirá un mensaje. Si el error es insignificante se concluye la ejecución como es usual, pero un error serio podría abortar el programa.

Generación del sistema

La generación del sistema comprende la creación del software y del sistema de test, así como de los distintos trabajos según se explica a continuación. Son especificadas y registradas siguiendo un formato según las características de la instalación incluyendo, tanto el hardware como el software. Esta descripción es traducida por el programa de generación de sistema a parámetros y direcciones que están en conexión con el software del sistema; por ejemplo, el sistema operativo o sistema operativo de chequeo, el interpretador y el paquete de procesos. Cuando se concluye una generación de sistemas de chequeo se debería registrar la lectura de programas lógicos, así como los ficheros y tablas a incluir en el sistema. Cuando se finaliza una generación del sistema operativo, el sistema está listo para su utilización.

Se necesitan las siguientes especificaciones para una generación del sistema:

- Tipo de generación, por ejemplo, generación del sistema operativo (funcional) o generación del sistema operativo de chequeo.
- Dirección final en la memoria primaria según el tamaño de la memoria primaria.
- Número de niveles de subrutinas.
- Descripción de las capacidades de discos.
- Definición del número y longitud de los programas.
- Cualquier instrucción opcional (sólo en la generación del sistema operativo funcional ya que el sistema operativo de chequeo incluye todas las instrucciones como estándar).
- Constantes que utilizarán los programas lógicos.
- Formatos de impresión que utilizarán los programas lógicos.
- Descripción de los módulos y tipos de módulos.
- Descripción de cada estación de trabajo en cuanto se refiere a los módulos conectados y al buffer de trabajo.
- Definición de ficheros incluyendo el tipo de ficheros, el número de registros y longitudes de los registros.
- Definición de las tablas, por ejemplo, número de elementos de las tablas y longitudes de éstas.
- Reserva de los buffers de programa o buffers de trabajo para cada estación de trabajo.

Programas de utilidad

- LOGED (Editor de LOGIC)

Programa editor para escritura o conversión y corrección de instrucciones de LOGIC-3. en lenguaje fuente sobre cinta de papel. El programa perfora el lenguaje fuente en un formato adaptado para el TRIC.

- TRIC (Traductor de LOGIC)

Programa de traducción que convierte las instrucciones de LOGIC-3, en lenguaje fuente a palabras lógicas en forma binaria y prepara un listado del programa contenido en direcciones de instrucciones lógicas, claves hexadecimales para las instrucciones de LOGIC-3 e impresiones de las correspondientes instrucciones en lenguaje fuente.

TRIC, es un traductor de dos fases.

El lenguaje objeto se perfora en cinta de papel en un formato adoptado para la entrada al sistema operativo de chequeo.

- TOS (Sistema operativo de verificación)

El sistema de verificación para los programas LOGIC-3, incluyendo varias estaciones de trabajo, consiste en un sistema ordinario FACIT 6501, complementado con una parte operativa especial.

El operador de cada estación de trabajo puede verificar y cambiar el contenido de áreas, ficheros, tablas o instrucciones de programa. Se pueden ejecutar los programas de LOGIC-3 bit por bit, paso a paso y puede realizarse un trazado de estos programas. Se puede interrumpir automáticamente la ejecución cuando se utiliza una o varias áreas determinadas. Además, pueden cargarse o descargarse programas mediante la cinta perforada.

- FG (Programa generador de ficheros y tablas)

Programa para la entrada de ficheros y tablas desde cinta de papel a la memoria de discos. Cada fichero (tabla) es perforado individualmente sobre una o más cintas. Los registros se pueden perforar en cualquier secuencia. Luego éstos se almacenan clasificados si esto es necesario o se requiere en el programa de entrada.

LOGIC-3 (Juego de instrucciones)

Instrucción	Función	Instrucción	Función
START	Poner el programa lógico en marcha.	GEN	Generar caracteres en una zona
ADEF	Definición de área	GET-REC	Leer un registro de un fichero indexado o secuencial.
IN	Entrada por un módulo.	PUT-REC	Grabar un registro o un fichero indexado o secuencial cancelando el bloqueo.
INO	Entrada por un módulo con salida concurrente en otro.	UPD-REC	Leer un registro de un fichero secuencial o indexado bloqueando el registro.
OUT	Salida por un módulo.	NEW-REC	Grabar un nuevo registro en un fichero indexado.
LOCK	Bloqueo de un módulo.	SUC-REC	Leer el siguiente registro de un fichero indexado.
UNLOCK	Desbloqueo de un módulo.	DEL-REC	Eliminar un registro de un fichero indexado.
WAIT	Espera de entrada/salida.	BUILD-LEX	Construir un diccionario para un fichero indexado.
MOVE	Mover una zona.	DEL-FILE	Eliminar un fichero secuencial.
CONV	Convertir una zona.	SORT-FILE	Clasificar un fichero secuencial.
ADD	Suma.	GET-TABLE	Tomar un elemento de una tabla.
SUB	Resta.	UPD-TABLE	Leer un elemento de una tabla bloqueando este elemento.
MULT	Multiplicación.	PUT-TABLE	Grabar un elemento de una tabla cancelando el bloqueo.
DIV	División.	GET-WS	Tomar el número de la estación de trabajo.
ADD-BIN	Sumar en binario.	GET-EC	Tomar el carácter final.
SUB-BIN	Restar en binario.	GET-NIC	Tomar el número de caracteres entrados
SQR	Raíz cuadrada.	GET-CC	Tomar código de error.
SHIFT	Desplazar.	MOVE-WORD	Mover una palabra
ROUND	Redondear.	END	Concluir un programa lógico
ROTATE	Rotación.	DO	Ejecutar la instrucción lógica situada en un área.
EDIT	Editar una zona.	CALL	Bifurcar a subrutina.
JUMP	Bifurcación incondicional.	RETURN	Retorno de una subrutina.
JINE	Bifurcación si la entrada no es igual		
JIG	Bifurcación si la entrada es mayor que...		
CDV	Comprobar y verificar el dígito de control bifurcando si hay error.		
JEQ	Bifurcación si es igual a...		
JGE	Bifurcación si es mayor o igual a...		
JNE	Bifurcación si es desigual.		
JLT	Bifurcación si es menor que...		
CASE	Bifurcación si el primer byte de una zona es igual al carácter especificado en la instrucción.		

LOGIC-3 Resumen

Introducción

LOGIC-3 es un idioma interpretativo orientado a las aplicaciones muy adecuado para un gran número de rutinas para el procesamiento de datos administrativos.

El tamaño del programa, no depende de la capacidad de la memoria primaria, ya que los programas están almacenados en código binario, en la memoria de discos y la sección de programa que se necesita en un momento dado se segmenta y transfiere a la memoria primaria para su ejecución.

Un juego simple y potente de instrucciones facilita la confección de programas. El lenguaje trabaja sobre zonas completas o bien sobre partes de zona, tanto en la memoria primaria como en los discos.

Se puede variar la longitud y tipo de las zonas durante la ejecución de un programa. Los datos variables, se almacenan en áreas, ficheros y tablas, mientras los datos permanentes se memorizan como constantes y formatos. La memoria primaria contiene áreas para el uso exclusivo individual de las estaciones de trabajo.

Los ficheros y tablas se almacenan en los discos y pueden ser utilizados concurrentemente por todos los programas y estaciones de trabajo incluidos en el sistema. Las constantes y formatos están almacenados en la memoria primaria y pueden asimismo ser utilizados por todos los programas y estaciones de trabajo del sistema.

El manejo de ficheros incluye: ficheros organizados secuencialmente, ficheros organizados indexados secuencial y tablas que son ficheros secuenciales directamente direccionables.

Estructuras de los programas

Una instrucción de LOGIC-3 consiste en un código de operación y varios operandos.

Los códigos de operación llevan nombres en inglés:

ADD, MOVE, CONV, EDIT... etc.

Los operandos tienen nombres permanentes.

Los datos se almacenan en dieciséis áreas-

A-0, A-1 ... A-15.

Se puede definir un área con una longitud máxima de 255 bytes (1byte = 8 bits).

LEN-0, LEN-1, ... LEN-255

Se puede definir un área como numérica: HEX-00 o alfanumérica: HEX-20.

Una instrucción puede operar en una parte de un área especificando un LIMIT.

LIM-1-1 ... LIM-1-255.

Se pueden ejecutar bifurcaciones a 1000 etiquetas distintas.

L-0, L-1 ... L-999.

Una instrucción I/O puede dirigir ocho módulos relativos.

M-0, M-1 ... M-7

En cada byte de un área se puede generar un valor binario.

HEX-00, HEX-01 ... HEX-FF

Se puede especificar la cantidad de caracteres a generar en un área.

Q-1, Q-2 ... Q-127

Los formatos de edición almacenados son como máximo 16.

FORMAT-0, FORMAT-1 ... FORMAT-15

Los datos universalmente válidos se almacenan en 16 constantes.

CONST-0, CONST-1 ... CONST-15

Pueden llamarse hasta 128 programas de LOGIC-3

LP-0, LP-1 ... LP-127.

Pueden direccionarse hasta 127 ficheros secuenciales e indexados secuenciales.

F-1, F-2 ... F-127

Pueden direccionarse indirectamente hasta 32767 registros en un fichero secuencial, con un número hexadecimal almacenado en los dos primeros bytes de un área opcional.

Pueden direccionarse directamente hasta 32767 registros en un fichero indexado secuencial, mediante el concepto de identificación almacenado en un área opcional.

Pueden direccionarse hasta 16 tablas.

T-0, T-1 ... T-15

Dirección indirecta de hasta 32767 elementos en una tabla, con el número del elemento almacenado en forma hexadecimal en los dos primeros bytes del área A-0.

Dirección directa de hasta 255 elementos.

E-1, E-2 ... E-255

Las especificaciones para las longitudes de registros y elementos son las mismas que para las áreas.

Funciones del programa

LOGIC-3 comprende las siguientes funciones, las cuales simplifican considerablemente el desarrollo del programa.

Instrucciones de ficheros

Potentes instrucciones que funcionan sobre registros y sobre ficheros completos.

Protección de registros

Facilidad de proteger un registro contra "su puesta al día", para evitar un acceso simultáneo al registro desde otra estación de trabajo.

Clasificación

Facilidad de clasificar todo un fichero en orden ascendente o descendente, por argumento (hasta 8), solapando sin tener en cuenta la estructuración del registro.

Verificaciones

Facilidad de investigación o anulación del estado actual de un parámetro del sistema en un área:

Facilidad de investigación o anulación del estado actual de un parámetro del sistema en un área:

- el número de caracteres recibidos desde un módulo,
- los caracteres finales que se recibieron desde un módulo,
- el número de la estación de trabajo,
- el código de error de la estación de trabajo.

Subrutinas

Facilidad para la llamada de subrutinas. Los programas lógicos y partes de programas lógicos pueden ser subrutinas. El número de niveles de subrutina son variables y se determina durante la generación del sistema.

Operaciones I/O

Facilidad de solapar las operaciones I/O acoplando la instrucción con una función de ESPERA o eliminándola. Cuando se incluye una función de ESPERA se concluye la instrucción I/O, antes de que el programa proceda a la subsiguiente instrucción. Sin una función de ESPERA el programa prosigue sin aguardar a que haya terminado la requerida operación I/O.

Entrada

Facilidad para que se ejecute la entrada de datos desde un módulo antes de que el programa haya alcanzado la instrucción en cuestión. Los caracteres que entran se almacenan en un buffer hasta que el programa llegue a la instrucción que requiere la entrada. A continuación, se mueve el contenido al área especificada en la instrucción. Sólo se puede realizar una operación de entrada sin iniciar a la vez.

Bloqueo y desbloqueo de módulos

Facilidad para el bloqueo y desbloqueo de un módulo de entrada. Cuando un módulo de entrada está bloqueado, se impide cualquier posibilidad de manejo.

D.L.: B. 46229-1973 Socitra - Salvadors, 22 - Barcelona

GISPERT, s.a.

Automación de la gestión empresarial

Sistemas - Equipos - Servicio