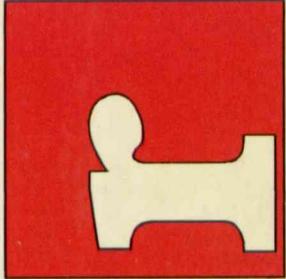


# INTRODUCCION AL DISEÑO ELECTRONICO

Serie "Recursos para la formación"

O  
C  
A  
D





# Presentación

# INTRODUCCION AL DISEÑO ELECTRONICO

A lo largo de la fase experimental de los Proyectos Atenas y Mercurio, la Secretaría de Estado de Educación, a través del Programa de Nuevas Tecnologías de la Información y la Comunicación, ha puesto a disposición del profesorado una serie de publicaciones entre las que se encuentra la titulada "Presentación". La experiencia acumulada durante esta fase ha llevado al Programa de Nuevas Tecnologías a matizar aspectos importantes de la utilización de las nuevas tecnologías en las actividades docentes. La publicación de la L. O. G. S. E. ha abierto nuevos horizontes para el enfoque de la labor docente por parte de los profesores y profesoras. Todo ello nos ha impulsado a iniciar una nueva etapa en la serie "Recursos para la formación", que ve la luz en el momento en que la reforma del Sistema Educativo comienza el proceso de implantación, tras un largo período de debates.

Esta serie de publicaciones se orienta a aportar formación que haga posible la utilización en el aula de materiales que utilizan nuevas tecnologías en el desarrollo de los nuevos diseños curriculares. De esta manera se intenta contribuir a enriquecer las posibilidades de elección del profesorado a la hora de planificar y llevar a cabo su labor docente.

Por esta razón, el eje conductor de la serie se define desde la perspectiva de las nuevas tecnologías entendidas como recursos que pueden ser utilizados en el diseño y desarrollo de actividades de enseñanza y aprendizaje, todo ello insertado en áreas curriculares concretas.



Ministerio de Educación y Ciencia

la información en Educación y Ciencia

de Estado

de Nuevas Tecnologías de la Información y la Comunicación

178-91-007-0

04-888-18021

tel. 34-20037-1001

Ministerio de Educación y Ciencia

de Educación

de Estado

de Educación y Ciencia



# INTRODUCCIÓN AL DISEÑO ELECTRÓNICO



---

**Ministerio de Educación y Ciencia**

---

Secretaría de Estado de Educación

---

*Programa de Nuevas Tecnologías de la Información y la Comunicación*

N. I. P. O.: 176-91-007-0

I. S. B. N.: 84-369-1983-1

Depósito legal: M-20831-1991

Imprime: MARIN ALVAREZ HNOS.

## **Presentación**

*A lo largo de la fase experimental de los Proyectos Atenea y Mercurio, la Secretaría de Estado de Educación, a través del Programa de Nuevas Tecnologías de la Información y la Comunicación, ha puesto a disposición del profesorado diferentes series de publicaciones entre las que se encuentra la titulada "Recursos para la formación". La experiencia acumulada durante esta fase ha llevado al Programa de Nuevas Tecnologías a matizar aspectos importantes de la formación del profesorado que utilice las nuevas tecnologías en las actividades de aula y, por otro lado, la aprobación de la L. O. G. S. E. ha abierto nuevos horizontes para el enfoque de la labor de profesores y profesoras. Todo ello nos ha impulsado a iniciar una nueva etapa en la serie "Recursos para la formación", que ve la luz en el momento en que la Reforma del Sistema Educativo comienza el proceso de implantación, tras un amplio periodo de debate.*

*Esta línea de publicaciones se orienta a aportar formación que haga posible la utilización en el aula de materiales que utilizan nuevas tecnologías en el marco de los nuevos diseños curriculares. De esta manera se intenta contribuir a enriquecer las posibilidades de elección del profesorado cuando planifica y lleva a cabo su labor docente.*

*Por esta razón, el eje conductor de la serie se define desde la perspectiva de las nuevas tecnologías entendidas como recursos que pueden ser contemplados en el diseño y desarrollo de actividades de enseñanza y aprendizaje, todo ello insertado en áreas curriculares concretas.*

*En estas publicaciones se opta por una aproximación a una estructura modular, definiendo características y apartados comunes a todas ellas que faciliten su adaptación a distintos cursos de formación del profesorado, a la práctica de aula y a la reflexión sobre los medios, a los aspectos pedagógicos y a los más técnicos. El uso de soportes multimedia, siempre que ello sea posible, es otra de las características que aumentará su presencia y relevancia en la nueva etapa de esta serie.*

*El esfuerzo que supone la publicación de esta serie se verá ampliamente justificado, si con ella se contribuye a que sus destinatarios, profesores y profesoras de los distintos niveles y áreas educativas, dispongan de una fuente de sugerencias y orientaciones en el proceso de innovación de la enseñanza y en el aumento de la calidad de la misma.*

Enero, 1991

**Elena Veiguela Martínez**

Directora del Programa de Nuevas Tecnologías  
de la Información y de la Comunicación



Ministerio de Educación y Ciencia

Secretaría de Estado de Investigación Científica y Tecnología

Programa de Nuevas Tecnologías de la Información y de la Comunicación

N.º C.P. O. 176-91 00-0

1 5 8 5 N - 84 386 000

Deposito legal: M. 1000-1991

Impreso en España

# Índice

	<i>Página</i>
ORCAD .....	7
AUTÓMATAS PROGRAMABLES .....	233
INTRODUCCIÓN A LA ROBÓTICA .....	299

ORCAD



# Índice

	<i>Página</i>
<b>CAPÍTULO 1 ORCAD SDT/III</b> .....	15
Características específicas de OrCAD/SDT .....	17
Requisitos hardware .....	21
Contenidos del paquete .....	23
Instalación del paquete .....	26
Instalación en diskettes .....	26
Instalación en disco duro .....	31
Configuración de OrCAD SDT/III .....	37
Prefijo de drive .....	38
Dispositivo de pantalla .....	38
Dispositivo de impresora .....	39
Dispositivo de plotter .....	39
Prefijo de librería .....	40
Archivos de librería .....	40
Prefijo de hoja de trabajo .....	41
Archivo de macros .....	42
Macro inicial .....	43
Tamaño del buffer de trazos .....	44
Tamaño del buffer de jerarquía .....	44
Paleta de colores .....	45
Tabla de tamaño .....	45
Campo llave .....	45
Actualizando la configuración .....	49
Filosofía de trabajo de OrCAD/SDT III .....	51
Archivo plano .....	52
Archivo jerárquico .....	52
Archivo simple .....	54





# Índice

	<i>Página</i>
CAPÍTULO 1. ORCAD SDT/III .....	15
Características específicas de OrCAD/SDT .....	17
Requisitos hardware .....	21
Contenidos del paquete .....	23
Instalación del paquete .....	25
Instalación en diskettes .....	25
Instalación en disco duro .....	31
Configuración de OrCAD SDT/III.....	37
Prefijo de driver.....	38
Dispositivo de pantalla .....	38
Dispositivo de impresora .....	39
Dispositivo de plotter .....	39
Prefijo de librería.....	40
Archivos de librería.....	40
Prefijo de hoja de trabajo .....	41
Archivo de macros.....	42
Macro inicial.....	43
Tamaño del buffer de macros.....	44
Tamaño del buffer de jerarquía .....	44
Paleta de colores.....	45
Tabla de tamaños .....	47
Campo llave.....	48
Actualizando la configuración .....	49
Filosofía de trabajo de OrCAD/SDT III .....	51
Archivo plano .....	52
Archivo jerárquico .....	52
Archivo simple .....	54

	<i>Página</i>
CAPÍTULO 2. COMANDOS DE ORCAD/SDT III .....	57
Arrancando Draft .....	59
El menú principal de Draft .....	61
Abandonando Draft .....	63
Adaptando Draft a nuestras necesidades .....	65
El comando establecer .....	67
Autopanning .....	67
Archivo de seguridad .....	67
Reconstrucción de buses .....	67
Sonido de error .....	67
Activación del botón izquierdo del ratón .....	68
Macro Prompts .....	68
Ortogonalidad .....	68
Mostrar numeración de pines .....	68
Bloque de título .....	68
Tamaño de la hoja de trabajo .....	69
Mostrar coordenadas .....	69
Parámetros de plantilla .....	69
Referencias de plantilla .....	70
Permanecer en plantilla .....	70
Visualizar puntos de la plantilla .....	71
Repetición de parámetros .....	71
Repetir abcisa .....	71
Repetir ordenada .....	71
Repetir etiqueta .....	72
Colocar con autoincremento .....	72
Visualizar rótulos .....	72
CAPÍTULO 3. UNA SESIÓN CON DRAFT .....	73
¿Qué se necesita para comenzar? .....	75

Comprobaciones previas .....	75
Personalización del comando Set .....	76
Cantidad de RAM disponible .....	76
Cantidad de memoria de hoja de trabajo .....	76
Cantidad de memoria del buffer de jerarquía .....	77
Cantidad de memoria del buffer de macros.....	77
Cantidad de memoria libre del sistema.....	77
El comando Library.....	78
Un método de trabajo .....	81
Paso 1: Extracción y colocación de componentes .....	82
El comando Get .....	82
El menú de colocación de componentes.....	85
Aplicación práctica .....	86
Cómo borrar.....	87
Paso 2: Conexionado de componentes.....	93
Macros .....	94
Conexionado de resistores .....	97
Conexionado de condensadores.....	98
Conexionado de semiconductores.....	99
Conexiones varias .....	99
Paso 3: Colocación de uniones .....	100
Paso 4: Colocación de terminales.....	103
Paso 5: Colocación de texto y líneas auxiliares.....	106
El comando Edit.....	106
Los comandos Tag y Jump.....	107
Macros de edición .....	108
Edición y modificación del valor de un componente .....	109
Edición de diodos .....	111
Edición del resto de componentes.....	111
Rellenado del cajetín .....	112

	<i>Página</i>
Recuadrando partes del esquema.....	112
Algunas consideraciones sobre la alimentación .....	113
Paso 6: Modificación y posibles mejoras.....	114
Operaciones con bloques .....	114
Subcomandos del bloque.....	115
Algunos retoques .....	120
Retoques finales .....	121
Modificaciones finales.....	122
Colocando un bus.....	123
Especificación de un bus.....	124
Construcción de un esquema jerárquico .....	126
Construcción del esquema RAIZ.SCH.....	128
Conexión de módulos .....	131
CAPÍTULO 4. EL EDITOR DE COMPONENTES.....	133
Aplicación práctica de modificación de componentes .....	135
Una sesión con Libedit .....	137
Cambio de referencia .....	139
Cambio de nombre .....	139
Incorporación del componente modificado a una determinada librería.....	141
Incorporación a la misma librería.....	141
Incorporación a otra librería existente .....	142
Incorporación a una librería propia.....	142
CAPÍTULO 5. UTILIDADES.....	145
Generalidades .....	147
Estructura genérica de una utilidad .....	149
Switches de carácter general.....	149
Repertorio de utilidades .....	151
Annotate .....	151

	<i>Página</i>
Backanno .....	152
Cleanup.....	153
Ercheck.....	155
Partlist.....	156
Netlist.....	157
Libarch .....	168
Liblist.....	169
Plotall .....	170
Printall.....	172
Crossref.....	173
Fldattrb.....	174
Fldstuff.....	175
Extract.....	176
Simple.....	176
Treelist .....	177
Xferovl .....	179
CAPÍTULO 6. APLICACIÓN PRÁCTICA DE ORCAD/VST.....	181
Requisitos previos .....	183
Arrancando SIMULATE .....	185
Generación de impulsos .....	187
Muestreo.....	191
Ejecutando la simulación .....	195
CAPÍTULO 7. APLICACIÓN PRÁCTICA DE ORCAD/PCB II.....	199
Introducción .....	201
Configuración.....	203
El comando Set.....	211
El comando Quit .....	213

Diseño de componentes.....	215
El menú de Place.....	219
Otros comandos de Place .....	221
El menú de Routing .....	223
Modo manual.....	223
Modo automático.....	225
Strategy .....	226
Otros comandos de PCB II .....	229

CAPÍTULO 4. EL DISEÑO DE COMPONENTES

Aplicación práctica de edición de componentes.....	231
Los seales son Libedit.....	231

CAPÍTULO 6. APLICACION PRÁCTICA DE ORCAD VST

Cambios de datos.....	181
Cambios de atributos.....	183
Arrendando SIMULATE.....	185
Generación de impulsos.....	187
Muestreo.....	191
Ejecutando la simulación.....	195

CAPÍTULO 7. APLICACION PRÁCTICA DE ORCAD PCB II

legislación.....	245
Configuración.....	249
El comando Set.....	251
El comando Cut.....	251



## Características específicas de OrCAD/SDT III

OrCAD/SDT III es un paquete de Diseño de Esquemas Electrónicos completo, flexible, que mediante un sistema de menús de fácil acceso, permite crear, editar, salvar, imprimir y plotear esquemas electrónicos, chequearlos, prepararlos para el trazado automático de pistas, crear componentes, numerarlos automáticamente, listarlos, etc.

Podemos considerarlo dividido en tres bloques fundamentales:

- Un editor de esquemas (DRAFT).
- Un editor de componentes (LIBEDIT).
- Un conjunto de utilidades que complementan al editor de esquemas.

El núcleo central del paquete es el editor de esquemas, llamado DRAFT, cuyas características más sobresalientes son las siguientes:

- Librería compuesta por más de 8,000 componentes.
- Colocación sencilla de conexiones, buses, conectores, etiquetas y uniones.
- Reconstrucción automática de conexiones y buses que han sido trasladados en un esquema.
- Rotación y reflejo de componentes.
- Operaciones con bloques.
- Autopanning.
- Cinco niveles de zoom.
- Hasta 100 macros definibles por el usuario.
- Limitados niveles de zoom.
- Acceso on-line a la biblioteca de componentes.
- Tamaño de texto definible por el usuario.
- Colocación de texto en vertical.
- 8 campos de definición por componente.
- Aplicación automática de los Teoremas de Nixson.

## Capítulo 1 Aplicación práctica de OrCAD SDT III



## Características específicas de OrCAD/SDT III

**OrCAD/SDT III** es un paquete de Diseño de Esquemas Electrónicos completo, flexible, que mediante un sistema de menús de fácil acceso, permite crear, editar, salvar, imprimir y plotear esquemas electrónicos, chequearlos, prepararlos para el trazado automático de pistas, crear componentes, numerarlos automáticamente, listarlos, etc.

Podemos considerarlo dividido en tres bloques fundamentales:

- Un editor de esquemas (**DRAFT**).
- Un editor de componentes (**LIBEDIT**).
- Un conjunto de utilidades que complementan al editor de esquemas.

El núcleo central del paquete es el editor de esquemas, llamado **DRAFT**, cuyas características más sobresalientes son las siguientes:

- Librería compuesta por más de 6.000 componentes.
- Colocación sencilla de conexiones, buses, conectores, etiquetas y uniones.
- Recomposición automática de conexiones y buses que han sido trasladados en un esquema.
- Rotación y reflejo de componentes.
- Operaciones con bloques.
- Autopanning.
- Cinco niveles de zoom.
- Hasta 100 macros definibles por el usuario.
- Ilimitados niveles de jerarquía.
- Acceso on-line a la biblioteca de componentes.
- Tamaño de texto definible por el usuario.
- Colocación de texto en vertical.
- 8 campos de definición por componente.
- Aplicación automática de los Teoremas de Morgan.

- Búsqueda automática de variables.
- Salida al DOS, tipo shell.
- Formatos de trabajo normalizados (DIN A4...A0).
- Shell con el simulador analógico PSpice.

En cuanto a las características del editor de librerías, podemos citar las siguientes:

- Personalización de librerías.
- Creación de nuevos símbolos standar.
- Posibilidad de convertir símbolos de componentes en datos utilizables por **DRAFT**.
- Utilización de un ensamblador (**COMPOSER**) y un desensamblador (**DECOMPOSER**) propios.
- Adaptación de esquemas realizados con antiguas versiones.

Además de estos dos bloques, esenciales en cualquier paquete de DEAO, el paquete se completa con una serie de utilidades que realizan una serie de funciones con el esquema ya terminado. Haremos una somera descripción de cada una de ellas, ya que serán ampliamente tratadas más adelante.

### **ANNOTATE**

Esta utilidad actualiza automáticamente cualquier componente dentro de un esquema, numerándolo de forma correlativa, si éste se ha añadido al esquema. También actualiza el número de pines asociados a dicho componente. Es muy útil en esquemas extensos y con gran densidad de componentes.

### **BACKANNO**

Actualiza las referencias correspondientes a componentes que han sido añadidos o modificados en un esquema tras la ejecución de Annotate.

### **CLEANUP**

Esta utilidad chequea las conexiones, buses, uniones, etiquetas y cualquier objeto, mostrando los mensajes oportunos cuando aparece algún objeto duplicado, borrándolo o solapándolo.

## CROSSREF

Mediante esta utilidad es posible muestrear cualquier esquema, recopilando información sobre los componentes utilizados, y crear un fichero de referencia que contiene la localización de cada componente.

## ERC

Realiza un chequeo de reglas eléctricas comunes, para advertir si existe algún error como por ejemplo, entradas sin señal, pines desconectados, etc.

## EXTRACT

Crea archivos fuente, utilizables por el módulo de PLDs, a partir de archivos creados con **SDT** (extensión .SCH).

## FLDATTRB

Modifica los atributos (referencia, valor y campos) de los componentes que integran un esquema, ya sea de bloques, jerárquico o simple, haciendo que estos sean visibles o invisibles.

## FLDSTUFF

Modifica los atributos (referencia y campos) de aquellos componentes que han sido definidos a través de un fichero de texto y que previamente han sido combinados mediante **KEY FIELD**.

## LIBARCH

Permite crear una librería fuente a partir de sólo los componentes utilizados en un esquema.

## LIBLIST

Genera un archivo en código ASCII a partir de un archivo de librería utilizable por **DRAFT**, conteniendo un listado de todos los componentes existentes en esa librería.

## NETLIST

Genera un fichero (NETLIST) preparado para ser capturado por un programa de colocación de componentes (PCB), en diferentes formatos, utilizados por programas de PCB más usuales.

## PARTLIST

Realiza un informe del esquema de trabajo, efectuando un listado de los componentes que intervienen en dicho esquema, así como sus principales características.

## PLOTALL Y PRINTALL

Sacan por plotter e impresora, respectivamente, los esquemas de todo tipo generados mediante **DRAFT**.

## SIMPLE

Convierte jerarquías complejas en jerarquías simples.

## TREELIST

Scanea los esquemas realizados con organización por jerarquía.

## XFEROVL

Transfiere información de configuración desde una versión a otra del archivo **ORCADSDT.OVL**. Ésto sólo es posible con versiones de **OrCAD/SDT III** superiores a la 3.12.

## Requisitos hardware

Aunque OrCAD/SDT III viene con tres discos de drivers, donde se encuentran prácticamente tanto las tarjetas gráficas, como los plotters e impresoras más variadas del mercado, se detalla a continuación una configuración con los requisitos mínimos para que el paquete funcione con un rendimiento adecuado:

- Ordenador IBM PC/XT o AT (80286 o 80386) o compatibles.
- Coprocesador matemático opcional
- 640 Kbytes de RAM como mínimo.
- Sistema Operativo MS-DOS 2.00 o superior.
- Disco duro de 20 Mbytes mínimo de capacidad.
- Tarjeta gráfica aconsejable: EGA color mínimo.
- Uno o dos puertos serie (para ratón y plotter).
- Un puerto paralelo.
- Ratón Microsoft, Logitech, Genius o compatible.
- Plotter HP o compatible (DIN A4... A0).

Normalmente, el tipo de ratones nombrados en esta configuración se acompaña con su propio driver (MOUSE.COM o MOUSE.SYS), que deberá añadirse oportunamente al fichero AUTOEXEC.BAT o bien al CONFIG.SYS, respectivamente.

No obstante, el programa funciona correctamente sin ratón, aunque la presencia de éste se agradece en esquemas de gran densidad y formato superior al DIN A-3.

Por último, el paquete se completa con tres discos de drivers, que hacen que este paquete sea totalmente versátil y adaptable a cualquier periférico disponible. Los diskettes 5 y 6, etiquetados DRIVER1 y DRIVER2, contienen los drivers de tarjetas gráficas, mientras que el 7 (DRIVER3) contiene, además de algunos drivers, también de tarjetas gráficas, los drivers correspondientes a impresoras y plotters.

Además de los diskettes, OrCAD/SDT III suministra una pastilla de protección hardware que se conecta a la puerta paralelo.

Se recomienda hacer copias de seguridad de cada uno de los siete diskettes.



## Contenidos del paquete

OrCAD/SDT III (Versión Educativa) se suministra con 7 diskettes de 5 1/4" DD. Los dos primeros, titulados **MASTER1** y **MASTER2**, contienen los archivos ejecutables del paquete, distribuidos así:

### **MASTER1**

ORCADSDT.OVL  
ANNOTATE.EXE  
BACKANNO.EXE  
CLEANUP.EXE  
DRAFT.EXE  
ERC.EXE  
FLDATTRB.EXE  
FLSTUFF.EXE  
LIBEDIT.EXE

### **MASTER2**

PSPICE.BAT  
CROSSREF.EXE  
EXTRACT.EXE  
LIBARCH.EXE  
NETLIST.EXE  
PARTLIST.EXE  
PLOTALL.EXE  
PRINTALL.EXE  
SIMPLE.EXE  
TREELIST.EXE  
XFEROVL.EXE  
MACRO1.MAC  
MACRO2.MAC  
PDRAFT.OVL

Los diskettes 3 y 4, titulados **LIBRARY1** y **LIBRARY2**, contienen las librerías de componentes, compactadas en los archivos **LIBRARY1.ARC** y **LIBRARY2.ARC**; el archivo descompactador **ARCE.COM**; el ensamblador propio **COMPOSER.EXE**; el desensamblador **DECOMP.EXE**; el fichero **LIBLIST.EXE** y dos ficheros de conversión, **FIX12X.EXE** y **FIX30X.EXE**, para adaptar ficheros de las versiones 1.2 y 3.0.

Por último, el paquete se completa con tres discos de drivers, que hacen que este paquete sea totalmente versátil y adaptable a cualquier periférico disponible. Los diskettes 5 y 6, etiquetados **DRIVER1** y **DRIVER2**, contienen los drivers de tarjetas gráficas, mientras que el 7 (**DRIVER3**) contiene, además de algunos drivers, también de tarjetas gráficas, los drivers correspondientes a impresoras y plotters.

Además de los diskettes, **OrCAD/SDT III** suministra una pastilla de protección hardware que se conecta a la puerta paralelo.

Se recomienda hacer copias de seguridad de cada uno de los siete diskettes.



# Instalación del paquete

Aunque en los requisitos hardware mencionamos la utilización de un equipo con disco duro, el paquete puede correr en equipos provistos con doble unidad de diskettes. Por este motivo, podemos considerar dos tipos de instalación: en diskettes y en disco duro.

## Instalación en diskettes

Podemos considerar, a su vez, tres casos (se suponen equipados con 640K de RAM y entorno MS-DOS):

- Equipo con unidades de disco de 5 1/4" de 360K (PC/XT compatibles).
- Equipo con unidades de disco de 3 1/2", 720K de capacidad (tipo PS/2)
- Equipo con unidades de disco de 3 1/2", 1.44M de capacidad (tipo AT).

A continuación ejemplificamos estos tres tipos de configuración con tres supuestos equipos.

**EQUIPO NUMERO 1.**- Ordenador compatible IBM equipado con 640 K de memoria RAM, doble unidad de diskette de 5 y cuarto, 360K, tarjeta gráfica CGA color, e impresora EPSON MX-80.

**Método de instalación.**- Necesitamos 5 diskettes DS DD, formateados, a los que etiquetaremos de la siguiente forma:

DISCO 1	-----	DRAFT
DISCO 2	-----	UTILIDADES DRAFT
DISCO 3	-----	EDITOR COMPONENTES
DISCO 4	-----	DRIVER/LIBRERIA
DISCO 5	-----	DISCO DE TRABAJO

A continuación copiaremos los ficheros que se indican con el comando COPY del MS-DOS o con cualquier utilidad (como PCTOOLS), desde los discos originales (con-

veniente, desde la copia de seguridad que hemos realizado previamente) a los diskettes recién etiquetados, siendo el contenido de cada diskette el siguiente:

### **ARCHIVOS DISCO 1**

DRAFT.EXE  
ORCADSDT.OVL  
ANNOTATE.EXE  
CLEANUP.EXE  
ERC.EXE  
PDRAFT.OVL  
PSPICE.BAT  
MACRO1.MAC  
MACRO2.MAC  
PARTLIST.EXE  
TREELIST.EXE

### **DONDE SE HALLAN**

MASTER1  
MASTER1  
MASTER1  
MASTER1  
MASTER1  
MASTER2  
MASTER2  
MASTER2 (OPCIONAL)  
MASTER2 (OPCIONAL)  
MASTER2  
MASTER2

### **ARCHIVOS DISCO 2**

ORCADSDT.OVL  
BACKANNO.EXE  
CROSREF.EXE  
NETLIST.EXE  
PLOTALL.EXE  
PRINTALL.EXE  
EXTRACT.EXE  
SIMPLE.EXE

### **DONDE SE HALLAN**

MASTER1  
MASTER1  
MASTER2  
MASTER2  
MASTER2  
MASTER2  
MASTER2  
MASTER2

### **ARCHIVOS DISCO 3**

ORCADSDT.OVL  
LIBARCH.EXE  
LIBEDIT.EXE  
LIBLIST.EXE  
COMPOSER.EXE  
DECOMP.EXE  
\*.LIB (descompactados)

### **DONDE SE HALLAN**

MASTER1  
MASTER2  
MASTER1  
LIB1  
LIB1  
LIB1  
LIB1 y LIB2

### **ARCHIVOS DISCO 4**

CGA2.DRV  
CGA4.DRV  
PRINTER.DRV

### **DONDE SE HALLAN**

DRIVER1  
DRIVER1  
DRIVER3

EPSONMX.DRV  
\*.LIB (descompactados)

DRIVER3  
LIB1 y LIB2

#### **ARCHIVOS Disco 4**

FIX12X.EXE  
FIX30X.EXE  
Ficheros de trabajo

#### **DONDE SE HALLAN**

LIB1  
LIB1  
—

Tanto el disco 3 como el 4 constan de una serie de ficheros que, originalmente, vienen comprimidos con objeto de ocupar el menor espacio posible y que han sido sometidos a un proceso llamado de **descompactación**, obteniéndose así archivos con la extensión .LIB que pueden ser interpretados desde **DRAFT**.

Estos ficheros se agrupan en dos extensos archivos denominados **LIBRARY1.ARC** y **LIBRARY2.ARC**. El proceso de descompactación se lleva a cabo con ayuda de un tercer fichero ejecutable llamado **ARCE.COM**, contenido en el disco etiquetado con LIB1. El método a seguir consta de los siguientes pasos:

1. Formatear dos diskettes y etiquetarlos como LIBRE1 y LIBRE2, por ejemplo.
2. Con el disco LIB1 en A: y LIBRE1 en B:, escribir:

ARCE LIBRARY1.ARC B:

y presionar ENTER. Aparecerá en pantalla lo siguiente:

**Archive: LIBRARY.ARC**

**UnCrunching > ANALOG.LIB**  
**UnCrunching > ASSEMBLY.LIB**  
**UnCrunching > BIT.LIB**  
**UnCrunching > CMOS.LIB**  
**UnCrunching > DEVICE.LIB**  
**UnCrunching > ECL.LIB**  
**UnCrunching > INTEL.LIB**  
**UnCrunching > LADDER.LIB**  
**UnCrunching > MEMORY.LIB**  
**UnCrunching > METER.LIB**  
**UnCrunching > MOTO.LIB**  
**UnCrunching > OPTOISOL.LIB**  
**UnCrunching > POWER.LIB**  
**UnCrunching > PSPICE.LIB**  
**UnCrunching > RF.LIB**

**UnCrunching > SHAPES.LIB**  
**UnCrunching > SPICE.LIB**  
**UnCrunching > TILSI.LIB**  
**UnCrunching > TTL.LIB**  
**UnCrunching > ZILOG.LIB**

Como vemos, aparece una serie de archivos con extensión **.LIB**; el nombre de cada archivo da una idea de los componentes que contiene.

3. Copiamos el archivo ARCE.COM al diskette LIB2, para lo cual, con el diskette LIB1 en A: y LIB2 en B:, escribimos:

COPY ARCE.COM B:

y pulsamos <ENTER>.

4. A continuación, introducimos LIB2 en A: y el diskette en blanco LIBRE2 en B: y tecleamos:

ARCE LIBRARY2.COM B:

y pulsamos <ENTER>.

Al igual que en el caso anterior, comenzará un proceso de descompactación mostrándonos en pantalla lo siguiente:

**Archive: LIBRARY2.ARC**

**UnCrunching > ALTERA\_M.LIB**  
**UnCrunching > ALTERA\_P.LIB**  
**UnCrunching > INTEL\_M.LIB**  
**UnCrunching > INTEL\_P.LIB**  
**UnCrunching > PLDGATES.LIB**  
**UnCrunching > VSTGATES.LIB**  
**UnCrunching > VSTOTHER.LIB**  
**UnCrunching > VSTRAM.LIB**

Los ficheros así obtenidos han de distribuirse entre los discos 3 y 4 (ficheros marcados con \*.LIB), procurando que los que se quieran cargar durante la sesión de trabajo se encuentren en el mismo diskette (recomendable el diskette 4, ya que contiene el resto de los drivers). De lo contrario, el programa emite un mensaje de error, anunciando que el fichero.LIB correspondiente no ha sido encontrado.

Por ejemplo, si deseamos realizar esquemas que lleven transistores, lógica TTL y CMOS, el contenido del diskette 4 quedaría de la siguiente forma:

PRINTER.DRV  
EPSONMX.DRV  
CGA2.DRV  
CGA4.DRV  
CMOS.LIB  
DEVICE.LIB  
TTL.LIB

Hay que tener en cuenta el espacio disponible en este diskette, ya que es posible que no podamos introducir las librerías que nos hacen falta.

**EQUIPO NÚMERO 2.**- Ordenador tipo PS/2, modelo 30 o compatible con doble unidad de diskette de 3 1/2", 720K de capacidad, 640K de RAM, tarjeta gráfica MCGA e impresora de 24 agujas EPSON LQ-1000.

**Método de instalación.**- Ante todo, es necesario solventar un primer problema: el cambio de formato. En efecto, el paquete se suministra en diskettes de 5 1/4", por lo que necesitaremos disponer de un ordenador con unidades de disco mixtas o un paquete de comunicaciones y dos ordenadores. Hoy día, este problema es de fácil solución con la configuraciones del mercado, y por tanto, supongamos que disponemos de un ordenador con dos unidades mixtas. Utilizaremos la A:, a la que suponemos de 5 1/4" para colocar nuestras copias de seguridad y la B: para adaptar el paquete a nuestras necesidades.

Necesitamos tres diskettes de 720K de capacidad, formateados y a los que etiquetaremos así:

DISCO 1	-----	DRAFT/UTILIDADES
DISCO 2	-----	LIBRERIAS/DRIVERS
DISCO 3	-----	DISCO DE TRABAJO

El contenido del disco 1, estará ahora formado por todos aquellos archivos que se encontraban en los diskettes 1 y 2 del caso anterior, mientras que en el disco 2, podemos agrupar todos los archivos que incluíamos en los discos 3 y 4 del caso anterior.

Para facilitar nuestro trabajo, podemos organizar este diskette número 2 en algunos directorios. El proceso a seguir sería el siguiente:

1. En primer lugar (suponemos formateado el diskette), vamos a crear dos directorios llamados **LIBRERIA** y **DRIVERS**. Para ello, haciendo uso del comando interno de MS-DOS **MD**, escribimos:

MD LIBRERIA <ENTER>  
MD DRIVERS <ENTER>

A continuación, utilizando las copias de seguridad del paquete, copiar mediante la orden COPY de MS-DOS o con cualquier otra utilidad, los siguientes archivos, al directorio raíz del diskette 2 que hemos etiquetado con el nombre de LIBRERIAS/DRIVERS y que suponemos situado en la unidad B:

<b>ARCHIVO</b>	<b>DONDE SE HALLAN</b>
ORCADSDT.OVL	MASTER1
LIBARCH.EXE	MASTER2
LIBEDIT.EXE	MASTER1
LIBLIST.EXE	LIB1

Seguidamente, mediante las órdenes B: <ENTER> y CD LIBRERIA <ENTER>, cambiamos de directorio y copiamos los siguientes archivos:

<b>ARCHIVO</b>	<b>DONDE SE HALLAN</b>
COMPOSER.EXE	LIB1
DECOMP.EXE	LIB1
*.LIB	LIB1 y LIB2

Los archivos comprendidos en la notación \*.LIB, son las librerías previamente descompactadas mediante el archivo ARCE.COM, cuyo proceso hemos visto anteriormente.

Una vez copiados todos estos archivos, cambiamos el directorio de trabajo a DRIVERS, mediante la ejecución de las órdenes CD.. <ENTER> y CD DRIVERS <ENTER> y copiamos los siguientes archivos:

<b>ARCHIVO</b>	<b>DONDE SE HALLAN</b>
CGA2.DRV	DRIVER1
CGA4.DRV	DRIVER1
PRINTER.DRV	DRIVER3
LQ1000.DRV	DRIVER3

A la hora de configurar **DRAFT**, habrá que especificar la ruta de búsqueda de los archivos de LIBRERIA y DRIVERS como más adelante se indica (ver el apartado titulado CONFIGURACIÓN DE OrCAD/SDT III).

Por último, en nuestro disco número 3, etiquetado como DISCO DE TRABAJO, copiaremos los archivos de conversión FIX12X.EXE y FIX30X.EXE presentes en el diskette LIB1.

**EQUIPO NÚMERO 3.**- Ordenador AT con doble unidad de disco de alta densidad, de 3 1/2" y 5 1/4" pulgadas, 640K de RAM (o 1024K), tarjeta gráfica VGA color e impresora HP LaserJet Serie II.

**Método de instalación.**- Con este equipo, podemos utilizar diskettes de 720K/1.44M ó 360K/1.2M de capacidad, indistintamente.

En el primer caso, podemos remitirnos al supuesto anterior con la salvedad de los drivers correspondientes a la tarjeta gráfica y la impresora, los cuales habrá que cambiar.

Resulta mucho más interesante la utilización de diskettes de alta densidad, en cuyo caso sólo necesitaremos uno de 3 1/2", mientras que para nuestro disco de trabajo, será suficiente con uno de 5 1/4", baja densidad. Tras un previo formateo, los etiquetaremos como DISCO DE PROGRAMA y DISCO DE TRABAJO respectivamente. En este último, al igual que en casos anteriores, habrá que copiar los archivos de conversión.

En cuanto al diskette de PROGRAMA crearemos dos directorios a los que llamaremos LIBRERIA y DRIVERS.

En el primero de estos directorios copiaremos sólo los archivos de librería (\*.LIB), previamente descompactados (o tan solo los que creamos necesarios), utilizando el mismo proceso seguido en el caso anterior.

En el directorio llamado DRIVERS, copiaremos los siguientes archivos: EGA16E.DRV, VGA640.DRV como drivers correspondientes a tarjetas gráficas y HPLASER1.DRV ... HPLASER6.DRV, como drivers para impresora, dependiendo de la resolución que deseemos. También hay que copiar PRINTER.DRV.

En el directorio raíz deberán copiarse el resto de los archivos distribuidos en los 3 primeros diskettes originales.

## Instalación en disco duro

Es la mejor forma de trabajar con el programa, ya que nos evitamos el trasiego de diskettes, siendo además la carga del paquete mucho más rápida, sin contar las ventajas de organización de que dispone un disco duro.

**Método de instalación.** - Supongamos, que la unidad de disco duro de que disponemos es la C: y la unidad de disco flexible es A:. Sigamos los siguientes pasos:

### **PASO 1:**

Situados en el directorio raíz C>, creamos un nuevo directorio llamado OrCAD. Para ello UTILIZAMOS el comando interno MD del MS-DOS y escribimos:

```
MD ORCAD <ENTER>
```

A continuación, nos situamos en este directorio con la orden

```
CD ORCAD <ENTER>
```

### **PASO 2:**

Vamos a crear ahora tres subdirectorios llamados **LIBRERIA**, **DRIVER** y **PLANOS**. Para ello escribiremos:

```
MD LIBRERIA <ENTER>
```

```
MD DRIVER <ENTER>
```

```
MD PLANOS <ENTER>
```

### **PASO 3:**

Coloquemos en la unidad A el diskette 1 MASTER1 y situados en el directorio OrCAD, escribamos:

```
COPY A:*. * C: <ENTER>
```

Cuando se hayan copiado todos los ficheros, extraigamos el diskette de la unidad.

### **PASO 4:**

Coloquemos el diskette 2 MASTER2 y procedamos de forma similar:

```
COPY A:*. * C: <ENTER>
```

Cuando se hayan copiado todos los ficheros, extraigamos el diskette de la unidad.

### **PASO 5:**

Coloquemos el disco 3, LIBRARY DISK1, en la unidad A y copiemos todos sus ficheros al subdirectorio llamado **LIBRERIA**, con la orden:

COPY A:\*. \* C:\OrCAD\LIBRERIA

Coloquémonos en este subdirectorio con la orden:

CD LIBRERIA <ENTER>

y copiemos el archivo LIBEDIT.EXE al directorio OrCAD

COPY LIBEDIT.EXE C:\OrCAD <ENTER>

borrándolo a continuación con:

DEL LIBEDIT.EXE <ENTER>

Extraigamos el disco de la unidad A:

### **Paso 6:**

Coloquemos el diskette 4, LIBRARY 2, en la unidad A y copiemos los dos archivos de que consta:

COPY A:\*. \* C:

Copiemos los archivos de conversión FIX12X.EXE y FIX30X.EXE al subdirectorio PLANOS mediante la orden:

COPY FIX\*. \* C:\OrCAD\PLANOS

Eliminamos dichos archivos con la orden:

DEL FIX\*. \*

Extraigamos el diskette de la unidad A.

### **Paso 7:**

Procedamos a la descompactación de archivos de librería, mediante las órdenes:

ARCE LIBRARY1.ARC <ENTER>

ARCE LIBRARY2.ARC <ENTER>

Eliminemos el fichero ARCE.COM mediante la orden:

DEL ARCE.COM <ENTER>

Situémonos en el directorio OrCAD con la orden:

CD OrCAD <ENTER>

### PASO 8:

Situándonos en el subdirectorio DRIVER mediante la orden:

```
CD DRIVER <ENTER>
```

Coloquemos ahora el diskette 5, DRIVER DISK1, en la unidad A y copiemos todos sus archivos al subdirectorio DRIVER, con la orden:

```
COPY A:*. * C: <ENTER>
```

Extraigamos el diskette de la unidad A.

### PASO 9:

Coloquemos el diskette 6, DRIVER DISK2, en la unidad A y copiemos todos los archivos al mismo subdirectorio con la orden:

```
COPY A:*. * C: <ENTER>
```

Una vez copiados, extraigamos el diskette de la unidad A.

### PASO 10:

Por último coloquemos el diskette 7, DRIVER DISK3, en la unidad A y copiemos todos los archivos de que consta, mediante la orden:

```
COPY A:*. * C: <ENTER>
```

Extraigamos el diskette de la unidad y volvamos al directorio OrCAD con la orden:

```
CD ORCAD <ENTER>
```

con lo que queda finalizada la instalación.

### NOTAS:

Si se va a utilizar el programa de simulación analógica **PSpice** conjuntamente con **DRAFT**, es necesario cargar todos los archivos y utilidades de dicho programa en el directorio OrCAD, o en su caso, en aquél donde se encuentre el archivo **DRAFT.EXE**.

Con objeto de economizar espacio en el disco duro, no es necesario copiar el contenido de los tres diskettes de drivers, sino tan solo aquellos archivos de drivers que se adapten a nuestro hardware. Estos archivos serán como mínimo los siguientes:

- Un driver para tarjeta gráfica.
- Un driver para impresora, plotter o ambos a la vez.
- El archivo PRINTER.DRV, necesario siempre que se utilice cualquier impresora.

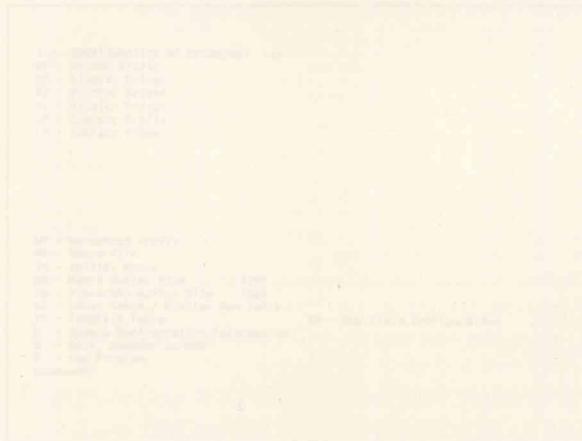
En las configuraciones para discos flexibles, expuestas anteriormente, hay varios ejemplos de utilización de estos drivers.

Si se va a trabajar de forma continuada con **OrCAD/SDT III**, es conveniente que el programa arranque de la forma más automatizada posible, lo cual podemos conseguir de dos formas:

1. Colocando la ruta de búsqueda mediante un **PATH**, en el archivo **AUTOEXEC.BAT** (PATH C:\OrCAD;).
2. Creando un pequeño fichero **.BAT** (por ejemplo **D.BAT**) en el directorio raíz (C:\) de la siguiente forma:

```
C:\>COPY CON D.BAT <ENTER>
CD OCAD <ENTER>
DRAFT <ENTER>
CD.. <ENTER>
<F6> <ENTER>
```

**Nota. No intentar colocar Draft mediante el comando Path del S.O, ya que se producirá un error al intentar cargarlo debido a la existencia de un fichero .OVL.**





# Configuración de OrCAD/SDT III

Una vez se ha configurado el paquete para nuestras necesidades hardware, es necesario que el programa reconozca esa configuración y pueda acceder a las rutas de búsqueda especificadas o a los diskettes, según sea el tipo de equipo para el cual hemos efectuado la configuración.

Como ya dijimos al comentar sus características, el archivo **DRAFT.EXE**, es el núcleo central del paquete y si queremos comenzar a trabajar en la realización de cualquier esquema, lo primero que hay que hacer es adaptar este fichero a la configuración que previamente hemos realizado.

Para ello, este archivo dispone de un **switch** mediante el cual se puede cambiar la configuración de **OrCAD/SDT III** en cualquier momento.

**Cuando arranquemos el programa por vez primera, es obligatorio hacer correr la opción de configuración, ya que de lo contrario, obtendremos un mensaje de error.**

Situándonos, pues, en el directorio ORCAD, tecleemos lo siguiente:

**DRAFT /C** <ENTER>

En pantalla aparecerá el menú de configuración de la figura 1.1.

```

::: CONFIGURATION OF OrCAD/SDT :::
DP - Driver Prefix
DD - Display Driver
PD - Printer Driver
PL - Plotter Driver
LP - Library Prefix
LF - Library Files

WP - Worksheet Prefix
MF - Macro File
IM - Initial Macro
MB - Macro Buffer Size      8192
HB - Hierarchy Buffer Size  1024
CT - Color Table / Plotter Pen Table
TT - Template Table
U - Update Configuration Information
Q - Quit, Abandon to DOS
R - Run Program
Command?

KF - Key Field Configuration
```

Figura 1.1

## Prefijo de dispositivo (driver prefix)

El prefijo de driver (dispositivo) sirve para indicar a **OrCAD/SDT III** la ruta o unidad de disco donde puede cargar los drivers para la tarjeta gráfica, impresora y plotter que vamos a utilizar.

Para seleccionar este **prefijo**, tecleamos **DP**, con lo que nos aparecerá una **barra en video inverso** en la que debemos introducir el nombre del directorio o unidad donde se encuentran los **drivers**. Si usamos diskettes, deberemos poner obligatoriamente **B:** (seguido del directorio si lo hubiera), mientras que si trabajamos con disco duro podemos poner **DRIVER\** y validar con <ENTER>.

**No olvidar colocar el signo “\” cuando especifiquemos el directorio de búsqueda, pues de lo contrario obtendríamos un mensaje de error.**

## Dispositivo de pantalla (Display Driver)

Este prefijo configura **OrCAD/SDT III** para la tarjeta gráfica de nuestro equipo. En esta ocasión, tecleando **DD** se nos muestran hasta tres pantallas de drivers (en la figura 1.2 se muestra la primera de ellas) correspondientes a las tarjetas gráficas más utilizadas en la actualidad, numeradas del 1 al 41, según un rango que va desde 320x200 (CGA) hasta 1.248x1.024. Basta teclear el número correspondiente al driver que queremos configurar y a continuación, con **Q**, volvemos al menú de configuración.

```

:::Display Driver Configuration:::          Current Selection =
Size Colors Adapter
1 - 320 x 200 4 Color Graphics Adapter      CGA4.DRV
2 - 640 x 200 2 Color Graphics Adapter      CGA2.DRV
3 - 640 x 200 16 EGA standard monitor       EGA16C2.DRV
4 - 640 x 200 16 Tecmar Graphics Master    TGM16N.DRV
5 - 640 x 350 1 EGA Monochrome monitor     EGA2.DRV
6 - 640 x 350 4 EGA (64K RAM)              EGA4E.DRV
7 - 640 x 350 16 EGA Enhanced monitor      EGA16E.DRV
8 - 640 x 400 2 AT&T 6300                  ATT2C.DRV
9 - 640 x 400 2 HP Vectra Multi-Video      HPVECTRA.DRV
10 - 640 x 400 2 NCR graphics adapter       NCR2.DRV
11 - 640 x 400 2 Tandy 2000 Graphics        TANDY2k.DRV
12 - 640 x 400 2 Televideo TeleCAT         TELECAT.DRV
13 - 640 x 400 2 Toshiba 3100 Laptop       T3100.DRV
14 - 640 x 400 2 Wyse-700                  WYSE700a.DRV
15 - 640 x 400 4 NCR graphics adapter       NCR4.DRV
16 - 640 x 400 16 AT&T 6300 DEB            ATT16C.DRV

<<<NOTE>>> More drivers are available using the GENDRIVE utility

M - More
Q - Quit
S - Special (non-listed above)
Selection ->
```

Figura 1.2

## Dispositivo de impresora (Printer Driver)

Para seleccionar una impresora, basta teclear DP desde el menú de configuración. Automáticamente se nos mostrará una pantalla donde existen hasta 16 tipos de impresoras, en orden alfabético, y, en la parte inferior, aparece un pequeño menú de selección con las siguientes opciones:

**M - More.** Permite visualizar hasta dos pantallas más de drivers de impresora.

**X - None of the Above.** Permite dejar en blanco la selección de dispositivo de impresora.

**Q - Quit.** Permite retroceder al menú de configuración.

**S - Special.** Permite seleccionar un driver especial (no listado) para lo cual hay que teclear el nombre de dicho dispositivo.

La figura 1.3 se muestra la primera de las 3 pantallas de drivers de impresoras de que el paquete dispone.

```

:::Printer Driver Configuration:::
Printer Model                               Current Selection =
                                           DPI
1 - AMT Office Printer (120x120) AMT.DRV
2 - AMT ACCEL-500 (120x120) AMT_ACCEL.DRV
3 - C. ITOH 1550/8510 (136x144) CITR1550.DRV
4 - C. ITOH P310 (136x144) CITOH310.DRV
5 - DataProducts 8012 (165x165) IBMCLDR.DRV
6 - Epson ( 60x 72) EPSON.DRV
7 - Epson MX (120x216) EPSONMX.DRV
8 - Epson MX/FX/RX (120x108) EPSONFX2.DRV
9 - Epson FX/RX (240x216) EPSONFX.DRV
10 - Epson LQ800/LQ1000/LQ1500 ( 90x 90) LQ1000_2.DRV
11 - Epson LQ800/LQ1000/LQ1500 (180x180) LQ1000.DRV
12 - Epson VP 7690 (180x180) EPSONVP.DRV
13 - HP DeskJet (Letter Paper)( 75x 75) HPDESK1.DRV
14 - HP DeskJet (Letter Paper)(100x100) HPDESK2.DRV
15 - HP DeskJet (Letter Paper)(150x150) HPDESK3.DRV
16 - HP DeskJet (Letter Paper)(300x300) HPDESK4.DRV

M - More
X - none of the Above
Q - Quit
S - Special (non-listed above)
Selection ->
```

Figura 1.3

## Dispositivo de plotter (Plotter Driver)

Para seleccionar cualquiera de los plotters de que dispone el programa, basta pulsar **PD**, desde el menú de configuración, y a continuación se nos mostrará una pantalla como la que muestra la figura 1.4, donde se observa cómo además de seleccionar

el plotter deseado, es necesario configurar los parámetros de la puerta serie a la que consideramos conectado el plotter seleccionado (canal, baudios, paridad, etc.). Para ello basta pulsar la tecla correspondiente al carácter ":" y configuremos dichas opciones. Disponemos de un pequeño menú de selección con las opciones **X**, **S** y **Q** que realizan la misma función que en el caso de selección de impresora.

```

:::Plotter Driver Configuration:::      Current Selection =
  Plotter Model
  1 - Apple 410                          APPLE410.DRV
  2 - Calcomp (Intelligent)              CALCOMP1.DRV
  3 - Calcomp (Non-Intelligent)          CALCOMP2.DRV
  4 - DXF (interface to AutoCad, Generic CAD, etc.)  DXF.DRV
  5 - Fujitsu FPG 315                    HP2.DRV
  6 - Houston Instrument DMP 40/41/42/51/52/56/etc. (DM/PL)  HI.DRV
  7 - Houston Instrument DMP 29 (DM/PL Etx is _)  HI29.DRV
  8 - HP 7475/7550/7580/7585/7586/etc (HP-GL) (.00098"/unit)  HP.DRV

: - Serial interface  Channel 1  9600 Baud  NO Parity  8 Bits

M - More
X - none of the Above
Q - Quit
S - Special (non-listed above)
Selection ->

```

Figura 1.4

## Prefijo de librería (Library Prefix)

Permite a **OrCAD/SDT III** encontrar el subdirectorio o la unidad donde se encuentran los archivos de librería.

Desde el menú de configuración, basta pulsar **LP** y se nos mostrará una barra en video inverso donde introduciremos el nombre del subdirectorio donde se encuentran nuestros archivos de librería, como por ejemplo **LIBRERIA\** (o **LIBSDT\**) en caso de disco duro o bien **B:LIBRERIA** en caso de unidad B:

## Archivos de librería (Library Files)

Para configurar las librerías que vamos a necesitar a la hora de extraer componentes para nuestros esquemas, basta pulsar **LF** desde el menú de configuración con lo que se nos mostrará un menú (ver figura 1.5) con las opciones de **AÑADIR (A)** una librería; **ELIMINAR (R)** una librería o **VOLVER (Q)** al menú de configuración.

✓ Cuando pulsamos **A**, aparece el mensaje **"Add File Name before? [1..2]"** a lo que hay que responder con el número de orden en que queremos añadir la librería.

**No conviene configurar todas las librerías de que consta el paquete (28 en total), ya que puede ocurrir que nos quedemos sin memoria suficiente, además de ralentizar la carga del programa.**

**El listado de las 28 librerías disponibles se mostró anteriormente, cuando se explicó el método de descompactación de archivos de librería.**

Además de las librerías que se suministran con el paquete, podemos añadir cualquier otra creada por nosotros mismos (por ejemplo, una librería que contenga componentes de reciente aparición en el mercado) o bien librerías **personalizadas**, es decir, hechas a nuestra medida, lo cual veremos en el capítulo cuatro.

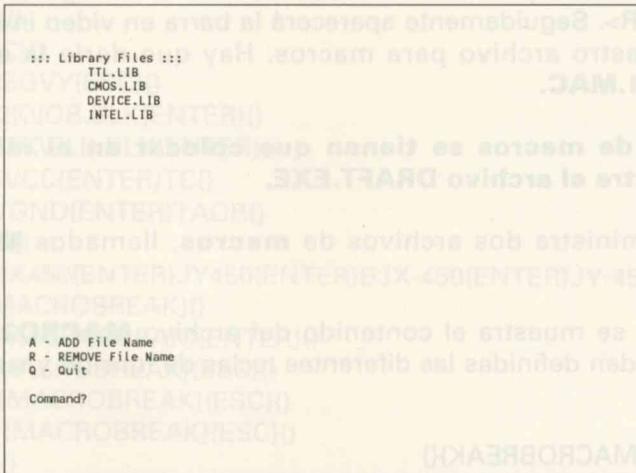


Figura 1.5

## Prefijo de hoja de trabajo (Worksheet Prefix)

Permite a **OrCAD/SDT III** encontrar el subdirectorio o unidad donde queremos almacenar nuestros archivos de trabajo.

Para dar nombre a este subdirectorio o unidad, basta pulsar **WP** desde el menú de configuración, apareciendo la consabida barra en vídeo inverso que nos invita a introducir el nombre que deseemos, como por ejemplo **PLANOS\** o **B:PLANOS**.

En este subdirectorio, es conveniente tener copiados los archivos **FIX12X.EXE** y **FIX30X.EXE**, que nos actualizan archivos de trabajo procedentes de otras versiones.

Los archivos de trabajo salvados desde **DRAFT** se graban sin extensión alguna, por lo que es conveniente añadirles alguna, con objeto de diferenciarlos de otros creados con otras utilidades del paquete. Así, por ejemplo, se les puede colocar la extensión **.SCH** (de esquema).

## Archivo de macros (Macro File)

**OrCAD/SDT III**, permite la creación y utilización de **macros** que se van a almacenar en uno o dos archivos. Estos archivos deben abrirse cuando se carga el programa, lo cual queda especificado si desde el menú de configuración tecleamos **MF** seguido de <ENTER>. Seguidamente aparecerá la barra en vídeo inverso pidiéndonos el nombre de nuestro archivo para macros. Hay que darle la extensión **.MAC**. Ejemplo: **MACRO1.MAC**.

**Los archivos de macros se tienen que colocar en el mismo directorio donde se encuentre el archivo DRAFT.EXE.**

El paquete suministra dos archivos de **macros**, llamados **MACRO1.MAC** y **MACRO2.MAC**.

A continuación se muestra el contenido del archivo **MACRO2.MAC**, donde se observa cómo quedan definidas las diferentes teclas de función y sus diferentes combinaciones:

```
{F1}=G{ENTER}{MACROBREAK}{}  
{F2}=PWB{  
{F3}=BSBEBG{  
{F4}=DBBE{  
{F5}=BDBE{  
{F6}=PBB{  
{F7}=PL{  
{F8}=PM{MACROBREAK}{  
{F9}=PS{MACROBREAK}{ENTER}B{MACROBREAK}E{  
{F10}=ZS{D}{D}{D}{ENTER}ZS{ENTER}{ESC}{  
{^F1}=LD{MACROBREAK}{ENTER}{  
{^F2}=PJP{ESC}{}
```

```

{\^F3}=BS{MACROBREAK}{ENTER}{MACROBREAK}{MACROBREAK}BG{}
{\^F4}=DB{MACROBREAK}{ENTER}{MACROBREAK}{}
{\^F5}=BM{MACROBREAK}{ENTER}{MACROBREAK}{}
{\^F6}=EERN{MACROBREAK}{ESC}{ESC}{}
{\^F7}=JX450{ENTER}JY450{ENTER}EE{MACROBREAK}{}
{\^F8}=ZS{D}{D}{D}{ENTER}{ESC}SW{MACROBREAK}ZS{ENTER}{ESC}{}
{\^F9}=PT{MACROBREAK}{ENTER}{ESC}{}
{\^F10}=C{}
{F1}=SAN{}
{F2}=SAY{}
{F3}=ZS{D}{D}{D}{ENTER}{MACROBREAK}P{ENTER}ZS1{ESC}{}
{F4}=DU{}
{F5}=ZI{ESC}{}
{F6}=ZO{ESC}{}
{F7}=QUEE{ESC}{}
{F8}=QUL{ESC}{}
{F9}=SGVN{ESC}{}
{F10}={ENTER}SGVY{ESC}{}
{I}=BI..\LCALIB2K\IOB.BLK{ENTER}{}
{C}=BI..\LCALIB2K\CLB.BLK{ENTER}{}
{P}=PPV{ESC}VVCC{ENTER}TC{}
{G}=PPV{ESC}VGND{ENTER}TAOB{}
{SHIFT-F1}=QAY{}
{SHIFT-F2}=DBJX450{ENTER}JY450{ENTER}BJX-450{ENTER}JY-450{ENTER}E{ESC}{}
{SHIFT-F3}=BI{MACROBREAK}{}
{SHIFT-F4}=BE{MACROBREAK}{ENTER}{}
{SHIFT-F5}=QI{MACROBREAK}{ESC}{}
{SHIFT-F6}=QU{MACROBREAK}{ESC}{}
{SHIFT-F7}=QW{MACROBREAK}{ESC}{}
{SHIFT-F8}=HM{}
{SHIFT-F9}=QS{}
{SHIFT-F10}=QUA{}
{MMB}=PWB{}

```

## Macro inicial (Initial Macro)

Permite ejecutar una macro específica de forma automática, una vez se ha invocado **DRAFT**. Para trabajar con una **Macro Inicial** hay que cargar previamente el archivo que la contenga.

Como ejemplo, se puede realizar una macro inicial de forma que, al arrancar DRAFT, aparezca la pantalla de trabajo en blanco y las coordenadas en el margen superior derecho. Como más adelante veremos, la composición de dicha **macro** es:

```
{F10}={ENTER}SXY{}
```

que podemos almacenar en un archivo llamado **MACRO.MAC**.

Para seleccionar una **Macro Inicial** se tecléa **IM** desde el menú de configuración y cuando aparece la barra en vídeo inverso teclearemos cada uno de los caracteres que forman el nombre de la macro que queramos correr, seguidos de <ENTER>, finalizando la introducción con <ENTER> nuevamente.

En nuestro ejemplo, la secuencia de caracteres a introducir en la barra iluminada sería: **F<ENTER>**, **1<ENTER>**, **0<ENTER>**, <ENTER>.

## Tamaño del buffer de macros

Esta opción del menú de configuración permite modificar la cantidad de memoria que ocupará el **buffer** destinado al almacenamiento y localización de las **macros**.

El mínimo tamaño de dicho **buffer** es de **16.384** bytes y el máximo de **65.535** bytes.

Para modificar este tamaño, se tecléa **MB** desde el menú de configuración, apareciendo el mensaje:

**"Macro Buffer Size?"**

Se tecléa el tamaño requerido, siempre comprendido en el margen antes citado y se pulsa <ENTER>.

## Tamaño del buffer de jerarquía

De forma similar a la anterior, esta opción permite modificar la cantidad de memoria que el programa asigna al **buffer** destinado al almacenamiento de todos los **planos** de jerarquía (en la siguiente sección se aclara este concepto), así como las **sendas**.

El tamaño mínimo de este es de **1.024** bytes que nos permite crear entre 75 y 100 **planos de jerarquía**, mientras que el tamaño máximo es de **65.535** bytes, que nos permite crear unos 200 **planos** complejos y densos.

Para modificar el tamaño de este **buffer**, basta con teclear **HB** desde el menú de configuración, y en respuesta al mensaje:

### "Hierarchy Buffer Size?"

teclea el tamaño deseado, siempre en el margen 1.024-65.535 y validar con <ENTER>.

## Paleta de colores (color table / plotter table)

Esta opción permite modificar los colores que **OrCAD/SDT III** asigna tanto a la pantalla como a los estilógrafos del plotter, referidos a:

- La numeración de pines del componente.
- El nombre de cada pin.
- Las conexiones del esquema.
- Los buses, uniones y conectores del esquema.
- Cualquier otro objeto procedente de una librería.

Para activar esta opción, basta teclear **TC** desde el menú de configuración, apareciendo una pantalla (ver figura 6) en la que podemos observar 7 columnas en la parte superior y un menú de 7 opciones en la parte inferior.

- La primera columna está formada por una serie de números (**items**) del 1-16 en una primera pantalla y del 15-30 (existe un solapamiento) en la siguiente, a la que se accede con el comando **M** (More) del menú inferior.
- La segunda columna está formada por una serie de denominaciones que denominaremos **objetos de librería** (cuerpo de los C.I., conexiones, buses, conectores, etc., hasta 30 distribuidos en dos pantallas) a los que le corresponde un **item** de la columna anterior.

Estas denominaciones son invariables en cuanto a concepto, pero pueden modificarse sus características, como por ejemplo el color de presentación en pantalla.

- En la tercera columna, se presentan una serie de colores seleccionados previamente a partir de una paleta que depende del tipo de tarjeta gráfica seleccionada en la configuración. En la figura 1.6 se muestra (columna 7) la paleta correspondiente a una tarjeta **EGA/VGA COLOR**.

```

::: Color / Plotter Configuration ::: Pen Width Velocity
1 - Part Body RED 1 .010 DEFAULT 0 - BLACK
2 - Pin Number GREEN 1 .010 DEFAULT 1 - BLUE
3 - Pin Name BROWN 1 .010 DEFAULT 2 - GREEN
4 - Part Reference LIGHT GRAY 1 .010 DEFAULT 3 - CYAN
5 - Part Value LIGHT GRAY 1 .010 DEFAULT 4 - RED
6 - 1st Part Field LIGHT GRAY 1 .010 DEFAULT 5 - MAGENTA
7 - 2nd Part Field LIGHT GRAY 1 .010 DEFAULT 6 - BROWN
8 - 3rd Part Field LIGHT GRAY 1 .010 DEFAULT 7 - DARK GRAY
9 - 4th Part Field LIGHT GRAY 1 .010 DEFAULT 8 - LIGHT GRAY
10 - 5th Part Field LIGHT GRAY 1 .010 DEFAULT 9 - LIGHT BLUE
11 - 6th Part Field LIGHT GRAY 1 .010 DEFAULT 10 - LIGHT CYAN
12 - 7th Part Field LIGHT GRAY 1 .010 DEFAULT 11 - LIGHT MAGENTA
13 - 8th Part Field LIGHT GRAY 1 .010 DEFAULT 12 - LIGHT RED
14 - Wire BROWN 1 .010 DEFAULT 13 - LIGHT MAGENTA
15 - Bus BLUE 1 .010 DEFAULT 14 - YELLOW
16 - Junction GREEN 1 .010 DEFAULT 15 - WHITE
C - Color select
P - Pen select
V - Velocity select
W - Pen Width select
F - Field Name Edit
M - More
Q - Quit
Command->

```

Figura 1.6

Estos colores pueden modificarse con la opción **C** del menú inferior. Cuando se activa, el programa nos pide el **Item** que deseamos modificar. Una vez introducido el número, nos pide el nuevo color (**New color**) a lo que contestaremos con otro número de la columna 7 correspondiente a la paleta de colores de la tarjeta gráfica seleccionada.

- La cuarta columna, sirve para seleccionar el tipo de **lápiz de plotter**. Al pulsar **P**, el programa nos pide el **número de lápiz**, seleccionable entre 1 y 16, avisándonos de que si entramos un **0** el plotter podrá detenerse y permitirá cambiar el lápiz. Si entramos **99**, el plotter puede suprimir denominaciones.
- La quinta columna sirve para modificar el **ancho** de escritura del lápiz del plotter. Para ello basta pulsar **W**, introducir el **item** deseado, validar con <ENTER> y ante la pregunta **"Pen Widht"**, introducir el nuevo tamaño (en pulgadas) y validar con <ENTER>.
- La sexta columna sirve para modificar la **velocidad** del lápiz del plotter. Es útil cuando queremos, por ejemplo, que los **buses** se impriman con mayor rapidez.

Para modificar la velocidad, basta teclear la opción **V** del menú inferior, seleccionar el **item** a cambiar y tras presionar <ENTER>, introducir la nueva velocidad deseada, cuidando de que ésta se adecúe al plotter de que disponemos. La opción **DEFAULT**, nos selecciona las velocidades por defecto. No olvidarse de presionar <ENTER> para validar.

Existe una opción en el menú inferior, que permite modificar el nombre de los 8 campos de definición de un componente.

Para ello tecleamos **F**, seleccionamos el **item** a modificar y tras pulsar <ENTER>, introducimos el nuevo nombre del campo. Las modificaciones efectuadas de esta forma deben quedar reflejadas en el comando **Edit** del menú de **DRAFT**.

## Tabla de tamaños (Template Table)

Permite cambiar el tamaño de varios parámetros, dentro de **DRAFT**. Entre estos parámetros se encuentran: dimensiones del formato, tamaño del texto, tamaño de los objetos, etc. La unidad de medida utilizada es la pulgada.

Esta opción se activa tecleando **TT**, desde el menú de configuración. Nos aparece una pantalla como la de la figura 1.7.

```

::: Template Configuration :::
1 - Horizontal      9.700  15.200  20.200  32.200  42.200
2 - Vertical       7.200   9.700  15.200  20.200  32.200
3 - Pin to Pin     0.100   0.100   0.100   0.100   0.100
4 - Pin Number     0.060   0.060   0.060   0.060   0.060
5 - Pin Name       0.060   0.060   0.060   0.060   0.060
6 - Part Reference 0.060   0.060   0.060   0.060   0.060
7 - Part Value     0.060   0.060   0.060   0.060   0.060
8 - 1st Part Field 0.060   0.060   0.060   0.060   0.060
9 - 2nd Part Field 0.060   0.060   0.060   0.060   0.060
10 - 3rd Part Field 0.060   0.060   0.060   0.060   0.060
11 - 4th Part Field 0.060   0.060   0.060   0.060   0.060
12 - 5th Part Field 0.060   0.060   0.060   0.060   0.060
13 - 6th Part Field 0.060   0.060   0.060   0.060   0.060
14 - 7th Part Field 0.060   0.060   0.060   0.060   0.060
15 - 8th Part Field 0.060   0.060   0.060   0.060   0.060
16 - Power Text   0.060   0.060   0.060   0.060   0.060

- Sheet
M - More
Q - Quit
Command->
  
```

Figura 1.7

En ella se observan 5 columnas (A..E), que se corresponden con los 5 formatos según norma DIN (A4..A0) y una serie variables (28 en total, distribuidas en dos pantallas) a cada una de las cuales corresponde un **item**.

Para modificar cualquier parámetro, se procede de la siguiente forma: supongamos que deseamos modificar la distancia entre **pines** de un **C.I.** para formatos A4. La secuencia de operación es la siguiente: desde el menú de configuración, tecleamos **TT**

y a continuación **A**, con lo que seleccionamos la columna correspondiente al formato A4. A continuación tecleamos el **item 3 (Pin to Pin)** y, a continuación, el valor que deseamos para la distancia entre **pinnes** (por ejemplo **0.050**).

Con la opción **M** del pequeño menú inferior, accedemos a una segunda pantalla, mientras, con **Q**, salimos al menú de configuración.

## Campo llave (Key Field)

Permite combinar el valor de un componente (**Part Value**) con una serie de atributos (**Part Field**), tales como la tolerancia o cualquier otra especificación que creamos oportuna incluir, cuando se ejecutan algunas utilidades de **SDT** como **ANNOTATE**, **EXTRACT**, **FLDSTUFF**, **NETLIST** y **PARTLIST**.

Para seleccionar esta opción de configuración, basta teclear **KF** desde el menú de configuración, con lo que se mostrará una pantalla como la de la figura 1.8.

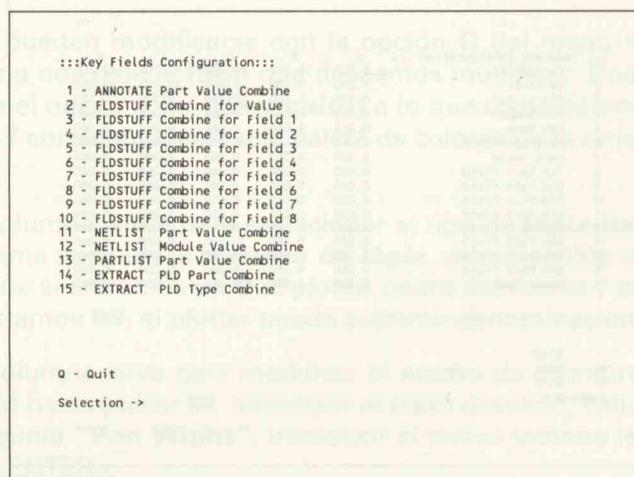


Figura 1.8

Seleccionemos el número del campo a configurar. Una barra iluminada aparece sobre la opción que deseamos elegir.

A continuación, introducir la combinación de campos del componente que se van a usar, teniendo en cuenta que **Draft** reconoce los siguientes caracteres en la representación de dichos campos: **R** como **referencia**, **V** como **valor** y los números **1..8** como **campos del componente**.

Veamos un ejemplo de combinación. Supongamos un esquema en el que, en un momento determinado, podamos saber las **temperaturas máxima y mínima** de trabajo de un transistor cuya **referencia** es Q1 y cuyo valor es **BC109**. Para asignar estas temperaturas al componente, **editaremos** éste utilizando el comando **Edit** y en la opción "**1st Part Field**" tecleamos, por ejemplo "**Tmax=75.°**" y en la opción "**2nd Part Field**" tecleamos "**Tmin=-10.°**".

Si queremos, por ejemplo, que la **temperatura máxima** aparezca junto al **valor BC109** a la hora de efectuar la numeración automática de componentes mediante la utilidad **ANNOTATE**, en la configuración de **Draft**, al seleccionar **KF**, elegiremos la opción 1, y cuando aparezca la barra de estado, introduciremos **V 1**, mientras que si queremos que aparezca la temperatura mínima, introduciremos **V 2**.

## Actualizando la configuración

Una vez configurado **OrCAD/SDT III** de acuerdo con nuestras necesidades, podemos guardar esta configuración, de modo que cada vez que arranquemos **DRAFT**, ésta se nos cargue en memoria. Para ello, basta teclear la opción **U (Update)** desde el menú de configuración.

En caso de que no se quiera guardar la configuración, podemos abandonar y salir al MS-DOS con la opción **Q (Quit)**. También podemos arrancar directamente el programa desde el menú de configuración, mediante la opción **R (Run)**. Previamente se nos preguntará si queremos actualizar la configuración.



# Filosofía de trabajo de OrCAD/SDT III

Antes de comenzar con los comandos de **DRAFT**, es conveniente analizar la forma de organizar los diseños que se pueden confeccionar con este programa.

La filosofía de trabajo parte del concepto de esquema electrónico. Normalmente, cuando se quiere realizar un diseño electrónico, el primero de los planos que se dibuja, es el llamado **DIAGRAMA DE BLOQUES**, que nos da una idea general del contenido del diseño, mediante un simple vistazo. Este tipo de diagrama es muy sencillo y consta generalmente de una serie de rectángulos o cuadrados (bloques) con un nombre en su interior que representan las diferentes partes de que consta el diseño, unidos entre sí mediante conexiones unifilares.

A continuación, se confecciona uno o varios planos denominados **DIAGRAMAS DE MONTAJE**. Es un diagrama más completo que el anterior, y en él se especifican una serie de detalles como **terminales de alimentación, terminales de entrada, terminales de salida**, etc., siendo la conexión entre las distintas partes más precisa. Con este tipo de diagrama se obtiene una idea del funcionamiento entre los distintos módulos de que se compone el diseño.

Por último, existe un número indeterminado de planos que detallan tanto el funcionamiento como la composición a nivel de componentes que integran cada uno de los módulos. Son los llamados **DIAGRAMAS DE DETALLE** o simplemente **MODULOS**.

Además de estos tipos de diagramas, también se incluyen una serie de planos auxiliares de montaje de algunos componentes especiales y los planos de **PCB (Printer Circuit Board)**, es decir, los planos de Placas de Circuito Impreso.

**OrCAD/SDT III** utiliza los tres primeros tipos, a los que llama **estructuras**, y los almacena en tres tipos de archivos:

- **ARCHIVO PLANO (FLAT FILE).**
- **ARCHIVO JERARQUICO (HIERARCHY FILE)**
- **ARCHIVO DE UNA SOLA HOJA (ONE SHEET FILE)**

Quizá no sea muy afortunada la traducción, pero lo importante es conocer el verdadero significado.

## Archivo plano o diagrama de bloques

**OrCAD/SDT** gestiona este tipo de diagramas como una estructura formada por varios módulos conectados entre sí por terminales de entrada o de salida (**module ports**).

Estos módulos contienen en su interior componentes, conexiones, buses, uniones, etc. o bien cualquier otro tipo de **objeto**, aunque, en realidad, un esquema de este tipo queda definido exclusivamente por dos **conceptos**:

- **El nombre asignado a cada módulo.**
- **El nombre de las señales presentes en los terminales.**

En la figura 1.9 se muestra un ejemplo de diagrama de este tipo, perteneciente a una **etapa amplificadora**.

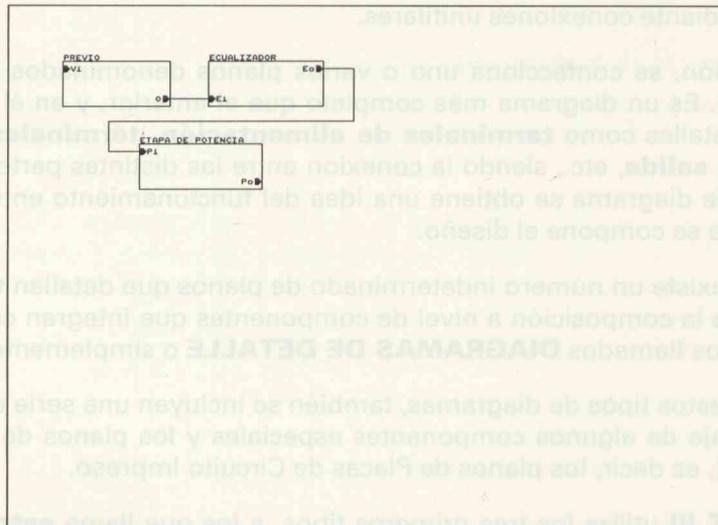


Figura 1.9

## Archivo jerárquico

Cuando se vayan a diseñar esquemas complejos, formados por varios módulos, es conveniente realizar un tipo de diseño que cumpla dos condiciones esenciales:

- **Visualizar el diseño completo.**
- **Visualizar el contenido de cada módulo.**

A la vez, cada módulo puede contener a su vez uno o varios módulos y así sucesivamente hasta aproximadamente 200 niveles.

Se obtiene así un **archivo jerárquico**, formado por un archivo **raíz** a partir del cual se puede ir **descendiendo**, introduciéndonos en cada bloque y viceversa, **ascender** desde un módulo formado por un esquema hasta el esquema **raíz**.

Para ascender o descender dentro de un **archivo jerárquico**, es preciso colocarse en el interior del **módulo** cuyo contenido deseamos visualizar y desde el menú principal de **DRAFT**, pulsar **Q (Quit)** y a continuación elegir las opciones **Enter sheet** o **Leave Sheet** (validando con ENTER o con el botón izquierdo del ratón) del submenú que aparece, si queremos **descender** o **ascender** respectivamente en un archivo de este tipo.

La característica esencial de este tipo de esquemas es que cada **módulo** está interconectado con otro u otros, a través de unas conexiones llamadas **nets**, cuyo trazado se logra desde el menú principal de **DRAFT**, seleccionando la opción **Place** y a continuación **Sheet**, seguida de **Add net**.

A continuación veamos cómo se especifican los niveles de un esquema de tipo jerárquico.

En el primer nivel de jerarquía (figura 1.10) se muestra el archivo raíz titulado "**Sumador de 16 bits**". Obsérvense los 4 módulos que corresponden a 4 **sumadores de 4 bits**.

Situándonos en el interior de uno de estos módulos y seleccionando el comando **Quit**, y a continuación **Enter Sheet**, obtenemos el nivel 2 de jerarquía, correspondiente a un sumador de 4 bits, que a su vez está formado por 4 módulos de **suma completa** de 1 bit (figura 1.11).

Repetamos la operación, pero ahora colocándonos en el interior de uno de los módulos que aparecen (**fulladd**). Obtenemos el tercer nivel de jerarquía, que corresponde a un **sumador completo de 1 bit**, compuesto a su vez por dos módulos **semisumadores** (figura 1.12).

Por último, colocándonos en el interior de uno de estos bloques y repitiendo la combinación de teclas anterior, pasamos al nivel más bajo de la jerarquía que corresponde al esquema **simple** del semisumador (figura 1.13). Si quisiéramos seguir descendiendo, un mensaje nos avisaría de que no hay más niveles.

Para ascender desde aquí, bastaría elegir la opción **Leave Sheet** en lugar de **Enter Sheet**, con lo que remontaríamos niveles hasta llegar al esquema raíz.

## Archivo de esquema simple (one-sheet)

En este caso, el diseño está **completamente** contenido en el esquema. Los **terminales (module ports)** no tienen ninguna conexión con otros esquemas.

Esta opción se utiliza en pequeños esquemas correspondientes a esquemas sencillos, con pocos componentes y que por tanto no necesitan de ningún tipo de diseño especial, o bien formando el nivel más bajo dentro de un esquema jerárquico.

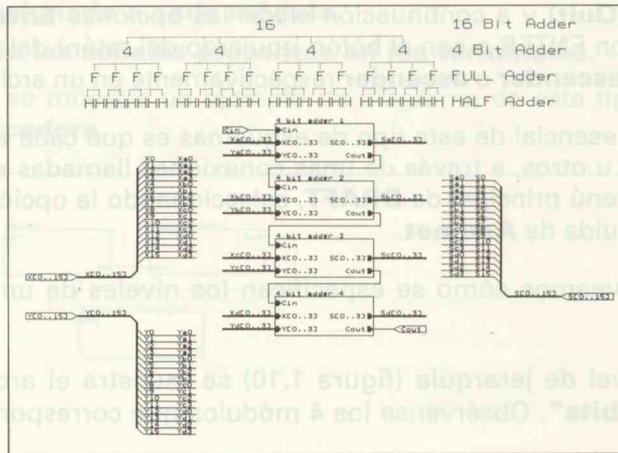


Figura 1.10

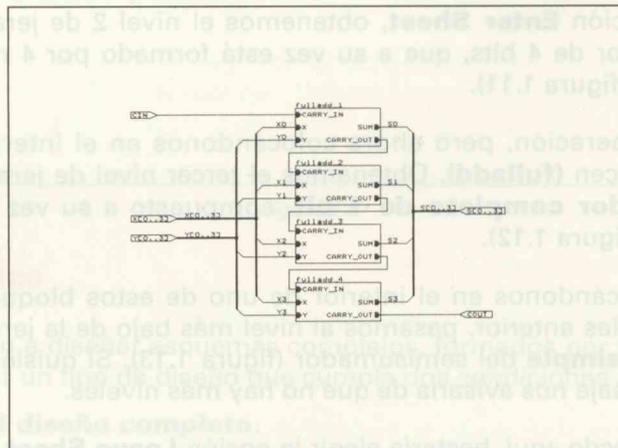


Figura 1.11

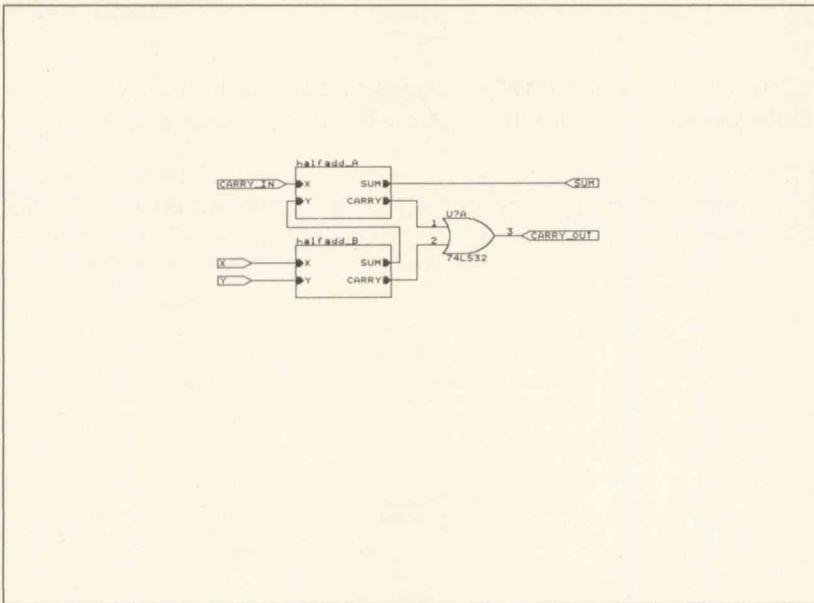


Figura 1.12

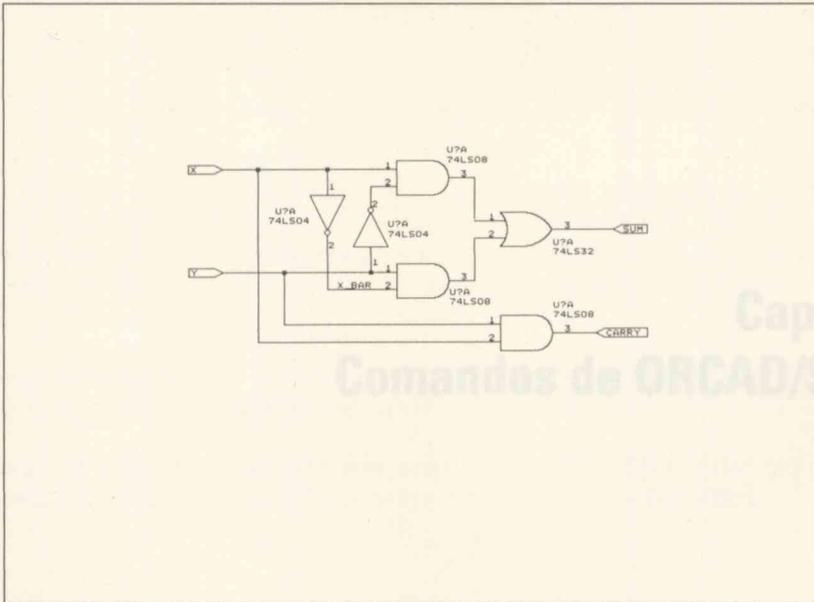


Figura 1.13



## Arrancando Draft

Suponiendo nuestro equipo configurado tanto por hardware como soft, teclearemos DRAFT desde el directorio de trabajo. Tras pulsar <ENTER> aparece el logotipo de OrCAD y un mensaje solicitándonos pulsación de una tecla para continuar.



Figura 2.1

Una vez pulsada cualquier tecla aparece un mensaje pidiendo la carga de algún fichero (Load file?). Si introducimos el nombre y extensión de cualquier archivo de trabajo aparecerá el mensaje <Working> y a continuación se mostrará el esquema en pantalla. Si deseamos comenzar un nuevo esquema, existen dos métodos de actuación:

- Teclar <ENTER> con lo que aparecerá el mensaje <Nuevo fichero>. Pulsar nuevamente <ENTER> para entrar en el menú.
- Teclar el nombre del fichero seguido de <ENTER>. Aparece el mensaje <Nuevo fichero>. Con <ENTER> entramos en el menú principal.

## Capítulo 2

### Comandos de ORCAD/SDT III



## Arrancando Draft de Draft

Suponiendo nuestro equipo configurado tanto por hardware como soft, teclearemos DRAFT desde el directorio de trabajo. Tras pulsar <ENTER> aparece el logotipo de OrCAD y un mensaje solicitándonos pulsación de una tecla para continuar.



Figura 2.2



Figura 2.1

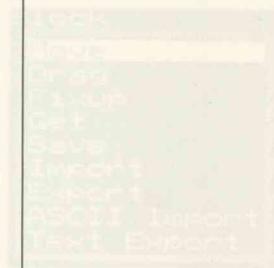


Figura 2.3

Una vez pulsada cualquier tecla aparece un mensaje pidiendo la carga de algún fichero (**Load file?**). Si introducimos el nombre y extensión de cualquier archivo de trabajo aparecerá el mensaje **<Working>** y a continuación se mostrará el esquema en pantalla. Si deseamos comenzar un nuevo esquema, existen dos métodos de actuación:

- **Teclear <ENTER>** con lo que aparece el mensaje **<Fichero desconocido>**. Pulsar nuevamente **<ENTER>** para entrar en el menú.
- **Teclear el nombre del fichero** seguido de **<ENTER>**. Aparece el mensaje **<Nuevo fichero>**. Con **<ENTER>** entramos en el menú principal.



## El menú principal de Draft

Es del tipo **pull-down (persiana)** y ocupa un espacio mínimo en la parte superior izquierda de la pantalla. Está formado por 17 **comandos iniciales** (Figura 2.2), la mayoría de los cuales se abren en nuevos **subcomandos** (Figura 2.3).

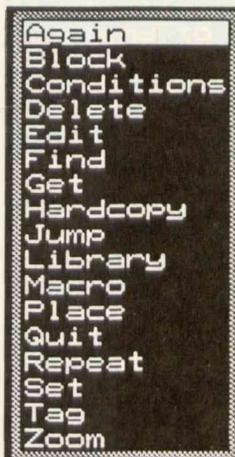


Figura 2.2

Para seleccionar cualquiera de las opciones de este **menú principal** podemos optar por tres métodos:

- 1.º Teclar la inicial del comando que deseamos seleccionar.
- 2.º Desplazarnos a través del menú mediante las flechas de cursor, pulsando <ENTER> sobre la que deseamos seleccionar.
- 3.º Desplazarnos por el menú mediante el ratón y seleccionando la opción iluminada mediante la pulsación del botón izquierdo.

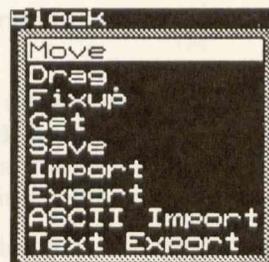


Figura 2.3



## Abandonando Draft



Figura 2.4

Para abandonar **Draft** es necesario utilizar el comando **Quit**. La selección de este comando se lleva a cabo mediante cualquiera de los tres procedimientos expuestos en la sección 2.2.

Tras la selección, aparece un submenú como el mostrado en la figura 2.4.

De las 10 opciones que aparecen en este **submenú**, las dos primeras (**Enter y Leave Sheet**) sirven para **descender** y **ascender** respectivamente, dentro de un esquema **jerárquico**, por lo que se tratarán más adelante.

Las tres últimas (**PSpice, Probe y Parts**), son utilidades relacionadas con un paquete de Simulación analógico-digital llamado **PSPICE**. El resto se explican a continuación.

### — Actualizar archivo (Update file).

Mediante esta opción se pueden salvar los cambios efectuados en cualquier esquema, así como realizar periódicas actualizaciones en esquemas complejos, en previsión de fallos de fluido eléctrico que resultarían fatales.

También puede utilizarse esta opción para almacenar hojas de trabajo que no han sido **nominadas** previamente, en cuyo caso, **DRAFT** preguntará el nombre del archivo del que queremos salvar nuestro trabajo (**Write to file?**). Basta dar un nombre seguido de <ENTER>.

### — Escribir archivo (Write file).

Esta opción permite guardar el diseño con el que estamos trabajando, en un archivo cuyo nombre podemos especificar. Cuando se selecciona, **DRAFT** devuelve la pregunta: **Write to file?**, a lo que se responderá con un nombre de archivo seguido de <ENTER>. El trabajo se guarda en dicho archivo y se vuelve al menú de **Quit**.

### — Inicializar (Initialize).

Permite cargar una hoja de trabajo en cualquier momento, limpiando o borrando de memoria cualquier otra con la que se estuviera trabajando. Antes de car-

gar la nueva hoja, nos muestra un mensaje que nos avisa de si estamos seguros o no de abandonar el trabajo en curso. Una vez se haya respondido (Yes o No) aparece el mensaje **Load file?**. Responderemos introduciendo un nombre de archivo seguido de <ENTER>. Si existe será cargado, apareciendo el esquema en pantalla. Si no existe aparece el mensaje <<**new worksheet**>>, indicándonos que se trata de una nueva hoja de trabajo. Con este método se pueden inicializar nuevas hojas de trabajo sin tener que salir del programa y arrancar de nuevo.

— **Salir al DOS (Suspend to DOS).**

Esta orden efectúa una salida al Sistema Operativo tipo **shell**, es decir, una salida que permite volver a trabajar con el programa sin tener que cargarlo de nuevo. Para ello basta teclear **Exit** desde el DOS y validar con ENTER.

**NOTA: Al utilizar esta opción, hay que recordar que el programa permanece cargado en la memoria RAM, por lo que hay que tener en cuenta la memoria disponible y no utilizar aplicaciones que puedan exceder de dicha cantidad de memoria disponible, en cuyo caso se obtendrá un mensaje del tipo "Memoria insuficiente".**

— **Abandonar la edición (Abandon Edits).**

Con esta orden, finaliza la ejecución del programa y se devuelve el control al MS-DOS. Antes de salir, un mensaje nos pregunta si estamos seguros y demanda una respuesta (yes o no).

# Adaptando Draft a nuestras necesidades

Antes de comenzar en el diseño de cualquier esquema, es necesario tener en cuenta una serie de parámetros que van a determinar de **qué modo** se presentará el diseño en la pantalla de nuestro ordenador.

Entre ellos podemos citar, por ejemplo, el tamaño del formato con el que deseamos imprimir o plotear el esquema, el ángulo de trazado de las conexiones, que aparezcan o no las coordenadas donde se sitúa el cursor, etc.

**OrCAD/SDT III**, puede ser adaptado a nuestras necesidades con el comando **Set**.



Figura 2.6

## Archivo de seguridad (Backup File)

Con este comando se pueden crear ficheros de seguridad del esquema de trabajo o bien actualizarlos cuando utilizamos el comando **Abandonar Edición** del menú desplegado por **Quit**. Tanto la creación como la actualización se producen de forma automática, por lo que hay que tener especial precaución a la hora de actualizar archivos con modificaciones no deseadas o erróneas. Al igual que el comando anterior, permite la activación o desactivación, por simple introducción de los terminos **Y** o **N**.

## Reconstrucción de buses (Wreg Buses)

Cuando se encuentra activado, este comando permite la reconstrucción de conexiones y buses que han sido desplazados anteriormente mediante la utilización del comando perteneciente al submenú de bloques, **Block Drag**.

## Señales de error (Error Bell)

Con este comando activado, cada vez que se produce un mensaje de error, el ordenador emite un pitido que nos avisa, de forma apática, de que hemos realizado una operación indebida.



## El comando "establecer" (Set)



Set	
Auto Pan	YES
Backup File	YES
Drag Buses	NO
Error Bell	YES
Left Button	NO
Macro Prompts	YES
Orthogonal	YES
Show Pins	YES
Title Block	YES
Worksheet Size	A
X,Y Display	NO
Grid parameters	
Repeat parameters	
Visible Lettering	

Figura 2.5

Para activar dicho comando basta utilizar cualquiera de los procedimientos utilizados en la sección 2.2. Aparecerá un menú como el de la figura 2.5 con los siguientes comandos:

### Autopanning

Este comando permite el desplazamiento libre de la pantalla a lo largo y ancho del formato elegido como plano de trabajo. Permite dos estados: **activado (yes)** o **desactivado (no)**. Cuando **Auto Pan** está desactivado, el cursor del ratón no puede salirse de los límites impuestos por el tamaño de pantalla. Normalmente se trabaja en el modo activado.

### Archivo de seguridad (Backup File)

Con este comando se pueden crear ficheros de seguridad del esquema de trabajo o bien actualizarlos cuando utilizamos el comando **Abandonar Edición** del menú desplegado por **Quit**. Tanto la creación como la actualización se producen de forma automática, por lo que hay que tener especial precaución a la hora de actualizar archivos con modificaciones no deseadas o erróneas. Al igual que el comando anterior, permite la activación o desactivación, por simple introducción de los términos **Y** o **N**.

### Reconstrucción de buses (Drag Buses)

Cuando se encuentra activado, este comando permite la **reconstrucción de conexiones y buses** que han sido desplazados anteriormente mediante la utilización del comando perteneciente al submenú de bloques, **Block Drag**.

### Sonido de error (Error Bell)

Con este comando activado, cada vez que se produce un mensaje de error, el altavoz del ordenador emite un pitido que nos avisa, de forma acústica, de que hemos realizado una operación indebida.

## Activación del botón izquierdo del ratón (left button)

Además del menú principal de **DRAFT**, que como dijimos anteriormente, es del tipo **persiana**, existen otra serie de submenús, caracterizados porque sus comandos, se encuentran en una **barra de estado**.

Cuando el comando de configuración **Left Button** se encuentra en **NO**, dichos comandos no pueden ser ejecutados directamente con el botón izquierdo del ratón, sino que es necesario que aparezcan en un menú tipo **persiana**.

Cuando el comando **Left Button**, se encuentra activado, sí se puede seleccionar cualquier opción de estos submenús, directamente, si bien es necesario mantener pulsado el botón izquierdo para desplazarse por el menú.

## Macro Prompts

Cuando está activado, este comando permite **visionar** en pantalla los comandos que integran una **macro**, cuando ésta se está ejecutando. Esta opción resulta muy útil para depurar **macros** o bien, para observar la secuencia de ejecución de comandos que componen una determinada **macro**.

## Ortogonalidad

Con esta opción activada, el trazado de **conexiones y buses** se realiza ortogonalmente, es decir, se trazan perpendicularmente entre sí. Cuando está desactivada, dicho trazado, puede ser efectuado bajo cualquier ángulo.

## Mostrar numeración de pines (Show Pin Numbers)

Cuando se encuentra activada, esta opción permite mostrar en pantalla el **número asignado a cada pin** (patilla de los C.I.). Hay casos, en que no interesa mostrar dicho número, como por ejemplo, cuando se realiza un **hardcopy**.

## Bloque de título (Title Block)

Esta opción activada permite colocar de forma automática el **cajetín** en la hoja de trabajo. Puede ocurrir el caso de que necesitemos de un **cajetín personalizado**, en

cuyo caso tendremos que desactivar esta opción y realizar uno nuevo mediante los comandos **Colocar conexiones** y **Colocar texto**, pertenecientes al subcomando **Place** del menú principal.

## Tamaño de la hoja de trabajo (Worksheet Size)

Cuando seleccionamos esta opción, se nos muestra un pequeño submenú (figura 2.6) que nos da a elegir hasta 5 tamaños de la hoja de trabajo. Estos están referenciados con las letras A, B, C, D, E, que corresponden, respectivamente a los formatos DIN A4, A3, A2, A1, A0.



Figura 2.6

Para poder seleccionar cualquiera de estos tamaños, es necesario haberlos configurado previamente con la opción **Template Table** del menú de configuración de **DRAFT**.

## Mostrar coordenadas (X,Y Display)

Cuando se activa esta opción, la parte superior derecha de la pantalla, muestra las coordenadas en que se encuentra el cursor en cualquier momento. Estas varían conforme nos desplazamos a lo largo y ancho de la hoja de trabajo. Como referencia, el punto de coordenadas **X=0, Y=0**, se encuentra en el extremo superior izquierdo de la pantalla, mientras que el extremo inferior derecho, mostrará las siguientes coordenadas, según el tamaño de hoja elegido.

## Parámetros de plantilla (Grid Parameters)

Una plantilla (grid) es un conjunto de rectas perpendiculares entre sí, trazadas a una distancia fija o paso, que puede ser modificada. **OrCAD** utiliza un **grid** cuyos parámetros se pueden modificar desde el menú de configuración.

Cuando seleccionamos esta opción desde el menú de **SET**, se nos muestra un pequeño submenú con tres opciones, cada una de las cuales admite dos posibilidades:

Activación (YES).

Desactivación (NOT).

Dichas opciones son las mostradas en la figura 2.7, siendo su traducción la siguiente:

- Referencias de plantilla (Grid References).
- Permanecer en plantilla (Stay on Grid).
- Visualizar puntos de plantilla (Visible Grid Dots).

## Referencias de plantilla

Con esta opción activada, aparece un borde alfanumérico en pantalla, que nos puede servir de área de referencia cuando trabajamos con esquemas con alta densidad de componentes y tamaños por encima de DIN A3. Estas áreas se nombran mediante una letra que corresponde a cada una de las columnas situadas en el margen superior de la pantalla y un número, correspondiente a cada fila, situado en el borde izquierdo de la pantalla, como muestra la figura 2.8.



Figura 2.7

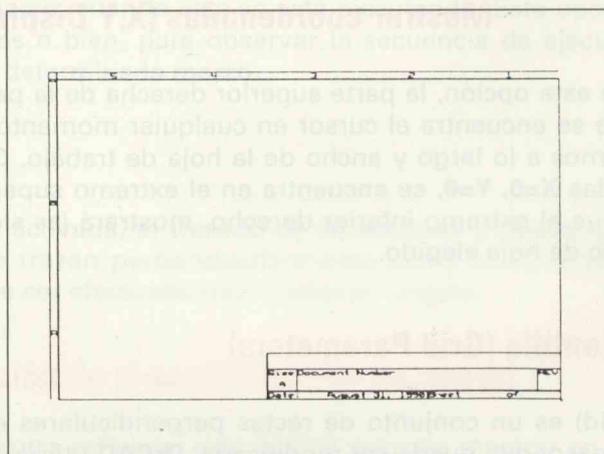


Figura 2.8

El tamaño de las áreas definidas por dichas referencias, varía de un formato a otro.

## Permanecer en plantilla

Cuando esta opción se encuentra activada, el cursor se desplaza a través de la hoja de trabajo a intervalos que corresponden al tamaño de **grid** definido con **Template Table**.

Cuando está desactivada, esta opción permite moverse con una resolución 10 veces mayor que la del **grid** preseleccionado, lo que se traduce en una **lentitud** de desplazamiento del cursor.

Esta opción debe permanecer activada siempre y cuando queramos hacer uso de las utilidades **NETLIST** y **ERC** (ver capítulo 4), ya que la colocación de **objetos** fuera del **grid** produciría mensajes de error.

## Visualizar puntos de la plantilla

Cuando esta opción se encuentra activada, en pantalla aparece un entramado mostrando de forma visible el paso empleado. Este varía según el tipo de escala del **zoom** empleado. Así, para un **zoom** de escala 1, el es una 1/10 parte de la unidad en que vienen dadas las coordenadas. Para una escala de **zoom** de 2, el paso es 2/10; para una escala de **zoom** de 5, es de 1/2; para una escala de 10, el paso es de 1, mientras que para una escala de 10, el paso es 2.

## Repetición de parámetros

En el menú principal de **DRAFT** existe un comando llamado **REPEAT** (Repetir) cuya función, como su nombre indica, es la de repetir el comando que se ha ejecutado de forma inmediatamente anterior. Hay algunas funciones que se pueden modificar utilizando el comando **Repetir Parámetros** del submenú **SET**.



Figura 2.9

Cuando se selecciona esta opción, **DRAFT** devuelve un submenú como el de la figura 2.9, donde se observan 4 opciones:

### Repetir Abcisa (X Repeat Step)

Cuando seleccionamos esta opción, el programa espera la entrada de un número entero (positivo o negativo) que determina el avance (o retroceso) automático de cualquier **objeto**, a lo largo del eje X, cada vez que se ejecuta el comando **Repeat** desde el menú principal. Si la entrada es un 0, no hay ningún tipo de desplazamiento cuando se ejecuta **Repeat**. Este avance depende de si la opción **Stay on grid**, está activada o no.

### Repetir ordenada (Y Repeat Step)

Realiza una función similar a la anterior, pero los desplazamientos se producen a lo largo del eje Y.

## Repetir etiqueta (Label Repeat Delta)

Cuando se selecciona esta opción, el programa espera la entrada de un número entero, que puede ser positivo o negativo. Cuando es positivo, cualquier **sufijo** numérico incluido en una etiqueta, se incrementa en dicho número, cada vez que se ejecute el comando **Repeat**. Por el contrario, si el número es negativo, se decrementa. Este subcomando funciona conjuntamente con el que se detalla a continuación.

## Colocar con autoincremento (Auto Increment Place)

Cuando esta opción se encuentra activada (YES), los nombres de las etiquetas se incrementan o decrementan automáticamente (según el número positivo o negativo que hayamos introducido en **Repetir etiqueta**) cuando se ejecuta el subcomando **Label** (seguido de **Place**), de la opción **PLACE** del menú principal. Cuando esta opción se encuentra desactivada, no se producen dichos cambios.

## Visualizar rótulos (Visible Lettering)

En un esquema, las diferentes partes que lo forman (componentes, conexiones, buses, terminales, etc.) van acompañadas, casi todas ellas, por algún **título** o **cadena** de caracteres, que las identifica o diferencia de otras. Así, por ejemplo, un **circuito integrado** muestra, cuando es seleccionado, el número de pines, el nombre, la referencia, etc.

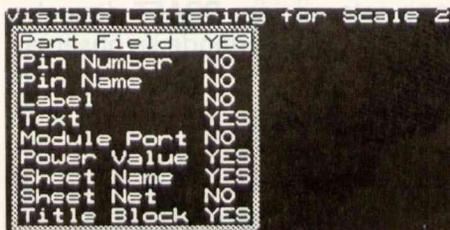


Figura 2.10

Cuando accedemos a la **escala 2** del comando **Zoom**, los rótulos no se muestran en pantalla, a no ser que coloquemos en **YES** los diferentes subcomandos de la figura 2.10 que aparecen cuando se selecciona la opción **Visible Lettering**. Para configurar dicha selección, basta elegir esta opción desde el menú de **SET**, con lo que se mostrará el submenú de la figura 2.10.

## ¿Qué se necesita para comenzar?

Suponemos configurado DRAFT, tanto para el hardware disponible (tipo de monitor, impresora, ratón, plotter, etc.) como para el software (directorios y subdirectorios, colores de pantalla, etc.).

A continuación, arrancamos siguiendo el método expuesto en la sección 2.1.

Vamos a crear un esquema simple (ONE SHEET) al que llamaremos RELÉS. Su esencia consiste en un circuito cuya función es la de controlar hasta 8 relés desde un microcontrolador, aprovechando las señales presentes en el puerto paralelo.

El corazón de dicho circuito es un latch de 8 bits direccionable, de tecnología CMOS (4099), capaz de seleccionar sus ocho salidas, mediante la combinación de sus tres entradas (A0, A1, A2).

La señal de ENABLE se consigue mediante 4 puertas EXOR (4000) -cada una de ellas conectada a uno de los bits de direcciones A0..A3, cuyos niveles de salida son recogidos por una NOR de 4 entradas (4002).

Cada una de las salidas del 4099 se conecta a la base de un transistor (BC547) que cuando entra en saturación activa el relé correspondiente.

Una vez aclarado el tipo de esquema que vamos a diseñar, hay que tener en cuenta los siguientes requerimientos:

### Comprobaciones previas

Esencialmente se reducen a tres:

- Comprobar que las opciones del comando SET se adaptan a las necesidades de nuestro diseño.
- Comprobar la cantidad de memoria libre.
- Comprobar la inclusión de los componentes necesarios en las librerías que han sido cargadas.

## Capítulo 3 Una sesión con Draft



## ¿Qué se necesita para comenzar?

Suponemos configurado **DRAFT**, tanto para el **hardware** disponible (tipo de monitor, impresora, ratón, plotter, etc.) como para el **software** (directorios y subdirectorios, colores de pantalla, etc).

A continuación, arrancamos siguiendo el método expuesto en la sección 2.1.

Vamos a crear un esquema simple (ONE-SHEET) al que llamaremos **RELES.SCH**. En esencia consiste en un circuito cuya función es la de controlar hasta **8 relés** desde un microordenador, aprovechando las señales presentes en el puerto paralelo.

El corazón de dicho circuito es un latch de 8 bits direccionable, de tecnología **CMOS** (4099), capaz de seleccionar sus ocho salidas, mediante la combinación de sus tres entradas (A0, A1, A2).

La señal de **ENABLE** se consigue mediante 4 puertas **EXOR** (4030), cada una de ellas conectada a uno de los bits de direcciones A0...A3, cuyos niveles de salida son recogidos por una **NOR** de 4 entradas (4002).

Cada una de las salidas del 4099 se conecta a la base de un transistor (BC547) que cuando entra en **saturación** activa el relé correspondiente.

Una vez aclarado el tipo de esquema que queremos diseñar, hay que tener en cuenta los siguientes requerimientos:

### Comprobaciones previas

Esencialmente se reducen a tres:

- Comprobar que las opciones del comando SET se adaptan a las necesidades de nuestro diseño.
- Comprobar la cantidad de memoria libre.
- Comprobar la inclusión de los componentes necesarios en las librerías que han sido cargadas.

## Personalización del comando Set



Figura 3.1

En la figura 3.1 se muestra un ejemplo de configuración con el comando SET, mientras que la figura 3.2 muestra un ejemplo de configuración del submenú parámetros de grid.



Figura 3.2

Para las dos últimas opciones del submenú, la configuración es la mostrada en las figuras 3.3 y 3.4 respectivamente.



Figura 3.3

## Cantidad de memoria RAM disponible

Para ello basta seleccionar la opción **CONDICIONES (Conditions)** desde el menú principal de **DRAFT** que nos va a permitir monitorizar en cualquier momento lo siguiente:



Figura 3.4

## Cantidad de memoria de la hoja de trabajo (Worksheet Memory size)

Muestra la cantidad de memoria utilizada por nuestro diseño, en bytes. Con una hoja en blanco y utilizando 640K de RAM, este tamaño es por defecto de 341 bytes, como muestra la figura 3.5.

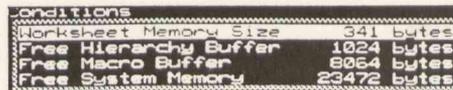


Figura 3.5

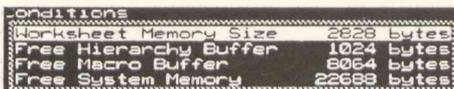


Figura 3.6

Conforme avanzamos en el diseño de nuestro esquema, este tamaño aumentará paulatinamente. Así, si activamos el comando **Condiciones** con una **hoja** cargada, veremos como la cantidad de memoria disponible ha disminuido proporcionalmente (ver figura 3.6).

### **Cantidad de memoria libre del buffer de jerarquía (Free Hierarchy Buffer)**

Muestra la cantidad de memoria útil en el buffer de jerarquía, que como vimos (Configuración) sirve para almacenar las hojas de trabajo y rutas de búsqueda de un esquema jerárquico.

### **Cantidad de memoria libre del buffer de macros (Free Macro Buffer)**

Muestra la cantidad de memoria libre del buffer utilizado para almacenar las **macros** definidas on-line, es decir, las macros definidas desde el programa.

### **Cantidad de memoria libre del sistema (Free System Memory)**

Muestra la cantidad de memoria RAM que queda libre, conforme avanzamos en nuestro diseño. Disminuye en proporción al aumento de memoria ocupado por la hoja de trabajo.

### **Comprobar que los componentes que se van a utilizar en el diseño (o al menos la mayoría), están incluidos en las librerías seleccionadas durante el proceso de configuración**

Esto no es necesario cuando se adquiere experiencia en el manejo del programa, pues con el uso se memorizan gran cantidad de componentes y por lo tanto, siempre sabremos si los componentes que vamos a utilizar en nuestro esquema están incluidos en las librerías o no.

No obstante, ocurrirá, sobre todo al comenzar a manejar el paquete, que habrá componentes que queramos incorporar a un esquema y no sabremos con certeza en qué librería se encuentran. En caso de que no se encuentren en ninguna de las librerías disponibles, no quedará más solución que **crearlos** e incorporarlos a una librería ya existente o a una nueva **propia**. Pero no adelantemos acontecimientos.

En el ejemplo que nos ocupa, utilizaremos las siguientes librerías para extraer los componentes necesarios:

- **CMOS.LIB** para el 4099, 4030 y 4002.
- **DEVICE.LIB** para el resto de componentes.

Esto no significa que sean las únicas librerías que deban cargarse, pero sí las necesarias para este caso.

Para saber si existe un componente en una librería, u obtener un listado de los componentes pertenecientes a cualquier librería, disponemos de un comando en el menú principal de **DRAFT**, llamado precisamente **Librería (Library)**.

## El comando Library

Cuando activamos este comando, nos aparece en pantalla un submenú de dos opciones (ver figura 3.7), que podemos interpretar como:



Figura 3.7

— **Mostrar directorio**

— **Hojear una librería**

— **Mostrar directorio.** Este subcomando permite seleccionar una librería y mostrar un listado de los componentes que la integran.

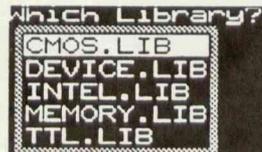


Figura 3.8

Al activar esta opción, se muestra un listado de las librerías disponibles, las cuales se han cargado desde el menú de configuración, pidiéndonos mediante el mensaje **Which Library?** la selección de una librería (ver figura 3.8).

Seleccionemos aquella librería cuyos componentes deseamos visionar (INTEL en nuestro ejemplo), con lo cual se mostrará un nuevo submenú que nos pide el tipo de **dispositivo** por el que deseamos sacar el listado.

Disponemos de tres dispositivos: **pantalla, impresora y archivo.**

**Pantalla.** Si elegimos este dispositivo como salida del listado de los componentes que integran la librería, obtendremos dicho listado a través del monitor de nuestro ordenador.

Cuando el listado ocupa más de una pantalla, aparece un mensaje pidiéndonos la pulsación de tecla para continuar.

**Impresora.** El listado se efectúa a través de la impresora conectada a nuestro ordenador. Un listado en este tipo de dispositivo resulta muy útil a la hora de iniciar un esquema ya que nos permite saber si los componentes que van a integrar nuestro esquema se encuentran o no en las librerías seleccionadas previamente.

**Archivo.** Con esta opción, el listado se almacena en un fichero de **texto** que puede ser leído por cualquier procesador de texto que incorpore esta opción. El interrogante **File?** nos preguntará el nombre que queremos dar al archivo.

En la figura 3.9 se ve el resultado obtenido tras elegir la primera opción.

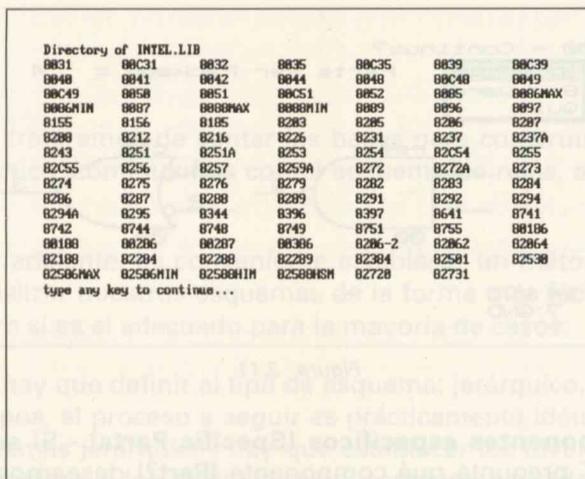


Figura 3.9

**Hojear una librería.** - Con este subcomando podemos visualizar el contenido de una librería, componente por componente, o bien seleccionar directamente un componente determinado.



Figura 3.10

Al activar dicho subcomando, aparece en pantalla un submenú como el de la figura 3.10 donde gozamos de dos posibilidades:

- 1.º Mostrar todos los componentes (All Parts).** Activando esta opción, el programa muestra las librerías cargadas previamente. Seleccionamos aquella que queramos visualizar (en nuestro ejemplo la TTL), apareciendo en pantalla un pequeño menú con las opciones **Avance**, **Retroceso** y **Abandonar**, cuya acción es obvia.

Además, aparece también en pantalla el primer componente de la librería (7400 en nuestro ejemplo) con indicaciones tales como los terminales de alimentación, número de puertas por pastilla, etc.

Habrán componentes (microprocesadores sobre todo) que no cabrán en la pantalla. Para obtener una visualización completa de dichos componentes, será necesario utilizar un nivel de **zoom** conveniente.

La figura 3.11 muestra el ejemplo seleccionado.

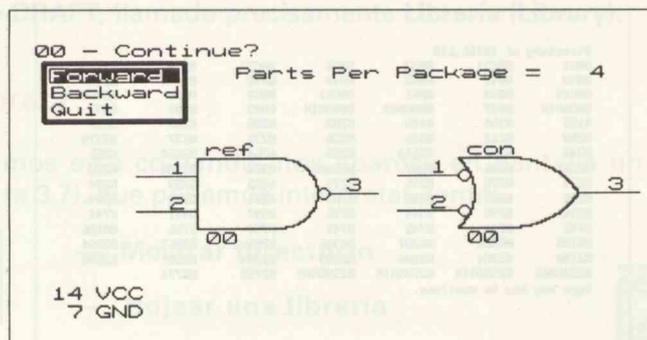


Figura 3.11

**2.ª Mostrar componentes específicos (Specific Parts).**- Si seleccionamos esta opción, DRAFT pregunta qué componente (**Part?**) deseamos visualizar. Bastará introducir el nombre del componente (74LS00 en nuestro ejemplo), seguido de <ENTER> para obtener una visualización del mismo.

## Un método de trabajo

En esta sección trataremos de sentar las bases para construir cualquier esquema. Como ejemplo práctico, continuamos con el esquema de relés, al que hemos titulado **RELES.SCH**.

Antes de seguir adelante, es conveniente establecer un método de trabajo lógico, que nos permita realizar nuestros esquemas de la forma más fácil posible. No pretende ser el mejor, pero sí es el adecuado para la mayoría de casos.

En primer lugar hay que definir el tipo de esquema: jerárquico, plano o simple. Para los dos primeros tipos, el proceso a seguir es prácticamente idéntico con la diferencia de que en los esquemas jerárquicos hay que establecer los niveles de jerarquía, asignando nombres (filename) a los archivos correspondientes a cada nivel. Para la confección de un esquema simple los pasos a seguir son los siguientes:

1. **Colocación de componentes previamente extraídos de las librerías.** Se llevará a cabo utilizando el comando **Get** y sus subcomandos asociados. A veces, es necesario establecer un nivel 2 de zoom con tal de ver un plano de conjunto de todos los componentes ya colocados.

**Nota:** Puede ocurrir que existan componentes u **objetos** que no se encuentren en ninguna de las librerías que se suministran con el paquete, en cuyo caso habrá que **crearlos**. Para ello se utiliza la utilidad **LIBEDIT** que se estudiará ampliamente en el capítulo 4.

2. **Conexionar las diferentes patillas de los componentes que intervienen en el esquema.** Para ello haremos uso del comando **Place** y del subcomando **Wire** o **Bus**, según sea el tipo de conexión que deseemos establecer.
3. **Colocación de uniones.** Es una tarea obligada, ya que es la única forma de distinguir los cruces de conexiones, de las verdaderas uniones. Utilizaremos el subcomando **Junction** de **Place**.
4. **Colocación de terminales.** Haremos uso del subcomando **Module Port** perteneciente al comando **Place**. La colocación de terminales es tarea obligatoria, en esquemas de tipo jerárquico y también se usa en diagramas de bloques, ya que de lo contrario, al chequear el circuito, se producirían mensajes de error.

5. **Colocación de texto y líneas auxiliares.** Una vez finalizado el esquema, éste se completará con las anotaciones que sean necesarias, modificando o trasladando aquellas que lo requieran e incluyendo el relleno del cajetín. Además, se trazarán las líneas necesarias para delimitar partes del esquema.

6. **Modificaciones y posibles mejoras.** En la mayoría de los casos, esta labor, en principio de carácter opcional, no resulta necesaria, pero sí conveniente, cuando vayamos a utilizar el esquema diseñado para posteriores aplicaciones.

Durante la ejecución de esta fase se suelen efectuar una serie de tareas como:

- Copiar una parte del esquema en otro lugar, ya sea del mismo u otro esquema diferente.
- Sustituir un componente por otro equivalente y reconstruir las conexiones.
- Editar algunas referencias o valores de los componentes con objeto de situarlos en lugares más apropiados.
- Mover bloques, mejorar la colocación de terminales y sustituirlos por **buses** si es necesario.

7. **Chequeo del circuito.** Para ello, el paquete incluye una serie de utilidades que nos van a permitir desde la numeración automática de componentes, hasta la creación de un informe que contenga el número de componentes que interviene en el esquema con su valor y referencia, pasando por chequeos eléctricos de conexiones y creación de ficheros para PCB.

Veamos de forma exhaustiva como se desarrolla cada paso y los comandos implicados a que da lugar.

**Nota: las pulsaciones de <ENTER> también equivalen a pulsar el botón izquierdo del ratón, mientras que las de <ESCAPE> equivalen a la pulsación del botón derecho.**

## Paso 1: Extracción y colocación de componentes

Una vez arrancado el programa y **comprobado** que todos los componentes que van a intervenir en el esquema se encuentran disponibles en las librerías (utilizar el comando **Library**) es necesario **extraerlos** y **colocarlos** en nuestra hoja de trabajo. Esto se realiza mediante el comando **GET** (TOMAR).

### El comando GET (Tomar)

Como su nombre indica, este comando permite **extraer** un componente desde cualquier librería y **colocar**lo en la hoja de trabajo de tres formas diferentes:

- **Modo normal.**
- **Girado.**
- **Convertido.**

Podemos, pues, considerar la actuación del comando Get dividida en dos partes: **extracción y colocación.**

### 1.ª Extracción

Existen dos formas de extraer un componente desde una librería:

- **Directamente.**
- **Seleccionando previamente la librería correspondiente.**

Para extraer un componente **directamente**, seleccionar el comando **Get** desde el menú principal y a continuación, a la pregunta **Get?** responder con el nombre de dicho componente, seguido de <ENTER>. Si el componente no se encuentra en ninguna de las librerías seleccionadas, aparece un mensaje de error (... **was not found**).

**Nota:** Existe una cierta flexibilidad a la hora de introducir el nombre del componente. Así, por ejemplo, podemos extraer el CI **74LS03** respondiendo a la pregunta **Get?**, de varias formas:

- Introduciendo textualmente 74LS03 <ENTER>.
- Introduciendo LS03 <ENTER>.
- Introduciendo simplemente 03 <ENTER>.



Figura 3.12

Mediante las dos primeras, el componente es extraído directamente de la librería y se nos muestra **desconectado** (outline), dentro de un submenú que llamaremos de **colocación**. Utilizando la última forma, **DRAFT** nos muestra las posibilidades o variedades (consumo, velocidad, etc.) de que se dispone.

En nuestro ejemplo nos aparece un menú como el de la figura 3.12.

En el caso de no tener la seguridad de que un componente se encuentra en una determinada librería, o bien se carece de la suficiente práctica, es aconsejable **extraer** los componentes **seleccionando previamente la librería**. Para ello, una vez seleccionado el comando **Get** desde el menú principal, responder con <ENTER> a la pregunta **Get?**

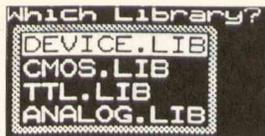


Figura 3.13

Aparecerá un menú donde se muestran las librerías que hayan sido configuradas. Seleccionemos aquella a la cual pertenece el componente que queremos extraer. Pulsando <ENTER> aparece una ventana con los **10** primeros nombres de los componentes que forman la librería.

En nuestro ejemplo, seleccionando la librería **TTL**, aparece la ventana de la figura 3.13. Notemos como aparecen los diez primeros componentes (figura 3.14). Seleccionemos **03** y tras pulsar <ENTER> nos encontraremos con el componente en el menú de **colocación** donde ya podremos situarlo como queramos.



Figura 3.14

## 2.ª Colocación

Una vez extraído el componente de la librería, éste aparece representado de una forma especial, que llamaremos **desconectada (outline)**. Esta representación se caracteriza porque el componente extraído se muestra durante un par de segundos como una silueta en la que se advierte el **tamaño del componente**, las **pastillas de conexión** y la **identificación**, que aparece **sombreada**.

En la parte superior de la pantalla, aparece el menú de colocación de componentes.

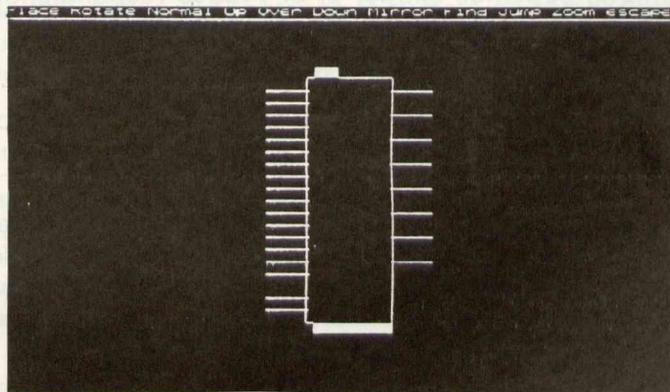


Figura 3.15

En esta situación, el componente puede ser **desplazado** rápidamente a través de la pantalla mediante el ratón o las teclas de cursor.

Es curioso observar (figura 3.15) que cuando lo dejamos fijo, durante unos momentos, la identificación se hace visible (numeración de pines, nombre del componente, etc.), mostrándose de idéntica forma a como será colocado en el esquema.



Figura 3.16

## El menú de colocación de componentes

Cuando el componente extraído, se encuentra situado en la zona deseada de nuestra hoja, pulsamos <ENTER> con lo que se abre un menú como el de la figura, al que llamaremos **menú de colocación de componentes**. Como se observa en la figura 3.16, se compone de 11 opciones, cada una de las cuales se detalla a continuación.

**Place (Situar).**- Este subcomando sitúa el componente en la hoja de trabajo.

**Rotate (Girar).**- Este subcomando gira el componente 90 grados en el sentido de las agujas del reloj.

**Convert.**- Existen numerosos componentes, sobre todo de la serie TTL, que admiten ser convertidos en sus equivalentes de Morgan. Con este subcomando se puede conseguir este objetivo, pudiéndose ver el componente convertido dejando estacionario el símbolo de **desconectado**.

**Normal.**- Como su nombre indica, devuelve la situación de un componente a su estado original, es decir tal y como se ha extraído de la librería.

**Up.**- Equivale a efectuar una rotación o giro de 90 grados en el sentido de las agujas del reloj.

**Over.**- Equivale a una rotación de 180 grados en el sentido de las agujas del reloj.

**Down.**- Equivale a una rotación de 270 grados en el sentido de las agujas del reloj.

**Mirror.**- Este subcomando produce una imagen **reflejada** del componente. La figura 3.17 muestra las diferentes posiciones que puede adoptar un componente mediante la utilización de estos subcomandos.

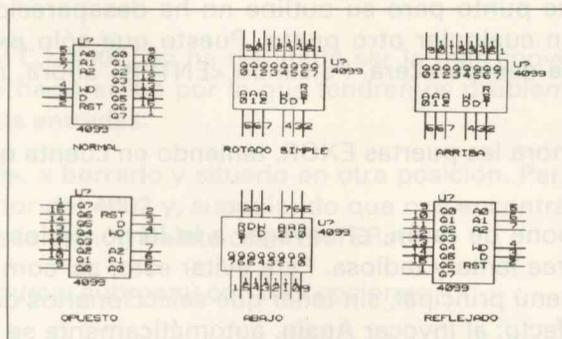


Figura 3.17

Los tres subcomandos siguientes, que aparecen en este submenú (**Find, Jump, Zoom**), desempeñan funciones muy similares a las realizadas por comandos de su mismo nombre que se encuentran en el menú principal, por lo que los estudiaremos más adelante.

Además, su utilidad más concreta tiene lugar en la modificación y sustitución de componentes, que se tratará más en profundidad en el **paso 6**.

El subcomando **Escape** nos devuelve al menú principal y además sirve para anular cualquier orden anterior.

## Aplicación práctica

Volviendo a nuestro esquema de relés, vamos a proceder a situar los componentes de que consta utilizando la **teoría** que acabamos de ver.

Asegurémonos de que el subcomando **Mostrar coordenadas** (X Y Display) del menú del comando **SET** se encuentra en ON. Esto es necesario ya que vamos a utilizar coordenadas para situar los componentes de nuestro esquema.

Suponemos **DRAFT** configurado y arrancado y en pantalla se nos muestra el menú principal.

Seleccionemos el comando **Get** y tras pulsar <ENTER> respondamos a la pregunta con **4099** y <ENTER>. Rápidamente observaremos cómo la silueta **desconectada** de dicho CI se nos muestra en pantalla. Mediante las flechas de cursor o con el ratón, desplacémonos a lo largo y ancho de la hoja de trabajo y observemos cómo cambian las coordenadas situadas en el margen superior derecho. Situémonos en el punto **X=5.70 Y=1.20** y pulsemos <ENTER>. El **menú de colocación** se despliega en la parte superior izquierda. Seleccionemos el subcomando **Place** seguido de <ENTER>. El 4099 quedará fijado en este punto pero su outline no ha desaparecido por si deseamos colocarlo **de nuevo** en cualquier otro punto. Puesto que sólo existe uno, pulsemos <ESCAPE> y el **outline** desaparecerá. Pulsando <ENTER> ahora, retornamos al menú principal de **DRAFT**.

Vamos a colocar ahora las puertas EXOR, teniendo en cuenta que son **cuatro** y que habrá que girarlas.

Cuando no se dispone de **ratón**, el navegar a lo largo de los menús y submenús puede resultar una tarea lenta y tediosa. Para evitar esto, así como para poder repetir comandos desde el menú principal, sin tener que seleccionarlos de nuevo, se utiliza el comando **Again**. En efecto; al invocar **Again**, automáticamente se muestra en pantalla el último comando utilizado.

Volviendo a nuestro ejemplo, al invocar **Again**, se nos mostrará en pantalla la palabra **Get?** que nos demuestra la repetición de dicho comando. Responderemos introduciendo la secuencia **4030** seguida de <ENTER>. Aparecerá el **outline** correspondiente. Situémonos en el punto **X=1.60 Y=3.80** y pulsemos <ENTER>. Se despliega el menú de colocación, pero ahora no seleccionaremos la opción **Place** sino **Down** seguido de <ENTER> con lo que el **outline** girará 180 grados en sentido horario. Seleccionando ahora **Place** <ENTER> el componente quedará situado definitivamente. ¡**ATENCIÓN!** Si pulsamos <ESCAPE> perderemos el **outline**, lo cual no nos interesa, ya que tenemos que colocar tres circuitos más.

Para ello, desplazemos el **outline** hasta el punto **X=2.60 Y=3.80** y situémoslo con **Place**. Repitamos la operación para los puntos de coordenadas **X=3.60 Y=3.80** y **X=4.60 Y=3.80**. Cuando hayamos colocado este último circuito, pulsemos <ESCAPE> y a continuación <ENTER> con lo que nos encontraremos de nuevo en el menú principal. Vamos a situar a continuación el sumador **4002**. Para ello seguimos un procedimiento análogo al anterior. Sinteticemos los pasos a seguir, de forma que nos sirva de corolario.

- 1.º Invocar **Get**. Utilizar **Again** si se ha utilizado **Get** por última vez. Pulsar <ENTER>.
- 2.º Responder con la secuencia **4002** ante la pregunta **Get?** y a continuación pulsar <ENTER>. Aparecerá el **outline**.
- 3.º Desplazar el **outline** hasta situarlo en la coordenada **X=3.10 Y=5.30**. Rotarlo con la opción **Down** y fijarlo con **Place**.
- 4.º Retornar al menú principal pulsando <ESCAPE> seguido de <ENTER>.

## Cómo borrar

La colocación del C.I. **4002**, no ha resultado ser la más conveniente, ya que queda un poco desplazado hacia arriba por lo que tendremos problemas a la hora de trazar las conexiones de sus entradas.

Procedamos, pues, a **borrarlo** y situarlo en otra posición. Para ello, desplazemos el cursor hasta el interior del **4002** y, suponiendo que nos encontramos en el menú principal, seleccionemos la opción **Delete** con <ENTER>.

Se despliega un nuevo submenú con tres opciones:

- **Object (Objeto)**.- Mediante esta opción, se borra el objeto (componente, pista, unión, etc.) sobre el que se encuentra el cursor.

- **Block (Bloque).**- Borra el contenido de un bloque. Esta opción la veremos más adelante.
- **Undo (Deshace).**- Recupera lo borrado con anterioridad. Esta opción resulta muy útil contra borrados accidentales.

Volviendo al ejemplo, seleccionemos **Object** con <ENTER>. En el caso de no encontrarse el cursor sobre el objeto a borrar, se obtiene un mensaje de error del tipo <<**Object not found**>>.



Figura 3.18

Aparece un nuevo submenú de cinco opciones, como el mostrado en la figura 3.18.

Las cuatro últimas opciones, son utilizadas en otros submenús de **DRAFT**, empleándose para esquemas ya confeccionados, con cierta densidad de componentes, donde haya que **buscar (Find)**, **ampliar (Zoom)**, **saltar (Jump)** hasta alguna zona donde se encuentren objetos para borrar. La opción **Escape** retorna al menú principal. Seleccionemos **Delete** con <ENTER> con lo que el C.I. **4002** desaparece de nuestra hoja de trabajo. Con <ESCAPE> retornemos al menú principal.

**Nota. Al borrar un componente, pueden quedar restos de éste, de forma momentánea. Con un scrolling de pantalla, desaparecen.**

Coloquemos de nuevo el **4002** en la posición **X=3.10 Y=5.70** siguiendo un proceso análogo al descrito anteriormente.

Una vez colocados los componentes de tipo integrado, se procede a colocar los de tipo discreto. Este orden no es obligatorio, pero sí muy útil para esquemas sencillos.

Comencemos por los **resistores**. En primer lugar vamos a situar los 4 resistores que van conectados a las entradas de las puertas **EXOR**.

Suponiendo que es la primera vez que manejamos **DRAFT**, los pasos a seguir son los siguientes:

- 1.º Invocar **Get** desde el menú principal.
- 2.º Responder con <ENTER> a la pregunta **Get?**
- 3.º Seleccionar la librería **DEVICE.LIB**. Presionar <ENTER>. Buscar **RESISTOR** y presionar <ENTER> o bien introducir directamente la palabra **RESISTOR** seguida de <ENTER>.

- 4.º Situar el **outline** en la posición **X=1.80 Y=2.60** correspondiente al primer resistor y presionar <ENTER>. Situar con **Place**.
- 5.º Desplacémonos a la posición **X=2.80 Y=2.60** y emplaceemos el segundo resistor presionando <ENTER> y situando con **Place**.
- 6.º Repetir la operación para los dos resistores siguientes, a los que colocaremos en las siguientes coordenadas:  
**X=3.80 Y=2.60 X=4.80 Y=2.60**
- 7.º Coloquemos un nuevo resistor en el punto de coordenadas: **X=5.20 Y=2.10**
- 8.º Por último, coloquemos tres resistores en posición horizontal (utilizar **Rotate** para girar el outline) en los puntos de coordenadas siguientes:  
**X=7.00, Y=1.20; X=7.10, Y=0.70; X=6.20, Y=0.10.**

**Nota.** Más adelante veremos que todo componente consta, entre otras partes, de una *referencia* y de un *valor*. En el caso concreto del resistor común, incluido en la referencia es siempre *R?*, mientras que el valor es *RESISTOR*. Por tanto podremos utilizarlo para colocar resistores de cualquier valor, aunque más tarde haya que editar el campo *valor* y sustituirlo por el valor de cada resistor. La alternativa más elegante consiste en fabricar una *librería personalizada* de la que se puedan extraer resistores con sólo invocar su *valor óhmico*.

Pasemos a continuación, a la colocación de los condensadores. Todos los tipos existentes se encuentran en la librería **DEVICE.LIB**.

Son 5 condensadores en total: 4 cerámicos de **1 nF** (C1..C4) y un electrolítico de **10 uF** (C5). El proceso a seguir comporta los siguientes pasos:

- 1.º Para extraer los cuatro primeros, utilizar el comando **Again** y a la pregunta **Get?** responder con **CAP NP** (CONDENSADOR NO POLARIZADO) seguido de <ENTER>.
- 2.º Cuando aparezca el **outline**, seleccionar la opción **Rotate** y desplazarse hasta el punto de coordenadas **X=1.00 Y=2.60**. Presionar <ENTER> y situar con **Place** y <ENTER>, con lo que quedará colocado el primer condensador.
- 3.º Sin abandonar la orientación del **outline** repitamos el proceso para los tres restantes condensadores, cuyas coordenadas son: **X=1.00, Y=3.00; X=1.00, Y=3.40; X=1.00, Y=3.80.**
- 4.º Volvamos al menú principal pulsando <ESCAPE> seguido de <ENTER>.

- 5.º Seleccionemos **Again** para repetir el comando **Get** al que respondemos con **CAPACITOR POL** (CONDENSADOR POLARIZADO).
- 6.º Traslademos el outline al punto de coordenadas **X=5.20 Y=0.50**. Fijemos con **Place** y volvamos al menú principal utilizando la secuencia <ESCAPE>, <ENTER>.

Pasemos a la colocación de los tres diodos, uno de ellos de utilización general, otro, un diodo LED y el último un diodo Zéner. Sus respectivos símbolos se encuentran en la librería **DEVICE.LIB**. Esquematicemos el proceso.

- 1.º Invoquemos **Again** para repetir el comando **Get**. Respondamos con **diode** y <ENTER> a la interrogación.
- 2.º Cuando aparezca el **outline**, seleccionemos la opción **Up** del **menú de colocación** y pulsemos <ENTER>. A continuación, desplacémonos hasta el punto de coordenadas **X=7.50 Y=0.30** y situemos el diodo con **Place** seguido de <ENTER>.
- 3.º Volvamos al menú principal. Repitamos el comando **Get** con **Again** y respondamos con **led** seguido de <ENTER>, a la interrogación.
- 4.º Cuando se muestre el **outline**, seleccionar la opción **Down** y situémoslo, mediante **Place** en el punto de coordenadas **X=6.80 Y=0.30**. Pulsar <ENTER>.
- 5.º Repitamos el proceso para el diodo Zéner. Para ello, respondamos con **diode zener** al interrogante de **Get?**. Tras validar con <ENTER> y girarlo 90 grados a la izquierda con **Up**, situemos el **outline** en la coordenada **X=5.70 Y= 0.30**. Fijar con **Place** y con <ESCAPE>, retornemos al menú principal.

Por último, pasemos a la colocación de otros **cuatro** componentes, cuyos símbolos se encuentran en la librería **DEVICE.LIB**. Para no hacer repetitiva en exceso, la tarea de colocar componentes, sugerimos al lector que sitúe los siguientes componentes en las coordenadas que se indican, siguiendo un proceso similar al efectuado anteriormente.

- 1.º Coloquemos, en primer lugar, un **pulsador**. Para ello, responder **SW PUSH-BUTTON** a la interrogación de **Get** y utilizar la opción **Up** para su colocación en el punto de coordenadas **X=4.80 Y=0.50**. Situar con **Place**. Notar que se superpone el texto de referencia con el del condensador. Para evitar esto, eliminemos dicha referencia, haciendo uso del comando **Edit**.

Para ello, retornemos al menú principal y seleccionemos **Edit** con <ENTER>. Aparece un submenú como el de la figura 3.19, en el cual, de las varias opciones posibles, elegimos precisamen-



Figura 3.19

**Edit** <ENTER> y en el submenú que aparece (figura 3.20), seleccionamos la opción **Part Value (Valor del componente)**, validando con <ENTER>.

A continuación seleccionamos **Name** <ENTER>, mostrándose una **línea de estado** donde aparece: **Value? SW PUSHBUTTON**. Borremos este contenido con la tecla <Borrar atrás> y pulsemos <ENTER> seguido de <ESCAPE> dos veces. Aunque hemos utilizado la opción **Edit**, en parte, más adelante la veremos con más profundidad.

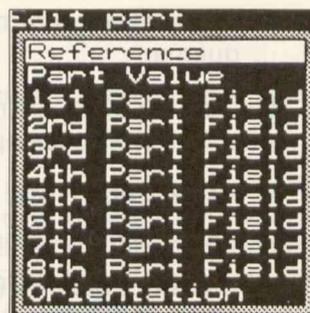


Figura 3.20

- 2.º Situar en el punto de coordenadas **X=8.20 Y=0.20**, un relé tipo "dos posiciones un circuito". Para ello, responder con **RELAY SPDT** a la interrogación de **Get**, situándolo tal y como se muestra el **outline**.
- 3.º Situemos un transistor **NPN** en el punto de coordenadas **X=7.70 Y=1.20**. Responder con **NPN** a la interrogación de **Get**.
- 4.º Colocar cuatro puntos de masa en las siguientes coordenadas:  
**X=7.80, Y=1.80; X=5.60, Y=2.60; X=0.60, Y=4.40; X=5.70, Y=0.90**.  
Responder **GND Power** a la interrogación de **Get?**
- 5.º Coloquemos, por último, el terminal de alimentación **VCC**.

Para **DRAFT**, este terminal se considera como un **objeto** y se encuentra ubicado como opción en el menú de **Place**, y no como componente de cualquier librería.



Figura 21

Seleccionemos la opción **Power** del submenú que aparece cuando pulsamos <ENTER> desde la opción **Place** del menú principal. Aparece un nuevo submenú como el de la figura 3.21. En él se distinguen una serie de opciones que a continuación se detallan.

Las cuatro últimas, son de utilización general y se usan, como ya indicamos anteriormente, en diseños con alta densidad de componentes o bien en diseños ya acabados.

Las que nos interesan a nosotros, son la cuatro primeras.

- **Place** sirve para colocar el **power**, una vez se haya elegido la situación donde queramos ubicarlo.

— **Orientation** determina la **orientación** que va a tener el **power** dentro del esquema.

La figura 3.22 muestra el menú correspondiente cuando se selecciona este subcomando. Son cuatro las opciones posibles:

- **Hacia arriba (Top).**
- **Hacia abajo (Bottom).**
- **Hacia la izquierda (Left).**
- **Hacia la derecha (Right).**

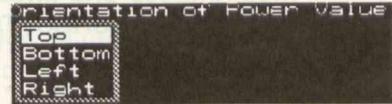


Figura 3.22

— **Value** asigna una denominación al **power**. Por defecto es **VCC**, aunque se puede cambiar por cualquier otra. Para ello basta con responder a la pregunta **Power Value?** con la denominación que deseemos.

— **Type** determina el indicador de **power**. Por defecto es un círculo, pero se puede cambiar por una **flecha (arrow)**, un **poste (bar)** o una **señal (wave)** con tan sólo elegir la opción conveniente del submenú del comando **Type**.

En nuestro ejemplo, bastará situarnos en **X=9.10 Y=0.20** y colocar un **Power** con **orientación hacia la derecha**, de **valor VCC** y de **tipo círculo**.

El resultado obtenido puede verse en la figura 3.23.

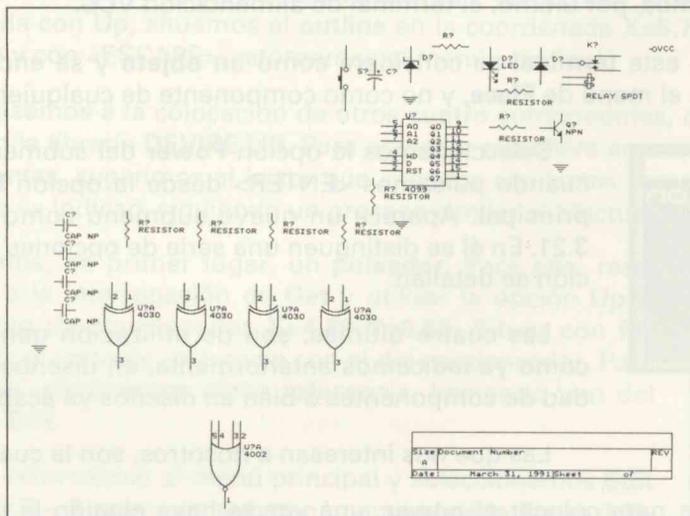


Figura 3.23

## Paso 2. Conexión de componentes

Con este paso vamos a interconectar los diferentes componentes que hemos situado anteriormente.

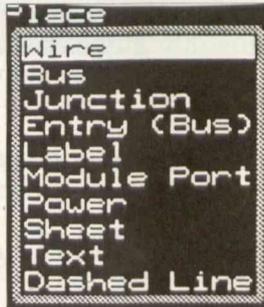


Figura 3.24



Figura 3.25

Para ello, desde el menú principal, invoquemos el comando **Place**. Nada más pulsar <ENTER> se nos mostrará una ventana como la de la figura 3.24. Seleccionemos el **subcomando Wire (Conexión)**. Tras validar con <ENTER> se despliega un nuevo submenú al que llamaremos de **colocación de conexiones**, donde se muestran cinco nuevos subcomandos como los de la figura 3.25.

Vamos a conectar en primer lugar las salidas de las cuatro puertas **EXOR** a las entradas del sumador **4002**. Para ello, desplacémonos hasta el punto de coordenadas **1.80, 4.70** y pulsemos <ENTER>. A continuación, seleccionemos la opción **Begin** seguida de <ENTER>. Descendamos con el cursor hasta el punto de coordenadas **1.80, 5.40**. Pulsemos <ENTER>. Se despliega otro submenú al que se han incorporado dos nuevos subcomandos: **End y New**.

Seleccionemos **New**, seguido de <ENTER> y, a continuación, **Begin** <ENTER>. Conectemos con la patilla 5 del **4002** situada en **(3.30, 5.40)**. Seleccionemos **End** y validemos con <ENTER>. La conexión quedará trazada definitivamente.

Repitamos la operación para las tres salidas de las **EXOR** restantes, uniendo las siguientes coordenadas y ejecutando los subcomandos que de forma resumida las acompañan.

**2.80, 4.70** con **3.20, 5.20**, New, Begin y unir con **3.20, 5.40**. Seleccionar New, nuevamente. **3.80, 4.70** con **3.40, 5.20**, New, Begin y unir con **3.40, 5.40**. Finalizar con New. **4.80, 4.70** con **4.80, 5.40**, New, Begin y unir con **3.50, 5.40**. Finalizar con End.

En segundo lugar, vamos a conectar las patillas del **4099** con las cuatro direcciones (A0..A4) del puerto paralelo, todavía sin **situarse**. Comencemos por conectar la patilla 5. Para ello, tracemos una conexión desde el punto de coordenadas **5.40, 1.30** hasta el punto **1.50, 1.30** utilizando un proceso similar al descrito. A continuación, podríamos repetir el mismo proceso para el conexionado de las siguientes patillas, pero resulta mucho más cómodo efectuar el proceso de forma automática. Para ello, situémonos

en el punto de coordenadas **5.40, 1.40** y retornemos al menú principal donde seleccionaremos la opción **Repeat**. Tras validar con <ENTER>, se trazará de forma automática una conexión paralela a la de la patilla 5. Para conexionar la patilla 7, repetiremos el proceso (utilizar el comando **Again** para evitar el repetir comandos), pero habiéndonos colocado previamente en la coordenada **5.40, 1.50**.

Para el conexionado de la patilla 3 no podremos emplear el método automático, por lo que deberemos colocarnos en la coordenada **5.40, 1.80** y realizar la conexión con la coordenada **1.50, 1.80** de forma normal.

La patilla 9, de salida, (**6.60, 1.30**) del 4030, también podemos conectarla a su resistor respectivo (**6.90, 1.30**), utilizando el mismo proceso.

La conexión de la patilla 4 (**5.40, 1.70**) con la salida del 4002 (**3.30, 6.60**), debido a su largo trazado, la realizaremos en dos pasos:

1.º Comenzando (Begin) en **5.40, 1.70** <ENTER> desplazémonos hasta la coordenada **0.50, 6.60**, pulsemos <ENTER> y seleccionemos la opción **New** seguida de <ENTER>.

2.º Situados en **0.50, 6.60**, seleccionar **Begin** y unir con **3.30,6.60**. Finalizar con **New**.

La conexión de la patilla 2, también la realizaremos en dos partes:

1.ª Situados en **5.40, 1.90**, desplazémonos hasta **5.30, 1.90** y subamos hasta el terminal negativo de **C5 (5.30, 0.80)**. Seleccionar **New** seguido de <ENTER>.

2.ª Situados en **5.30, 1.90**, seleccionemos Begin <ENTER> y unamos el extremo de la resistencia situado en **5.30, 2.00**. Finalizar con **End**.

Las siete patillas de salida restantes, no las conectaremos por el momento.

## Macros

La labor de conexionado de componentes lleva consigo gran cantidad de pulsaciones de la tecla <ENTER> o de los botones del ratón, lo que produce una labor tediosa y repetitiva que repercute en una mayor lentitud en la confección de esquemas.

Para evitar esto, podemos recurrir al comando Macro del menú principal, que nos va a permitir realizar una serie de funciones como las que indica el submenú mostrado en la figura 3.26.

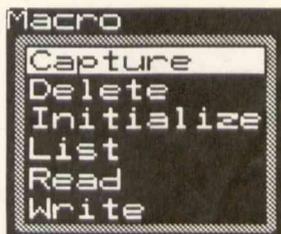


Figura 3.26

**1. Capture.**- Permite **asignar** a una tecla o combinación de teclas, una secuencia de activaciones de comandos que se ejecutarán automáticamente en el momento que sea invocada la tecla o combinación de teclas asignadas.

Al seleccionar este subcomando, aparece el mensaje <Macro>. A partir de este momento, podemos comenzar a introducir la **secuencia** de activaciones de que se compone la macro.

Finalizaremos con la letra <M> si el comando se ejecuta en su totalidad, o bien con la pulsación simultánea de las teclas <Ctrl> + <End>, si el comando no ha finalizado su completa ejecución.

En ambos casos la macro quedará almacenada en memoria, en un lugar preparado a tal efecto como es el **buffer de macros**.

Si queremos salvarla de modo permanente, será necesario utilizar el comando **Write**.

Existirán **macros**, en cuya secuencia de acciones se produce una **pausa** con objeto de entrar o modificar texto a través del teclado. Cada vez que se presente una de estas pausas, hay que introducir la secuencia de teclas <CTRL+HOME> seguida de <ENTER>.

**2. Delete.**- Este subcomando permite **borrar** cualquier macro que hayamos almacenado, ya sea en memoria (**buffer de macros**) o de modo permanente.

Al seleccionarlo, aparece el mensaje **Delete macro?** al que responderemos con el nombre de la tecla o teclas asignadas a la macro. Tras validar con <ENTER>, ésta quedará eliminada.

**3. Initialize.**- Con este subcomando se **inicializan** todas las **macros** que se hallen cargadas en memoria. Al seleccionarlo aparece el mensaje **Erase all macros?** que nos demanda el borrado de las teclas asignadas como macros, existentes hasta el momento.

**4. List.**- Efectúa un listado de las macros que se hallan cargadas en memoria en ese momento.

**5. Read.**- Permite cargar un archivo de macros desde **Draft**. Para ello, basta responder al mensaje **Read all macros from?** con el nombre del archivo donde se encuentran almacenadas las macros que nos interesan.

**6. Write.**- Almacena de modo permanente cualquier **macro**, en un archivo cuyo nombre y extensión hay que especificar ante el mensaje: **Write all macros to?**

**SDT** suministra dos archivos de macros con el programa base que pueden incluirse en la configuración: **MACRO1.MAC** y **MACRO2.MAC**.

### Utilización práctica de macros

Veamos cómo podemos hacer uso de la facilidad que posee **SDT** para elaborar y ejecutar **macros**.

En primer lugar, podemos elaborar una macro que configure el comando **Set** del menú principal, de forma que permita la **reconstrucción de buses (Drag buses)** y mostrar coordenadas (**X,Y Display**). Procesemos el trabajo en los siguientes pasos:

- 1.º Seleccionemos el comando **Macro** del menú principal.
- 2.º Seleccionemos el subcomando **Capture** del submenú que aparece a continuación. Responder con la pulsación simultánea de las teclas **<Alt+S>** a la pregunta **Capture Macro?**
- 3.º Realicemos la siguiente secuencia de acciones:
  - Seleccionar **Set** desde el menú principal.
  - Seleccionar **Drag buses** desde el submenú de **Set** y a continuación, activarlo con **Yes**.
  - Seleccionar **X,Y Display** y activarlo con **Yes**.
- 4.º Finalizar la secuencia introduciendo el carácter **M**.
- 5.º Salvar la macro mediante el comando **Write** en un archivo que llamaremos **MACROPRO.MAC**.

A continuación, vamos a realizar una segunda **macro**, para **colocar conexiones**, a la que asignaremos la tecla **F1**.

Siguiendo un proceso paralelo al anterior, realizaremos la siguiente secuencia de operaciones:

- 1.º Seleccionar **Macro** desde el menú principal.
- 2.º Seleccionar **Capture** y responder con **F1** al interrogante **Capture?**
- 3.º Asignar la siguiente secuencia de actividades:
  - Seleccionar **Place** desde el menú principal.

- Seleccionar **Wire** desde el submenú de **Place**.
  - Seleccionar **Begin** desde el submenú de **Wire**.
- 4.º Finalizar con **<CTRL> + <END>** ya que el comando no se ejecuta totalmente.
  - 5.º Salvar en el archivo **MACROPRO.MAC**.

## Conexión de los resistores

Teniendo en cuenta el apartado anterior, podemos abordar la conexión de los 7 resistores de nuestro esquema, utilizando las dos macros creadas para lo cual será preciso cargar el archivo **MACROPRO.MAC**. No es necesario arrancar de nuevo **Draft**, ya que disponemos del subcomando **Read** que, como vimos anteriormente, carga un archivo de macros sin abandonar **Draft**. Basta responder con **MACROPRO.MAC** al interrogante **Read all macros from?**. Previamente, es conveniente la utilización del subcomando **Initialize**, con objeto de liberar el **buffer de macros**.

**Nota.** Si, desde el menú de configuración de **Draft**, configuramos la opción **Macro File** con alguno de los dos archivos (**MACRO1.MAC** o **MACRO2.MAC**), las teclas **<Alt+S>** y **F1** responderán a funciones diferentes.

Volviendo de nuevo a nuestro esquema, comencemos por la colocación de los cuatro primeros resistores, a los que conectaremos así:

### Extremo superior

(**1.90, 2.50**) con (**1.90, 1.50**) correspondiente a la línea **A2**; (**2.90, 2.50**) con (**2.90, 1.40**) correspondiente a la línea **A1**; (**3.90, 2.50**) con (**3.90, 1.30**) correspondiente a la línea **A0**; (**4.90, 2.50**) con (**4.90, 1.80**) correspondiente a la línea **A3**.

### Extremo inferior

(**1.90, 3.00**) con (**1.90, 3.50**) de la entrada de la **EXOR**.  
 (**2.90, 3.00**) con (**2.90, 3.50**) de la segunda **EXOR**.  
 (**3.90, 3.00**) con (**3.90, 3.50**) de la tercera **EXOR**.  
 (**4.90, 3.00**) con (**4.90, 3.50**) de la cuarta **EXOR**.

A continuación, conectaremos el extremo inferior del resistor conectado a la patilla 2 del **4099**, de coordenadas **5.30, 2.50** con el terminal de masa (**5.70, 2.60**). Notar que es necesario efectuar un ángulo recto.

Conectemos ahora, la base del transistor (**7.60, 1.30**) al terminal libre del resistor de base (**7.40, 1.30**). Por último, conectemos los terminales de los resistores restantes de la siguiente forma:

Para el primero:

- El terminal izquierdo (**7.00, 0.80**), con el cátodo del diodo LED (**6.90, 0.80**). Finalizar con **New**.
- El terminal derecho (**7.50, 0.80**), con el colector del transistor (**7.90, 1.10**). Finalizar con **New**. Notar que hay que trazar un ángulo recto.

Para el segundo:

- El terminal izquierdo (**6.10, 0.20**), con el extremo superior del pulsador (**4.90, 0.20**). Finalizar con **New**.
- El terminal derecho (**6.60, 0.20**) con el positivo VCC (**9.10, 0.20**) de la alimentación.

## Conexión de condensadores

Vamos a comenzar por los cuatro de 1 nF. En primer lugar, conectemos el extremo izquierdo del superior (**0.90, 2.70**) con el terminal de masa. Será necesario ir hasta (**0.70, 2.70**) y a continuación bajar hasta (**0.70, 4.40**). Finalizar con **New**. Para facilitar la tarea, podemos utilizar perfectamente la macro **F1**.

Continuemos conectando componentes. El terminal derecho (**1.20, 2.70**) se conecta al extremo del resistor, en el punto de coordenadas (**1.90, 3.10**). Para ello, desplácese hasta (**1.50, 2.70**), bajemos hasta (**1.50, 3.10**), uniendo a continuación con (**1.90, 3.10**). Utilizar el comando **New**. Para el segundo condensador conectemos el extremo izquierdo (**0.90, 3.10**) con (**0.70, 3.10**). Finalizar con **New**.

El terminal derecho (**1.20, 3.10**) lo conectaremos con el punto **1.40, 3.10**. Para ello, descendamos hasta **1.40, 3.20** y unamos con el punto de coordenadas (**2.90, 3.20**).

Para el tercer y cuarto condensador se procede de forma similar al apartado anterior.

El proceso a seguir es el siguiente:

- Unir la coordenada (**0.90, 3.50**) con la (**0.70, 3.50**).
- Conectar la coordenada (**1.20, 3.50**) con la (**1.40, 3.50**).
- Seleccionar **New**. Subir hasta (**1.40, 3.30**) y unir con (**3.90, 3.30**).

- Unir la coordenada **(0.90, 3.90)** con **(0.70, 3.90)**.
- Conectar la coordenada **(1.20, 3.90)** con la **(1.50, 3.90)**, subimos hasta la **(1.50, 3.40)** y unimos finalmente con **(4.90, 3.40)**.

Por último, conectemos el condensador electrolítico. Para ello, conectemos, siguiendo el proceso antes descrito, el terminal positivo de dicho condensador, cuya coordenada viene determinada por el punto **(5.30, 0.40)**, con la línea de alimentación positiva, en el punto **(9.10, 0.20)**. Notar que es preciso subir previamente hasta **(5.30, 0.10)**. El extremo correspondiente al negativo ya se encuentra conectado.

## Conexión de semiconductores

Comencemos por el transistor. La base ya se encuentra conectada al correspondiente resistor. Para el colector, situémonos en **(7.90, 0.80)** y unamos con el relé en **(7.90, 0.70)**. Por último, conectemos el emisor **(7.90, 1.50)** con masa en el punto **(7.90, 1.80)**.

Pasemos a conectar los tres diodos. En primer lugar, conectemos el ánodo del **1N4148 (7.60, 0.70)** con el punto de coordenadas **(7.60, 0.80)**. El cátodo ya se encuentra conectado a la línea de alimentación positiva en **(7.60, 0.20)**.

El diodo **LED**, ya tiene conectados sus dos extremos: el ánodo en **(6.90, 0.20)** con la línea de alimentación y el cátodo **(6.90, 0.80)** con el extremo izquierdo de sus resistencia de carga en el punto de coordenadas **(7.00, 0.80)**.

Por último, conectemos el diodo **Zener** de la siguiente forma: el ánodo a la línea positiva de alimentación en **X=5.80 Y=0.20**.

El cátodo **(X=5.80 Y=0.70)** con masa en **(Y=5.80 Y=0.90)**.

## Conexiones varias

Con este título, englobamos una serie de conexiones que restan de nuestro esquema y que son las siguientes: **Pulsador**. El extremo superior ya se encuentra conectado a la línea de alimentación en el punto **(4.90, 0.20)** mientras que el extremo inferior **(4.90, 1.10)** lo conectaremos con el punto de coordenadas **(5.30, 1.10)**.

**Relé**. El extremo superior **(7.90, 0.60)** lo conectaremos con la línea de alimentación en el punto **(7.90, 0.20)**. El extremo inferior ya quedó conectado al colector del transistor.

Entradas restantes de las puertas EXOR. Realizaremos las siguientes conexiones:

- (1.70, 3.50) con (1.90, 2.30) pasando por (1.70, 2.30)
- (2.70, 3.50) con (2.90, 2.30) pasando por (2.70, 2.30)
- (3.70, 3.50) con (3.90, 2.30) pasando por (3.70, 2.30)
- (4.70, 3.50) con (4.90, 2.30) pasando por (4.70, 2.30)

El resultado final es el mostrado en la figura 3.27.

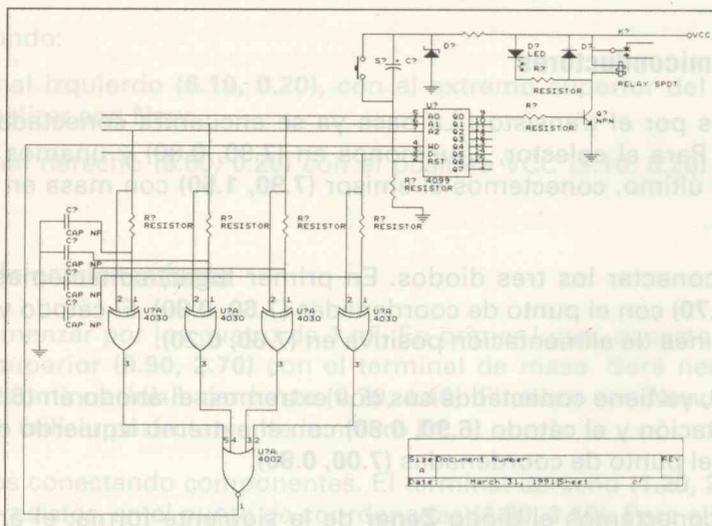


Figura 3.27

### Paso 3. Colocación de uniones

En nuestro programa, existen una serie de líneas que se cruzan perpendicularmente, debiéndose distinguir una conexión de un cruce. Siempre que dos o más conexiones o buses finalicen en un punto común, será necesario establecer una unión. Esto es necesario, además de por su efecto aclarativo, porque DRAFT, posee dos aplicaciones en concreto (NETLIST y ERC) que necesitan de esta información ya que de lo contrario producirían sendos mensajes de error al confundir unión con cruce.

Para colocar una unión en un esquema, seleccionemos la opción Place del menú principal y tras pulsar <ENTER>, elegiremos la opción Junction del submenú. Tras pulsar <ENTER>, aparece el submenú de la figura 3.28.



Figura 3.28

Veamos cómo actúa cada una de ellas.

*Place.* Sitúa la unión en la coordenada elegida previamente. Es la opción más utilizada.

*Find.* Al igual que su homónimo del menú principal, esta opción sirve para buscar cadenas de caracteres en el esquema de trabajo.

Normalmente se utiliza en esquemas densos y mediante su uso, podemos desplazarnos rápidamente adonde deseemos colocar una unión, sin nada más que introducir el nombre, la referencia o cualquier otra cadena de caracteres.

*Jump.* Cuando seleccionamos este subcomando, se nos muestra un nuevo submenú (ver figura 3.29) desde el cual podemos saltar hasta el punto marcado previamente con la opción Tab (hasta un máximo de 8). También podemos desplazarnos hasta cualquier referencia (Reference), hasta cualquier abscisa X o hasta cualquier ordenada Y.



Figura 3.29

*Zoom.* Con este subcomando, podemos utilizar las posibilidades de que muestra el submenú de la figura 3.30: Center, In, Out, Select.

Las estudiaremos en profundidad, más adelante, por lo que sólo haremos mención de la opción Select.



Figura 3.30

Supongamos que necesitamos realizar una unión en un esquema extenso, con gran densidad de componentes. En lugar de desplazarnos con el ratón por toda la hoja de trabajo, podemos visualizar de una forma rápida el lugar donde queremos colocar nuestra unión, si utilizamos el nivel de zoom adecuado. Para ello,

seleccionemos la opción Select y a continuación, el nivel adecuado (2 por ejemplo).

El esquema se mostrará de forma más comprimida, pudiendo localizar el lugar donde se va a colocar la unión, de forma directa.

*Escape.* Con esta opción, abandonamos el submenú de Junction.

Volviendo a nuestro ejemplo, colocaremos las nueve uniones que se adivinan de forma más clara, dejando el resto sin colocar de forma intencionada, con objeto de corregirlas cuando sean detectadas al ejecutar la utilidad NETLIST.EXE.

Veamos cómo se realiza la primera de ellas, quedando el resto como práctica para el lector.

Para ello, situémonos en X=1.90 Y=1.50 y seleccionemos la opción Place del submenú anterior. Presionemos <ENTER> con lo que quedará efectuada la unión entre la línea A2 y el resistor correspondiente.

Para facilitar el trabajo, podemos utilizar la macro asignada a la tecla de función F3 y que se encuentra contenida en el archivo MACRO1.MAC. También podemos optar por crearla e incorporarla a nuestro archivo particular MACROPRO.MAC, para lo cual, habrá que realizar la siguiente secuencia de actividades: Place - Junction - Place - <ESCAPE> y finalizar con M, ya que se ejecuta el comando de forma completa.

Para las siguientes uniones podemos utilizar este marco. A continuación se especifican las coordenadas del resto de las uniones.

Para el resto de líneas de impresora:

(2.90, 1.40) en la línea A1; (3.90, 1.30) en la línea A0 y (4.90, 1.80) en la línea A3.

Para los extremos inferiores de las cuatro resistencias de ataque de las entradas de las EXOR:

(1.90, 3.10), (2.90, 3.20), (3.90, 3.30), (4.90, 3.40).

Para el extremo superior del resistor de RESET: (5.30, 1.90).

El resultado final es como el que muestra la figura 3.31.

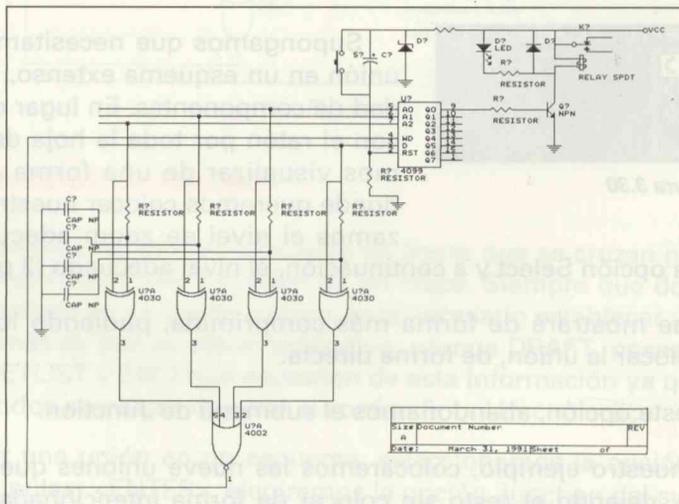


Figura 3.31

Obviamente, existen en el esquema otros puntos donde resulta necesario colocar más uniones. Como ya citamos anteriormente, se han dejado a propósito con objeto de que se produzcan los errores oportunos. Aquellos lectores que deseen completar el esquema, deberán colocar uniones en los siguientes puntos:

- Extremos superiores de los resistores de ataque de las puertas EXOR.
- Patilla izquierda de los 3 últimos condensadores no polarizados.
- Unión del pulsador con el terminal negativo del condensador electrolítico.
- Uniones del condensador electrolítico, los diodos y el relé con la línea de alimentación general.

## Paso 4. Colocación de terminales

DRAFT llama a los terminales, con el nombre de Module Ports y deben colocarse obligatoriamente en los esquemas tipo Jerarquía y en los del tipo Diagrama de bloques.

Como su nombre indica, sirven para especificar nombres de terminales de conexiones o buses.

Para colocar un terminal basta seleccionar el comando Place desde el menú principal de DRAFT y elegir el subcomando Module Port. Tras pulsar <ENTER>, aparece el mensaje "Module Port Name?" que nos pide el nombre con que titularemos el terminal.

En nuestro ejemplo, vamos a colocar cuatro terminales, que titularemos con A0, A1, A2, A3, respectivamente. Corresponden a las cuatro líneas procedentes de la conexión Centronics y por tanto serán terminales de entrada.

Respondamos, pues, con A0 seguido de <ENTER>, al mensaje. Automáticamente se descuelga un menú como el de la figura 3.32 donde se muestran cuatro tipos de terminales.

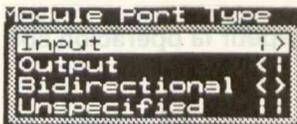


Figura 3.32

*Input.* Se emplea sólo cuando las señales son de entrada. Es el caso de nuestro ejemplo.

*Output.* Se emplea cuando el terminal es usado sólo por una señal de salida.

*Bidirectional* para señales que pueden circular en ambos sentidos.

*Unspecified (Sin especificar)*. Se usa en aquellos casos en que el terminal se emplea para transferir alimentaciones de una hoja de trabajo a otra.

Seleccionemos la opción Input. Tras validar con <ENTER> aparece el outline y su menú de colocación, con las opciones siguientes (ver figura 3.33):



Figura 3.33

*Place*. Sitúa el terminal en el lugar ocupado por el cursor.

*Value*. Es el nombre con el que vamos a designar el terminal. Por tanto, es toda cadena de caracteres que se escriba a continuación de la pregunta "Module Port Name?".

*Type*. Mediante esta opción, podemos seleccionar uno de los cuatro tipos antes mencionados: entrada, salida, bidireccional y sin especificar.

*Style*. Mediante esta opción, podemos seleccionar entre cuatro tipos de indicadores para terminales, según la orientación de la punta de flecha. Estos cuatro indicadores (derecha, izquierda, ambas, sin ningún), responden a la terminología anglosajona siguiente:

*Right pointing*

*Left pointing*

*Both pointing*

*Neither pointing*

El estilo de un terminal es independiente del tipo que sea dicho terminal. Por ejemplo, un terminal del estilo "ambas puntas de flecha" (both pointing) no tiene por qué corresponder necesariamente a un terminal de tipo bidireccional.

Volviendo a nuestra aplicación práctica seleccionemos la opción Input y tras validar con <ENTER>, situemos nuestro terminal A0 en la coordenada X=1.10 Y=1.30 mediante la opción Place seguida de <ENTER>.

Podemos efectuar una macro que nos active automáticamente hasta la aparición del mensaje Module Port Name?, a la que asignaremos la tecla F4. La secuencia de acciones a realizar sería: Place, Module Port, <Ctrl+End>.

Puesto que el resto de terminales son de entrada, podemos repetir la operación con tan sólo cambiar el valor que corresponda a cada terminal.

Notemos que tras colocar cada terminal, se nos pregunta de forma automática por el siguiente.

De esta forma colocaremos:

A1 en el punto X=1.10 Y=1.40

A2 en el punto X=1.10 Y=1.50

A3 en el punto X=1.10 Y=1.80

A continuación, podemos colocar también 6 terminales de salida correspondientes a las salidas 1..6 del C.I 4099. Esto es así, porque de otra forma, quedarían sin conectar, cuando en realidad a ellos van conectados los circuitos de activación de los diferentes relés, que por repetición, no se colocan.

Procediendo de la misma forma que en el caso de los terminales de entrada A0..A3, pero seleccionando la opción Output, colocaremos los siguientes terminales de salida:

RL1 en el punto X=6.60 Y=1.40

RL2 en el punto X=6.60 Y=1.50

RL3 en el punto X=6.60 Y=1.60

RL4 en el punto X=6.60 Y=1.70

RL5 en el punto X=6.60 Y=1.80

RL6 en el punto X=6.60 Y=1.90

Por último, coloquemos dos terminales de salida en los contactos del relé. Los nombraremos como NC y NA que corresponden al contacto "normalmente cerrado" y al contacto "normalmente abierto" y los situaremos en X=8.70 Y=0.30 y X=8.70 Y=0.50 respectivamente.

Con esta operación, queda finalizado el cuarto paso, obteniéndose un esquema como el de la figura 3.34.

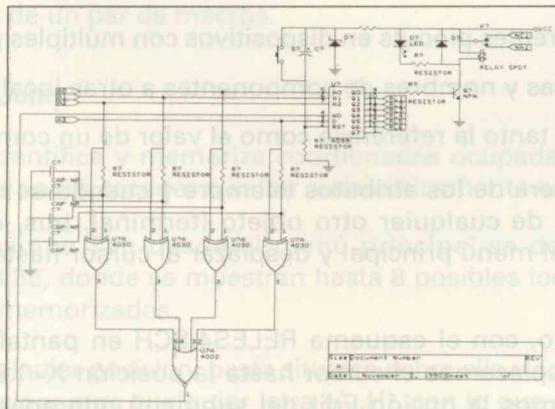


Figura 3.34

## Paso 5. Colocación de texto y líneas auxiliares

Si observamos nuestro esquema, veremos que aparecen una serie de componentes cuyos valores son nombres genéricos, cosa no deseable en ningún esquema. Así, por ejemplo, observaremos que el valor de las resistencias es idéntico y además un texto como RESISTOR. Para los condensadores y transistores ocurre otro tanto (CAP NP y NPN, respectivamente).

Esto podría subsanarse de modo automático, utilizando librerías que incorporasen, además de la referencia del componente, su valor óhmico. De esta forma, en una librería de resistores, por ejemplo, al responder a Get? con 100K, obtendríamos un outline, que al ser emplazado mediante el subcomando Place, incorporaría a su símbolo la referencia R? y el valor 100K, en lugar de RESISTOR.

Más adelante, cuando tratemos el Editor de librerías, veremos cómo es posible la confección de una librería personalizada de este tipo.

### El comando Edit

Aunque anteriormente hemos hecho uso de él, vamos a tratarlo ampliamente en esta sección.

Con el comando Edit del menú principal, podemos efectuar las siguientes acciones:

- Editar terminales, etiquetas, terminales de alimentación, bloques de título (cajones), símbolos en esquemas jerárquicos, así como la referencia, valor y campo de un componente.
- Seleccionar diferentes pinouts en dispositivos con múltiples partes por pastilla.
- Mover referencias y nombres de componentes a otras localizaciones.
- Hacer invisibles tanto la referencia como el valor de un componente.

Para editar cualquiera de los atributos (siempre y cuando se muestren en pantalla) de un componente o de cualquier otro objeto (terminal, bus, etc), basta invocar el comando Edit desde el menú principal y desplazar el cursor hasta el objeto que queremos editar.

En nuestro ejemplo, con el esquema RELES4.SCH en pantalla y con el comando Edit seleccionado, desplazamos el cursor hasta la posición X=1.90 Y=2.70 y pulsemos <ENTER>. Seleccionemos la opción Edit del submenú que aparece. Tras validar con <ENTER>, aparece un nuevo submenú como el de la figura donde podemos optar por



Figura 3.35

editar la referencia (R?), el valor (RESISTOR), una serie de campos de información adicional y la orientación de cualquier componente extraído de una librería, y en caso de tratarse de un dispositivo con varias partes por pastilla, seleccionar qué dispositivo a editar. En nuestro esquema, nos interesa modificar el valor de los componentes pertenecientes a la librería DEVICE.LIB, que como ya apuntamos, tienen valores genéricos. Por tanto, seleccionemos el comando Part Value. Tras validar con <ENTER> aparece un nuevo submenú (ver figura 3.35) con tres opciones:

Name	Location	Visible
------	----------	---------

Name sirve para editar el valor del componente. En nuestro caso, al seleccionar esta opción y validar con <ENTER> aparecerá el mensaje: Value? RESISTOR. Borremos RESISTOR con <Borrar atrás> y tecleemos 4K7 que es el valor que queremos asignar a este resistor. Validemos con <ENTER> con lo que volveremos al submenú de Edit Part. Pulsemos <ESCAPE> y coloquémonos en el punto de coordenadas 2.90, 2.70, correspondiente al segundo resistor.

Pulsar <ENTER>, seleccionar Edit en el submenú que aparece y tras validar con <ENTER> seleccionemos de nuevo la opción Part Value con <ENTER>. Elegir Name y responder con 4K7 tras borrar RESISTOR con <Borrar atrás>. Pulsar <ENTER> para validar y a continuación <ESCAPE>, con lo que nos encontraremos de nuevo en el submenú de editar componente.

Podemos automatizar bastante el proceso, utilizando los comandos Tag (Marca) y Jump (Salto) además de un par de macros.

### Los comandos Tag y Jump

El comando Tag identifica y memoriza coordenadas ocupadas por el cursor en un esquema, a las que se podrá acceder o saltar automáticamente con el comando Jump.

Cuando seleccionamos Tag desde el menú principal se despliega una ventana como la de la figura 3.36, donde se muestran hasta 8 posibles localizaciones (de la A a la H) que pueden ser memorizadas.

Para memorizar cualquier posición, basta situarse sobre ella y activar el comando Tag. A continuación, seleccionamos cualquier letra (A..H), con lo que dicha posición quedará marcada de modo invisible y lista para ser capturada mediante Jump.

*Nota. Las marcas memorizadas en un esquema, no se mantienen cuando éste se salva o se abandona.*

Apliquemos la teoría que acabamos de ver y procedamos a marcar los restantes resistores. Para ello, desde el menú principal, seleccionemos Tag y tras validar con <ENTER>, asignemos las letras siguientes a la posición que ocupan dichos resistores, de acuerdo con la siguiente tabla:



Figura 3.36

MARCAS	POSICIONES	
A	X=3.90	Y=2.70
B	X=4.90	Y=2.70
C	X=7.20	Y=1.30
D	X=5.30	Y=2.20
E	X=7.20	Y=0.80
F	X=6.40	Y=0.20

## Macros de edición

En primer lugar realicemos una macro, que edite el valor de cada componente, que permita rectificar dicho valor y sustituirlo por otro diferente.

Para la realización de cada una de estas macros, es necesario en primer lugar, colocar el cursor sobre cualquier componente ya que, de lo contrario, al invocar por primera vez el comando Edit durante la secuencia, se produciría un mensaje de error. Además, es conveniente que el valor del componente elegido para capturar la macro, sea el de mayor número de caracteres (RESISTOR en nuestro ejemplo) ya que si no se hace así, quedarían caracteres sin borrar al ejecutar la macro.

El proceso podemos esquematizarlo del modo siguiente:

1. Desde el menú principal, seleccionemos el comando Macro.
2. Seleccionemos el subcomando Capture y asignemos la tecla F5 a la macro.
3. Asignemos la siguiente secuencia de operaciones:

Edit, Edit, Part Value, Name, <Borrar Atrás>, <Ctrl+Home>, <ENTER>, <ESCAPE>, <ESCAPE>, M.

Esta secuencia, se puede crear también mediante la utilización de un simple editor de textos capaz de generar código ASCII, mediante un fichero de formato de macro (macro file format), siendo su contenido de la siguiente forma:

```
{F5}=eepn{Borr}{Ctrl Home}{Enter}{Escape}{Escape}{}
```

4. Finalicemos la macro con el carácter de fin de macro "M".

Análogamente, procedamos a la realización de otra macro que sea capaz de desplazar el valor del componente. Esto se consigue con la utilización del subcomando Location.

Asignemos la tecla F6. El proceso a seguir es similar al anterior, diferenciándose tan sólo en la secuencia de operaciones donde cambiaremos Name por Location.

Su contenido, expresado en macro file format es el siguiente:

```
{F6}=eepI{Borr}{Ctrl Home}{Enter}{Escape}{Escape}{}
```

Por último, realicemos una macro para desplazar la referencia del componente, a la que asignaremos la tecla F7. Su contenido en macro file format es el siguiente:

```
{F7}=eerI{Borr}{Ctrl Home}{Enter}{Escape}{Escape}{}
```

## Edición y modificación del valor del componente

Procedamos, por fin a la edición-modificación de los resistores. Para acceder automáticamente a sus respectivas posiciones, hagamos uso del comando Jump.

Cuando seleccionamos este comando, aparece un submenú como el de la figura 3.37, donde se muestran los 4 tipos de salto automático que es capaz de ejecutar Draft:

- 1.º Salto a cualquiera de las posiciones A..H, que hayan sido marcadas previamente con el comando Tag.
- 2.º Salto a una referencia determinada.
- 3.º Salto a cualquier posición a lo largo del eje X.
- 4.º Salto a cualquier posición a lo largo del eje Y.



Figura 3.37

Puesto que ya hemos realizado las marcas previamente y hemos modificado los dos primeros resistores, procederemos de la siguiente forma:

- 1.º Con el esquema que vamos a modificar, en pantalla, seleccionemos el comando Jump y, a continuación, la primera opción del submenú (A Tag), con lo que el cursor se desplazará automáticamente sobre el tercer resistor.
- 2.º Pulsemos F5 para que se ejecute la macro correspondiente y sustituyamos el valor RESISTOR por el de 4K7.
- 3.º Repitamos el proceso para el resto de resistores, marcados con Tag, con C, D, E, F, G, a los que asignaremos, respectivamente, los siguientes valores: 4K7, 10K, 4K7, 500, 680. Pasemos a la edición-modificación de los condensadores. El primero lo realizaremos en modo manual y para el resto, utilizaremos macros.

Comencemos por editar los cerámicos. Para ello, situémonos en X=1.10 Y=2.70 y tras pulsar <ENTER> seleccionar Edit <ENTER>, Part Value <ENTER>, Name <ENTER>. Ante el mensaje Value? CAP NP, borremos mediante la tecla <Borrar Atrás> el valor genérico CAP NP y en su lugar tecleemos "1 nF". Validemos con <ENTER>.

Procedamos ahora a cambiar la localización de la referencia de este condensador.

Procurando que el cursor se halle sobre el condensador, seleccionemos Reference seguido de <ENTER> y a continuación seleccionemos Location. Veremos cómo se iluminan tanto la referencia (C?), como el valor (1 nF), pero tan sólo es trasladable la referencia.

Para el resto de condensadores, el proceso a seguir es análogo al descrito, pero con la utilización de las macros F6 y F7, se automatiza totalmente. Al igual que para los resistores, podemos utilizar los comandos Tag y Jump para saltar automáticamente a las posiciones de los componentes que queremos editar.

Las posiciones de los condensadores restantes, cuyo valor genérico es CAP NP y que queremos cambiar por el de 1 nF, así como las posiciones respectivas de sus nuevas referencias son las siguientes:

#### **ANTIGUAS**

X=1.10 Y=3.10

X=1.10 Y=3.50

X=1.10 Y=3.90

#### **NUEVAS**

X=1.00 Y=3.00

X=1.00 Y=3.40

X=1.00 Y=3.80

Con respecto al condensador electrolítico, vamos a asignarle un valor de 10 uF (en lugar de CACITOR POL) haciendo uso de la macro F5 y a cambiar la situación de dicho valor utilizando la macro F6, hasta colocarlo en X=5.19 Y=0.50.

### Edición de diodos

Comencemos por la colocación del diodo de tipo general. Siguiendo un proceso paralelo al anterior, situémonos en la posición X=7.60 Y=0.50 y activemos F7. Situar en X=7.51 Y=0.30. Con esto hemos variado la referencia D?

A continuación, vamos a asignarle un valor ya que, como en el caso del condensador electrolítico, el valor genérico Diode fue borrado. Activemos la macro F5. Responder con "1N4148" al interrogante Value?

Tras situar este valor, observamos que se superpone al relé, por lo que será preciso trasladarlo. Lo vamos a colocar en la posición X=7.49 Y=0.30 utilizando la macro F6.

El diodo LED, permanecerá sin variación, mientras que el diodo Zener adoptará un valor de 5V1 utilizando la macro F5. Su referencia, podemos trasladarla mediante la macro F7 a la posición X=5.70 Y=0.29.

### Edición del resto de componentes

Transistor. Vamos a cambiar el nombre genérico NPN por el valor BC547B, para lo cual utilizaremos la macro F5, previa colocación en la posición X=7.90 Y=1.30.

Relé. Situémonos en X=8.30 Y=0.60. Mediante la macro F5, cambiemos el valor RELAY SPDT por RELE DIL 12 V.

Pulsador. En primer lugar, desplazemos la referencia hasta X=4.80 Y=0.49 haciendo uso de la macro F7. A continuación, asignemos un valor, que llamaremos PULSADOR mediante la macro F5. Este nombre ha sido elegido lo suficientemente largo con objeto de ver una aplicación del subcomando Visible.

Visible sirve para ocultar el valor del componente. Cuando se selecciona esta opción Draft da a elegir las opciones Yes o No. Yes, que es la opción por defecto hace visible el valor del componente, mientras que No lo hace invisible.

Este subcomando afecta a la pantalla, la hardcopy y todos los niveles de Zoom. En nuestro ejemplo, situémonos sobre el pulsador, seleccionemos Edit desde el menú principal, nuevamente Edit y, seguidamente, seleccionemos Part Value. Tras validar con <ENTER>, elijamos la opción Visible del submenú y a continuación, la opción No, con lo que el valor PULSADOR, quedará oculto.



## Rellenado del cajetín

El cajetín aparece en la parte inferior derecha del esquema. Inicialmente, todos sus campos se encuentran sin rellenar. Para conseguir esto, es necesario colocarse previamente en el interior del cajetín (en nuestro ejemplo, basta colocarse en  $X=7.00$   $Y=6.50$  por ejemplo); a continuación, seleccionar la opción Edit desde el menú principal. Tras validar con <ENTER> aparece un submenú como el de la figura 3.38.

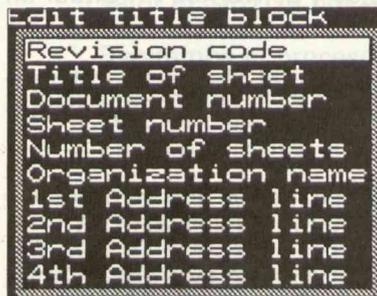


Figura 3.38

Respondamos de la siguiente forma a las opciones que deseemos rellenar:

Con un 1 a Revision Code.

Con TARJETA DE RELES a Title of Sheet.

Con RELES-5 a Document number.

Con 5 a Sheet Number.

Con 8 a Number of Sheets.

Con P.N.T.I.C. a Organization name.

El resto de los campos quedan en blanco. Pulsemos <ESCAPE> para salir.

## Recuadrando partes del esquema

Puede ocurrir que sea conveniente resaltar una zona concreta de un esquema que realiza una función determinada. En nuestro ejemplo, vamos a **recuadrar** con **línea de trazos** la zona correspondiente al transistor y el relé, con su circuitería asociada, titulándola "CIRCUITO DE POTENCIA", para distinguirla del resto del esquema.

Para ello, desde el menú principal, seleccionemos **Place** y tras validar con <ENTER>, seleccionamos **Dashed Line**. Tras pulsar <ENTER>, traslademos el cursor hasta la coordenada  $X=6.70$   $Y=0.10$ . Pulsemos <ENTER>. Aparece un submenú del que seleccionamos **Begin**. Desplacémonos hasta  $X=9.40$   $Y=2.10$  y <ENTER>. Seleccionemos **New** <ENTER>, **Begin** <ENTER> y subamos hasta  $X=9.40$   $Y=0.10$ ; unamos con  $X=6.70$   $Y=0.10$  <ENTER>, **End** y <ENTER>, con lo que quedará efectuado el **recuadro**.

Pasemos a colocar el texto. En cualquier momento, se puede colocar cualquier texto, en cualquier parte de un esquema, seleccionando la opción **Text** del submenú de **Place**. Una vez realizado esto, respondamos con "**CIRCUITO**" al interrogante

**Text?** y situemos el cursor en **X=8.30 Y=1.60**. Pulsemos <ENTER> y situemos con **Place**. Tras pulsar <ENTER> aparece de nuevo el interrogante **Text?**. Respondamos con **"DE"** y llevemos el cursor hasta **X=8.50 Y=1.80** donde situaremos con **Place**.

Repitamos la operación y coloquemos en **X=8.30 Y=2.00**, el texto **"POTENCIA"**. Respondamos al último **Text?** con <ESCAPE>.

## Algunas consideraciones sobre la alimentación

Aunque las consideraciones que a continuación se citan afectan en concreto al esquema que estamos confeccionando, pueden hacerse extensivas a cualquier esquema.

- 1.ª En el esquema práctico que estamos realizando hay dos niveles de alimentación positiva: +12 voltios para los **relés** y +5 voltios para los **circuitos integrados**.
- 2.ª Existen componentes **CMOS** cuyos pines de alimentación están especificados como **Vdd** y **Vss**, diferentes por tanto de los generales **Vcc** y **GND**.

Como consecuencia, debemos especificarlo de la siguiente forma:

1. Conectando **Vcc** a un **power** de **+12 v**.
2. Conectando **Vdd** a un terminal de **+5 v**.
3. Conectando **Vss** a un terminal de **GND**.

Para ello, **editemos Vcc** y sustituyamos el nombre de **VCC** por **+12 V**.

A continuación situemos y **editemos tres power** en los siguientes puntos de coordenadas:

**X=4.40 Y=5.90**, seleccionar **Edit** y sustituir el nombre (**Name**) por **VDD**.

**X=5.10 Y=5.90**, seleccionar **Edit** y sustituir el nombre por **+5V**.

**X=4.40 Y=6.40**, seleccionar **Edit** o la macro correspondiente y sustituir el nombre por **VSS**.

Seguidamente, colocaremos un **GND POWER** en **X=5.00 Y=6.40**.

Conectemos los dos primeros entre sí, así como el tercero con el cuarto.

Salvemos el esquema seleccionando **Quit** desde el menú principal y a continuación, el subcomando **Write to file** al que responderemos con **"RELES-5"**. Si deseamos abandonar **Draft** seleccionemos la opción **Abandon edits**.

El esquema final queda como el de la figura 3.39.

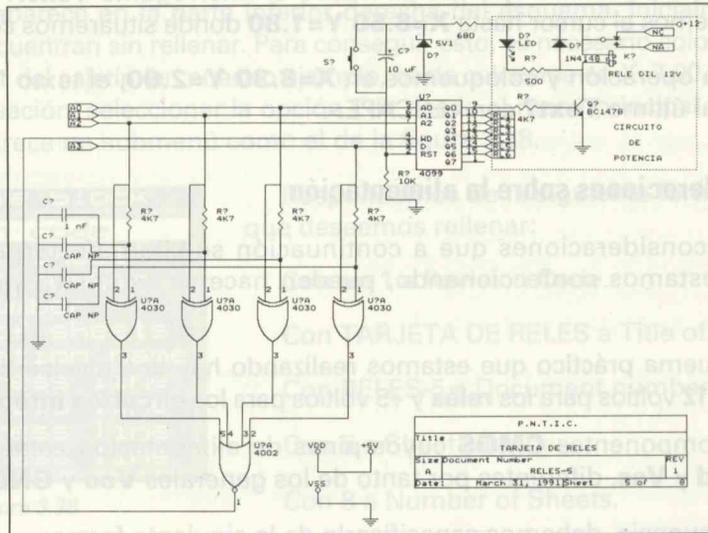


Figura 3.39

## Paso 6. Modificaciones y posibles mejoras

Todo esquema terminado es susceptible de posibles mejoras. En el ejemplo que nos ocupa, quedaba un terminal de salida libre. A él vamos a conectar el **circuito de potencia** conectado a la salida Q1, pero sin necesidad de dibujarlo de nuevo.

También vamos a redibujar las líneas procedentes de la impresora sin perder la conectividad.

Por último, vamos a colocar un **bus** que sustituirá a dichas líneas de entrada.

Las dos primeras acciones a realizar, se llevan a cabo utilizando el comando **Block** y sus subcomandos asociados, mientras que la tercera acción, aunque podía haber sido realizada en el tercer paso, es oportuno incluirla aquí, como ejemplo de mejora.

## Operaciones con bloques

Para **Draft**, un bloque consiste en una porción de la hoja de trabajo, que puede ser tratada como un objeto. Con este comando podemos **mover, reconstruir, copiar, importar** y **exportar** un área de nuestro esquema.

El menú de subcomandos que se pueden utilizar con la opción **Bloque** se obtiene seleccionando la opción **Block** del menú principal. Tras pulsar <ENTER> aparece en pantalla un menú como el de la figura 3.40. Veamos como actúan estos subcomandos.



Figura 3.40

### Subcomandos de bloque

**MOVE.** Este comando sirve para **mover** de lugar un área (o un componente) de nuestra hoja de trabajo y colocarla en otro lugar. El antiguo lugar ocupado quedará en blanco. Para definir el área que se desea mover, una vez seleccionado **Move** <ENTER>, aparece un submenú de 5 opciones, como el de la figura 3.41.



Figura 3.41

Seleccionemos **Begin** y desplacémonos por la pantalla hasta la zona donde se encuentra el conjunto que queremos mover. Tras pulsar <ENTER>, moviendo el cursor (o el ratón) veremos cómo se forma un rectángulo de tamaño variable. **Encerremos** en dicho rectángulo la zona de nuestro circuito que queramos mover y con la opción **End** seguida de <ENTER> finalizamos la operación.

Es conveniente notar cómo cambia la opción **Begin** por **End** nada más validar con <ENTER>. El resto de las opciones de este menú (**Find**, **Jump**, **Zoom**, **Escape**) sirven para facilitar la localización de componentes en esquemas de gran densidad, como ya vimos anteriormente.

Si seleccionamos un área que no contiene absolutamente nada, se obtiene un mensaje que nos indica que **no hay nada para mover**.

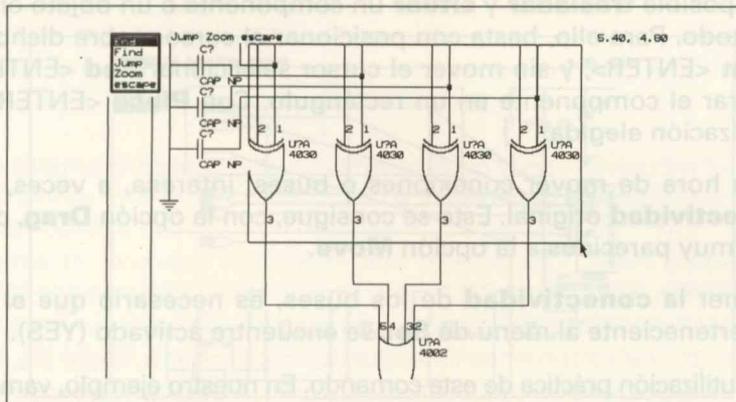


Figura 3.42

Una vez seleccionada el área que vamos a mover, nos aparece un nuevo submenú con las opciones que muestra la figura 3.42. Notemos cómo los componentes encerrados en el área seleccionada se convierten en **outline**, indicándonos que pueden ser desplazados por toda la pantalla.

Cuando nos encontremos en la nueva localización, seleccionaremos la opción **Place**. Tras pulsar <ENTER>, el área seleccionada quedará fijada definitivamente, desapareciendo de su anterior posición. Es interesante observar, que hasta que no se pulse <ENTER>, esta desaparición no tiene lugar, permaneciendo en su lugar original los componentes del área seleccionada.

En el ejemplo mostrado en la figura 3.43, hemos supuesto que trasladamos las cuatro puertas **EXOR** hasta la coordenada **X=9.40, Y=5.10**.

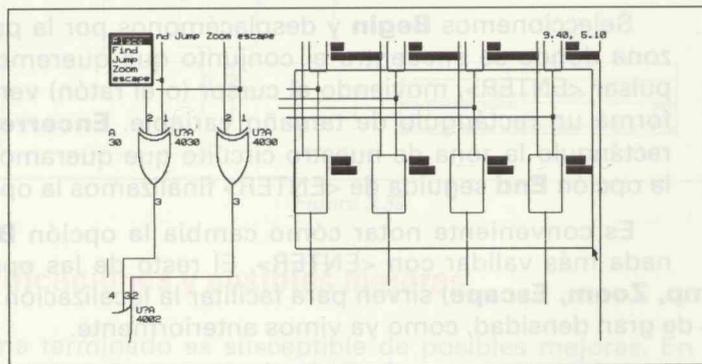


Figura 3.43

También es posible **trasladar** y **situar** un componente o un objeto en general, mediante este método. Para ello, basta con posicionar el cursor sobre dicho objeto seleccionando **Begin** <ENTER>, y sin mover el cursor seleccionar **End** <ENTER>, sin necesidad de encerrar el componente en un rectángulo. Con **Place** <ENTER>, lo colocaremos en la localización elegida.

**DRAG.** A la hora de mover conexiones o buses, interesa, a veces, que éstos no pierdan su **conectividad** original. Esto se consigue, con la opción **Drag**, cuyo empleo y submenús son muy parecidos a la opción **Move**.

Para mantener la **conectividad** de los buses, es necesario que el subcomando **Drag Buses** perteneciente al menú de **Set** se encuentre activado (YES).

Veamos una utilización práctica de este comando. En nuestro ejemplo, vamos a modificar la situación de las cuatro líneas de conexión (A0..A3). El método a seguir es el siguiente:

1. Desde el menú principal, seleccionemos la opción **Block** seguida de <ENTER>. Se abre un submenú en el que seleccionamos la opción **Drag** y validemos con <ENTER>.
2. Seleccionemos el área que queremos trasladar. Para ello, situémonos en la coordenada **(0.90, 1.10)** y a continuación, seleccionemos **Begin** <ENTER>. Desplacémonos hasta el punto **(4.30, 2.00)** y pulsemos <ENTER>. Seleccionemos **End** en el submenú que aparece. Finalizar con <ENTER> (ver figura 3.44).

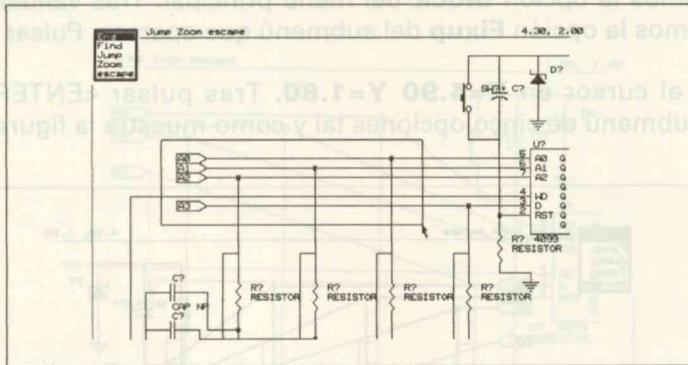


Figura 3.44

3. Traslademos el **outline** hasta la coordenada **(4.30, 1.10)**. Obsérvese cómo se mantienen unidas las conexiones. Validemos con <ENTER>. Situar con **Place** seguido de <ENTER>.

El resultado se observa en la figura 3.45.

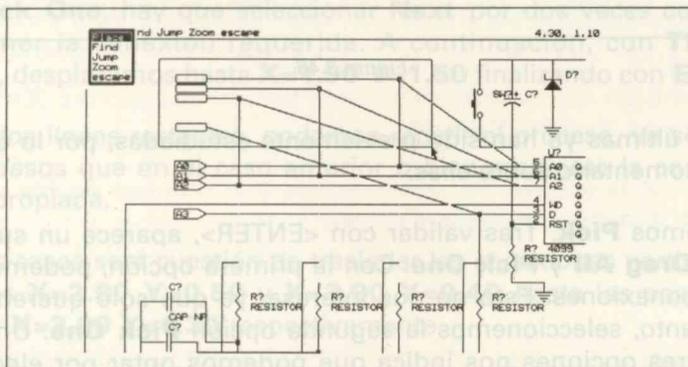


Figura 3.45

< **FIXUP**. Con la opción **Drag**, ya hemos visto como podemos trasladar bloques de esquema, conteniendo conexiones o buses, con la ventaja de mantener la **conectividad**. Pero esta ventaja, no es tal cuando observamos el resultado final.

En efecto; existen cruces de conexiones y sobre todo falta la **ortogonalidad**. Esto se palia con la opción **Fixup**. Vamos a verlo, a continuación, de una forma totalmente práctica, para lo cual sigamos el siguiente proceso:

1. Seleccionemos la opción **Block** del menú principal. Tras validar con <ENTER>, seleccionemos la opción **Fixup** del submenú que aparece. Pulsar <ENTER>.
2. Situemos el cursor en **X=4.90 Y=1.80**. Tras pulsar <ENTER>, aparece un pequeño submenú de cinco opciones tal y como muestra la figura 3.46.

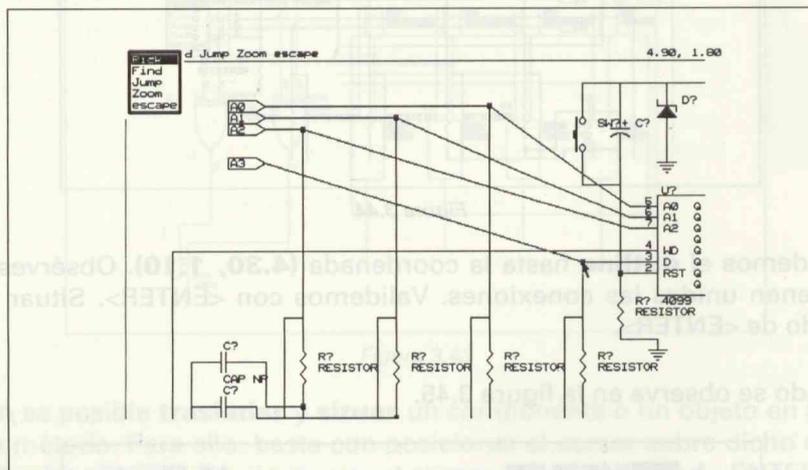


Figura 3.46

Las cuatro últimas ya han sido previamente estudiadas, por lo que omitiremos cualquier comentario sobre ellas.

Seleccionemos **Pick**. Tras validar con <ENTER>, aparece un submenú de dos opciones: **Drag All** y **Pick One**. Con la primera opción, podemos **reconstruir** todas las conexiones. Esto no nos interesa, ya que sólo queremos reconstruir una. Por tanto, seleccionemos la segunda opción **Pick One**. Un pequeño submenú de tres opciones nos indica que podemos optar por elegir la siguiente conexión (**Next**), la anterior (**Previous**) o bien la actual (**This**).

3. Como a priori queda iluminada la conexión que nos interesa, seleccionemos **"This"**. Desplacémonos hasta **X=1.50 Y=1.80** y tras validar con <ENTER>, seleccionemos **End**. Si quisiéramos reconstruir las dos conexiones que forman el ángulo recto, una a una, seleccionaríamos la opción **Drop** <ENTER> seguida de **End** <ENTER>. El resultado final de la operación (ver figura 3.47) es que el terminal **A3** quedará conectado ortogonalmente a la entrada **D** del integrado **4099**.

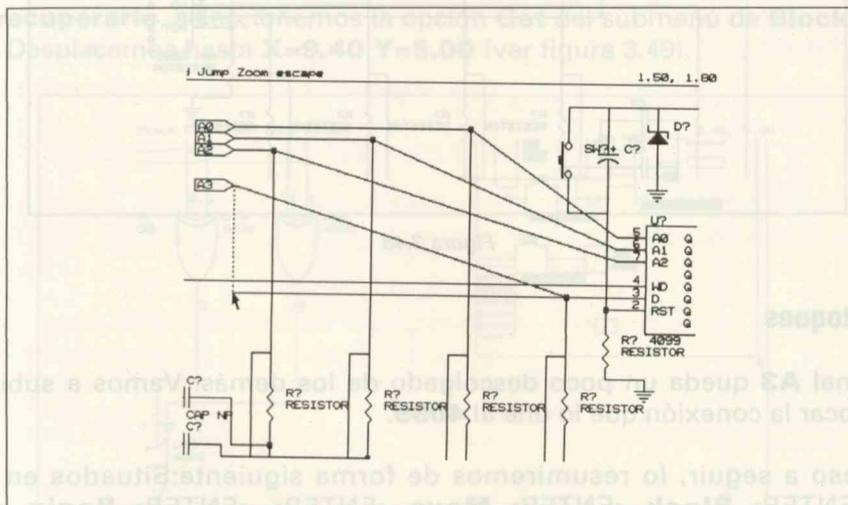


Figura 3.47

4. Situémonos en **X=1.90 Y=0.60** y repitamos el proceso. En este caso, tras seleccionar **Pick One**, hay que seleccionar **Next** por dos veces consecutivas para recomponer la conexión requerida. A continuación, con **This** seguido de <ENTER>, desplazamos hasta **X=1.90 Y=1.50** finalizando con **End**.
5. Para las dos líneas restantes, podemos repetir el proceso, ya sea siguiendo los mismos pasos que en el caso anterior o bien mediante la confección de una **macro** apropiada.

En ambos casos será cuestión de trasladar las conexiones pertinentes desde las posiciones **X=2.90 Y=0.50** y **X=3.90 Y=0.40** hasta las posiciones **X=2.90 Y=1.40** y **X=3.90 Y=1.30** respectivamente.

El aspecto final que presentan ahora las conexiones es como el de la figura 3.48.

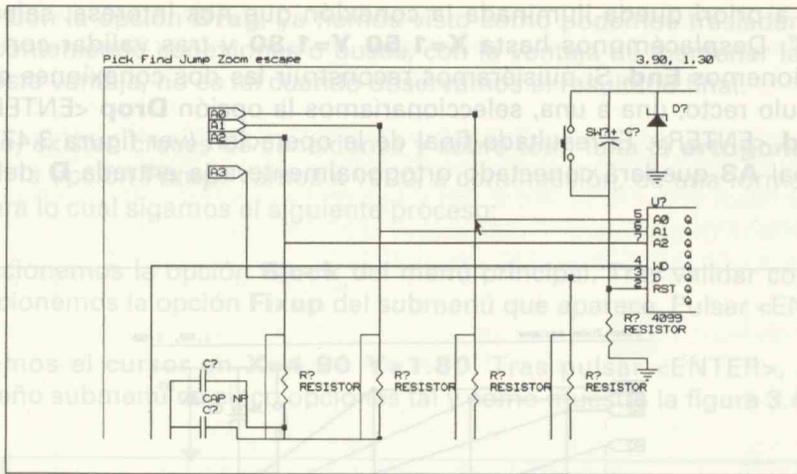


Figura 3.48

### Algunos retoques

El terminal **A3** queda un poco descolgado de los demás. Vamos a subirlo hacia arriba y retocar la conexión que lo une al **4099**.

El proceso a seguir, lo resumiremos de forma siguiente: Situados en **X=1.20 Y=0.90** <ENTER> **Block** <ENTER> **Move** <ENTER> <ENTER> **Begin** <ENTER> <ENTER> **End** <ENTER> y desplacémonos hasta **X=1.20 Y=0.70** <ENTER> **Place** <ENTER>.

A continuación, borremos con el comando **Delete**, la conexión que une este **terminal** con el **4099**.

Borremos también las uniones situadas en las posiciones **X=1.90 Y=0.60**, **X=2.90 Y=0.50** y en **X=3.90 Y=0.40**.

Haciendo uso de la **macro F3**, coloquemos tres uniones en los puntos **X=1.90 Y=1.50**, **X=2.90 Y=1.40** y **X=3.90 Y=1.30**.

### Comandos Save y Get

Recordemos que el terminal de salida **Q7** del **4099** quedó desconectado adrede. Vamos a conectarlo a un **circuito de potencia** idéntico al conectado a la salida **Q0**.

Vamos a procurar no tener que dibujarlo de nuevo, para lo cual habrá que **copiarlo** del circuito conectado a **Q0**.

Para ello, seleccionemos la opción **Block** desde el menú principal y, a continuación, el subcomando **Save**. Tras pulsar <ENTER>, desplazemos el cursor hasta la posición **X=6.60 Y=0.10**, seleccionemos **Begin** y finalizemos con **End** en la posición **X=9.40 Y=2.10**. Con esta acción conseguimos que el trozo de circuito que ha sido recuadrado, quede almacenado en memoria.

Para **recuperarlo**, seleccionemos la opción **Get** del submenú de **Block**. Aparece el **outline**. Desplacemos hasta **X=9.40 Y=5.00** (ver figura 3.49).

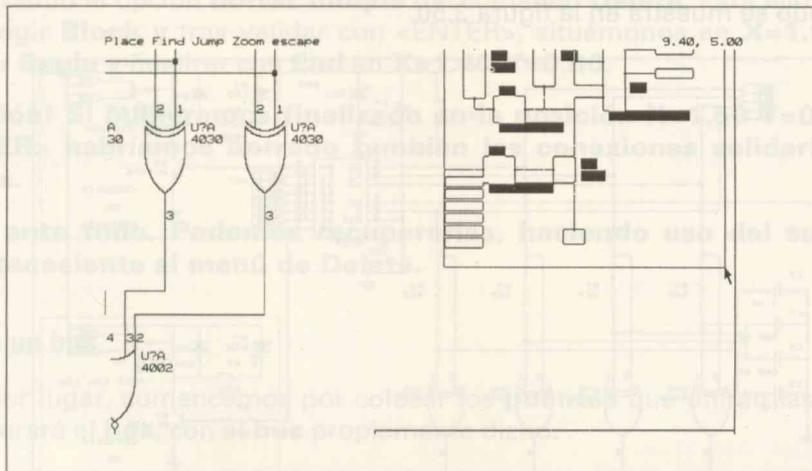


Figura 3.49

**Situemos con Place** y salgamos con <ESCAPE>. De esta forma queda **copiado** el **circuito de potencia** pero con **objetos** que no nos interesan.

## Retoques finales

En efecto; tanto los terminales **RL1..RL6** como el **power Vcc**, son **objetos** que debemos borrar con **Delete**. También debemos borrar la conexión que une los diodos y el relé.

A continuación traslademos el valor **RELE DIL 12V** hasta la posición **X=8.20 Y=0.19**. Hagamos uso de la **macro F6**.

Haciendo uso de la **macro F1**, tracemos un par de conexiones: la primera, de forma que una el punto de coordenadas **X=6.90 Y=3.10** con **X=9.10 Y=0.20** y la segunda, de forma que una la posición **X=6.60 Y=2.00** con la posición **X=6.60 Y=4.20**.

Por último procedamos a rellenar el **cajetín**. Puesto que estamos trabajando con un esquema anterior, habrá que modificar tan solo dos campos: **Document number**, que rellenaremos con **RELES-6** y **Sheet number** que rellenaremos con el dígito **6**.

Salvemos el esquema así obtenido con la opción **Write to file** de **Quit**, con el nombre de **RELE6**.

El resultado se muestra en la figura 3.50.

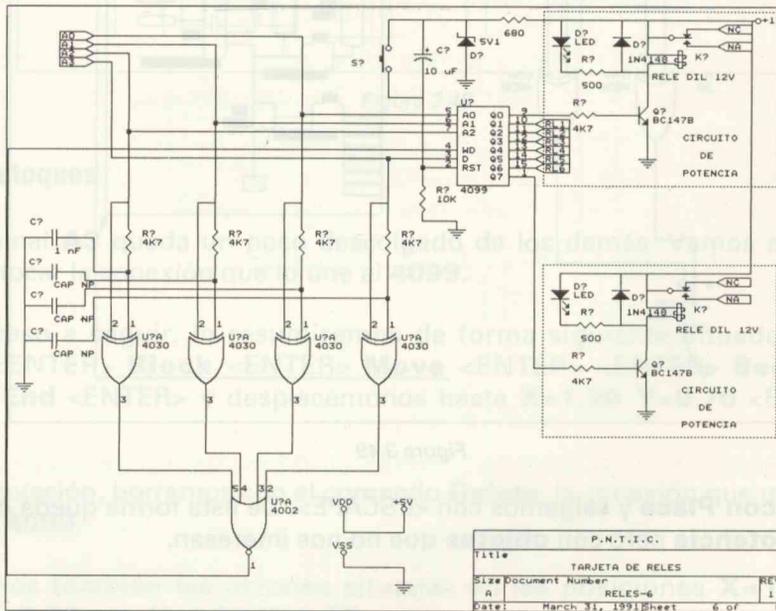


Figura 3.50

## Modificaciones finales

En la mayoría de esquemas electrónicos que incorporan circuitería digital, así como entornos de desarrollo con microprocesadores, es habitual la utilización de un tipo de conexión especial llamada comunmente **Bus**.

**OrCAD/SDT III** trata este tipo de conexión, de un modo especial. De hecho, existen dos subcomandos pertenecientes al menú de **Place** relacionados directamente con **Bus**:

**Bus** propiamente dicho.

**Bus (Entry)**, relacionado con las entradas.

Siguiendo la metodología empleada hasta el momento, veamos como sustituímos los cuatro terminales de entrada **A0..A3** por un **Bus**.

Comenzemos por borrar dichos terminales. Podemos borrarlos de forma individual o bien utilizando la opción **borrar bloque** del comando **Delete**. Para ello, seleccionar **Delete**, elegir **Block** y tras validar con <ENTER>, situémonos en **X=1.00 Y=0.30**. Seleccionar **Begin** y finalizar con **End** en **X=1.40 Y=0.80**.

**¡Atención! Si hubiéramos finalizado en la posición X=1.50 Y=0.80, al pulsar <ENTER> habríamos borrado también las conexiones solidarias con los terminales.**

**Calma ante todo. Podemos recuperarlas, haciendo uso del subcomando Undo perteneciente al menú de Delete.**

### Colocando un bus

En primer lugar, comencemos por colocar los **puentes** que unirán las conexiones que incorporará el **bus**, con el **bus** propiamente dicho.

Para ello, desde el menú de **Place** seleccionemos el subcomando **Entry (Bus)**. Aparece un menú, tipo barra, en la parte superior de la pantalla, que se despliega en una ventana con las opciones de la figura 3.51, tras pulsar <ENTER>.



Figura 3.51

Entre las opciones disponibles en este menú, seleccionemos **"/**. Desplacémonos con el cursor hasta la posición **X=1.50 Y=0.40**, donde situaremos con la opción **Place**. Repitamos la operación para las tres líneas restantes.

Pasemos seguidamente a la colocación del **Bus**. Para simplificar el proceso, podemos confeccionar una **macro**. Asignemos la tecla **F2**. De forma análoga a **F1**, la secuencia de actividades de la **macro** es la siguiente: **Place, Bus, Begin, <Ctrl End>** o, en formato de **macro-archivo: {F2}=pbb {Ctrl+End}{}**.

Situados en **X=1.40 Y=0.50**, pulsemos **F2**, descendamos hasta **X=1.40 Y=1.50** y a continuación hacia la izquierda hasta **X=0.20 Y=1.50**. Pulsemos de nuevo **F2**, subamos y giremos a la derecha hasta la posición **X=2.00 Y=0.20**. El **bus** se mostrará con trazo grueso y de color azul (según configuración).

Quizás pueda parecer un poco **retorcido** ya que forma casi un recuadro. Esto es así, porque de otra forma no cabría el **terminal** de entrada que pondremos a continuación.

## Especificación de un bus

Un **bus** va indefectiblemente ligado a dos o más conexiones que cuando se conectan a éste pasan a formar una única conexión. Para distinguir esta **conexión especial** de cualquier otro tipo, además de utilizar un color diferente y trazo más grueso, hay que especificar tres cosas:

- 1.ª Nombre de cada una de las conexiones que entran/salen del bus.
- 2.ª Nombre propio del bus especificando el número de conexiones que incorpora.
- 3.ª Nombre especial del terminal de entrada/salida, si lo hubiere.

Resumamos estas especificaciones en los tres apartados siguientes:

### Primer paso

En nuestro esquema, el **bus** que pretendemos situar, finaliza en **cuatro** salidas que son las líneas **A0..A3**. Para **nombrar** cada una de estas conexiones, seleccionemos el subcomando **Label** del menú **Place**. Cuando se selecciona este subcomando, aparece el interrogante **Label?** demandando el nombre de la etiqueta que deseamos colocar. Respondamos con **"A0"** (es conveniente usar mayúsculas).

Tras validar con <ENTER> aparece un menú en la parte superior de la pantalla, que llamaremos **menú de colocación de etiqueta**, donde podemos **colocar (Place)**, **orientar (Orientation)**, **cambiar de valor (Value)**, ajustar el **tamaño de letra (Larger o Smaller)**, **encontrar (Find)**, **saltar automáticamente (Jump)**, efectuar un **Zoom** o **escapar**. Desplazemos el cursor hasta la posición **X=1.50 Y=0.40** y situemos la etiqueta **A0** con **Place**.



Figura 3.52

Automáticamente aparecerá de nuevo el interrogante **Label?** pidiendo la segunda etiqueta. Respondamos con **A1** seguido de <ENTER> y coloquemos dicha etiqueta en la posición **X=1.50 Y=0.50** con **Place**.

Repitamos el proceso para las etiquetas **A2** y **A3** a las que situaremos en **X=1.50 Y=0.60** y **X=1.50 Y=0.70** respectivamente. Ante una nueva interrogación de **Label?**, respondamos con **Escape**.

### Segundo paso. Especificación del nombre del bus

El formato general es el siguiente:

**nombre del bus [0.. último miembro del bus].**

En nuestro caso, seleccionemos nuevamente el subcomando **Label** desde el menú de **Place** y respondamos al interrogante **Label?** con **A[0..3]**

**A** equivale a la **Conexión Centronic** mientras que **[0..3]**, indica que el **bus** se compone de cuatro miembros.

A continuación, validemos con <ENTER> y situemos dicha **etiqueta**, mediante el comando **Place**, en el punto de coordenadas **X=0.40 Y=1.50**. **Es conveniente colocar esta etiqueta lo más cerca posible de la línea del bus, tanto si aquella se coloca de forma horizontal como vertical.**

### Tercer paso

Coloquemos, por último, un **terminal, de entrada** en este caso, de tal forma que el **bus** quede especificado definitivamente.

Utilizemos para ello la **macro F4** y respondamos al interrogante **Module Port?** con **A[0..3]**.

Como podemos observar, dentro del corchete hay que especificar el **número** del primer miembro del bus, seguido de dos puntos, y por último, especificar el **número** del último miembro.

Situemos este **terminal** con el comando **Place** en el punto de unión con el **Bus**.

Puesto que lo único que hemos reformado con respecto al esquema anterior, ha sido la sustitución de los **terminales A0..A3** por un bus, cambiemos en el **cajetín** tan sólo el campo correspondiente a **Document number**, donde pondremos **RELES-61**.

El resultado final es el mostrado en la figura 3.53.

También podíamos haber colocado el **bus**, sin necesidad de borrar los **terminales de entrada** tal y como muestra la figura 3.54.

Obsérvense las uniones presentes en el **bus** así como las etiquetas.

## Construcción de un esquema jerárquico

Hasta ahora hemos considerado un esquema **simple** o de **una sola hoja**. Esto es válido para esquemas con baja densidad de componentes pero resulta bastante engorroso a la hora de tratar con esquemas que incorporan muchos componentes.

Como ejemplo de esquema jerárquico, vamos a construir uno de **dos niveles** cuyas especificaciones son las siguientes:

- **Esquema raíz.** Está compuesto por dos **módulos** titulados **Etapa de control** y **Etapa de potencia**. Además forman parte de este esquema los correspondientes **buses** así como los **terminales** pertinentes. El nombre de este esquema es **RAIZ.SCH**.
- **Primer Nivel de jerarquía.** El módulo **Etapa de potencia** está compuesto a su vez por **ocho** módulos (rele0.. rele7), cada uno de los cuales incorpora un **bus** de entrada y **dos buses** de salida. El módulo **Etapa de control** no tiene ningún nivel descendente de jerarquía. Este primer nivel está formado por dos esquemas titulados **POTEN.SCH** y **CONTROL.SCH**.
- **Segundo nivel de jerarquía.** Cada uno de los **ocho** módulos que integran la **Etapa de potencia** queda resumido en un esquema **simple** formado por el **circuito de activación del relé**. El esquema correspondiente a este nivel de jerarquía lleva por título **RELE.SCH**.

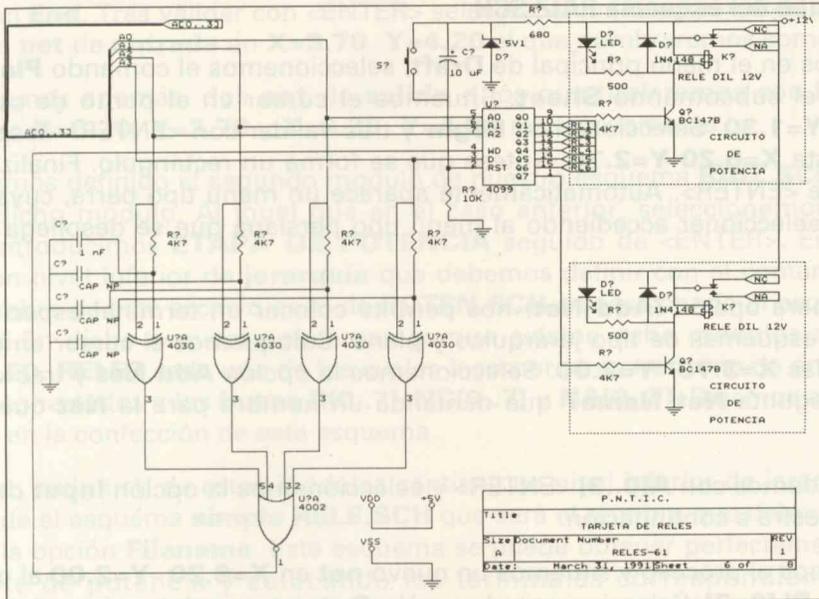


Figura 3.53

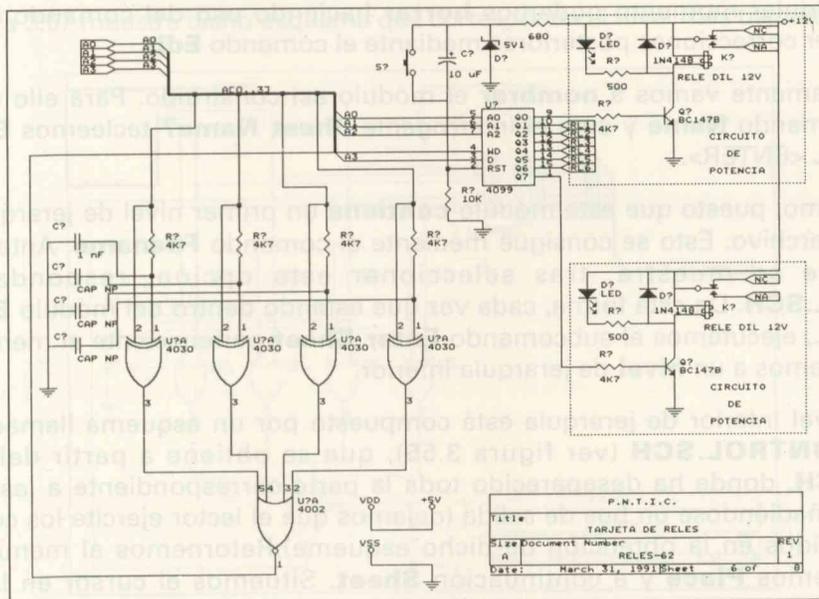


Figura 3.54

## Construcción del esquema RAIZ.SCH

Situados en el menú principal de **Draft**, seleccionemos el comando **Place** y a continuación el subcomando **Sheet**. Situemos el cursor en el punto de coordenadas **X=3.70 Y=1.30**. Seleccionemos **Begin** y tras validar con <ENTER> desplacemos el cursor hasta **X=6.20 Y=2.70**. Nótese que se forma un rectángulo. Finalizar con **End** seguido de <ENTER>. Automáticamente aparece un menú tipo barra, cuyas opciones podemos seleccionar accediendo al menú tipo persiana que se despliega tras pulsar <ENTER>.

La primera opción (**Add Net**) nos permite colocar un terminal especial, llamado **Net**, para esquemas de tipo jerárquico y plano. Coloquemos el cursor en el punto de coordenadas **X=3.70 Y=2.00**. Seleccionemos la opción **Add Net** y tras validar aparece la pregunta **Net Name?** que demanda un nombre para la **Net** que queremos añadir.

Respondamos con **A[0..3]** <ENTER> y seleccionemos la opción **Input** del submenú que se muestra a continuación.

Repitamos el proceso y situemos un nuevo **net** en **X=6.20 Y=2.00** al que titularemos como **RL[0..7]**. Seleccionemos la opción **Output** para determinar el tipo de **net**.

En cualquier momento podemos **borrar** haciendo uso del comando **Delete** así como hacer correcciones posteriores mediante el comando **Edit**.

Seguidamente vamos a **nombrar** el módulo así construido. Para ello seleccionemos el comando **Name** y ante el interrogante **Sheet Name?** tecleemos **ETAPA DE CONTROL** <ENTER>.

Por último, puesto que este módulo **contiene** un primer nivel de jerarquía asignémosle un archivo. Esto se consigue mediante el comando **Filename**. Ante el interrogante que se muestra, tras seleccionar esta opción, respondamos con **CONTROL.SCH**. De esta forma, cada vez que estando dentro del módulo **ETAPA DE CONTROL**, ejecutemos el subcomando **Enter Sheet** perteneciente al menú de **Quit**, descenderemos a un **nivel** de jerarquía inferior.

Este nivel inferior de jerarquía está compuesto por un esquema llamado precisamente **CONTROL.SCH** (ver figura 3.55), que se obtiene a partir del esquema **RELE7.SCH**, donde ha desaparecido toda la parte correspondiente a las etapas de potencia añadiéndose un bus de salida (dejamos que el lector ejercite los conocimientos adquiridos en la obtención de dicho esquema). Retornemos al menú principal. Seleccionemos **Place** y a continuación **Sheet**. Situemos el cursor en la posición **X=3.70 Y=3.40** y seleccionemos **Begin**. Desplacemos hasta **X=6.20 Y=5.00** y fina-

licemos con **End**. Tras validar con <ENTER> seleccionemos la opción **Add Net** y coloquemos un **net** de **entrada** en **X=3.70 Y=4.20** al que nombraremos como **R[0..7]**.

Coloquemos además, dos **net** de **salida** a los que titularemos con **NC[0..7]** y **NA[0..7]** en **X=6.20 Y=3.70** y **X=6.20 Y=4.60** respectivamente.

Ya tenemos definido el segundo módulo de nuestro esquema **RAIZ.SCH**. Vamos a nombrar dicho módulo. Al igual que en el caso anterior, seleccionemos la opción **Name** e introducimos **ETAPA DE POTENCIA** seguido de <ENTER>. Este módulo contiene un nivel inferior de **jerarquía** que debemos definir con el comando **Filename**. El nombre de este archivo es el de **POTEN.SCH** y corresponde al esquema de la figura 3.56. En dicho esquema observamos que existen ocho módulos nombrados como **RELE0..RELE7** cada uno de los cuales incorpora tres **net** (uno de entrada y dos de salida) conectados a los **buses R[0..7]**, **NC[0..7]** y **NA[0..7]**. Dejamos al lector que se ejercite en la confección de este esquema.

A su vez, cada uno de estos módulos contiene un nivel inferior de jerarquía al que corresponde el esquema **simple RELE.SCH** que será el nombre que deberemos introducir con la opción **Filename**. Este esquema se puede obtener perfectamente, aislando la parte de potencia y colocando los terminales correspondientes, desde **RELES7.SCH**.

La figura 3.57 muestra dicho esquema que corresponde al nivel más bajo de **jerarquía**.

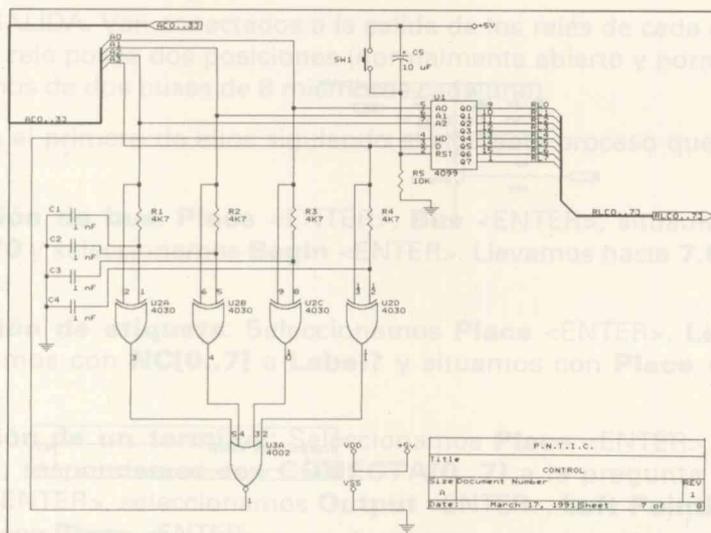


Figura 3.55

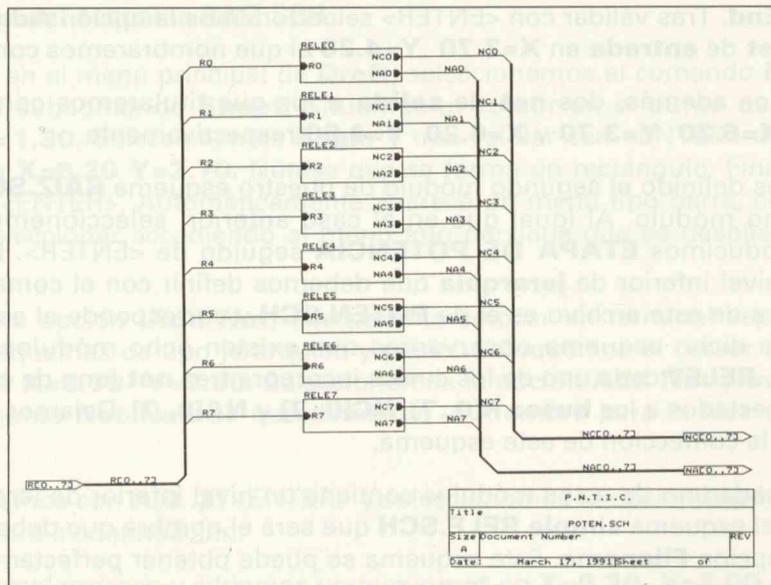


Figura 3.56

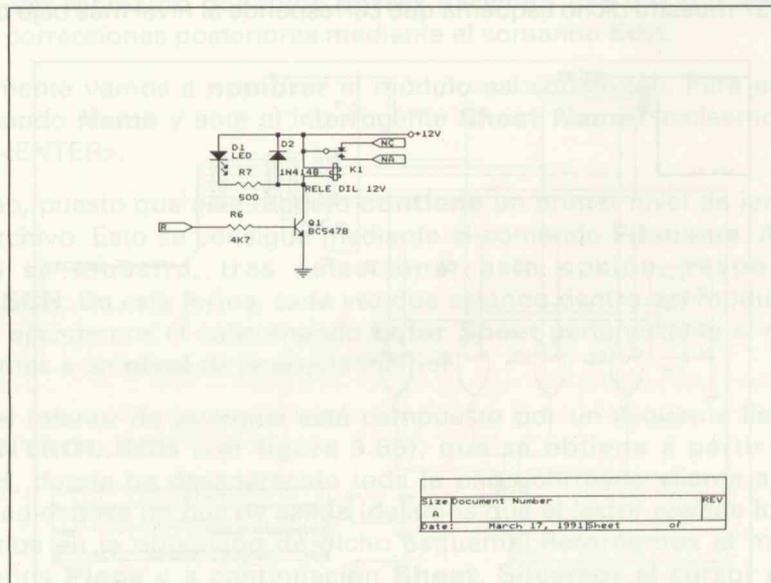


Figura 3.57

## Conexión de módulos

Continuando con la confección del esquema **RAIZ.SCH**, pasemos a la conexión de los módulos. Para ello, colocaremos tres **buses**: uno de entrada, uno de conexión y otro de salida.

**BUS DE ENTRADA.** Desde el menú principal seleccionemos **Place** y a continuación **Bus**. Situemos el cursor en **X=3.70 Y=2.00** y con **Begin** traslademos hasta **X=1.70 Y=2.00** y finalicemos con **End**.

Pasemos a **etiquetarlo** para lo cual desde **Place** seleccionamos **Label** y respondemos con **A[0..3]** y situemos con **Place** en **X= 2.30 Y=2.00**.

Por último, coloquemos un **Module Port** de nombre **CENTRO[0..3]**, de **entrada** y en el punto de coordenadas **X=0.60 Y=2.00**.

**BUS DE CONEXION.** Siguiendo un proceso similar al anterior (podemos efectuar una macro), seleccionemos **Place, Bus**, situemos el cursor en **X=6.20 Y=2.00**, seleccionemos **Begin**, desplazemos hasta **X=6.90 Y=2.00** y descendamos hasta **X=6.90 Y=3.00**. Seleccionemos **New, Begin** y desplazemos hasta **X=3.00 Y=3.00**, descendiendo a continuación hasta **X=3.00 Y=4.20**. De nuevo seleccionemos **Begin** y desplazemos hasta **X=3.70 Y=4.20**, finalizando con **End**.

Etiquetaremos dicho **Bus** con **R[0..7]** que colocaremos en **X=4.60 Y=3.00** con **Place**.

**BUSES DE SALIDA.** Van conectados a la salida de los relés de cada etapa de potencia. Como cada relé posee dos posiciones (normalmente abierto y normalmente cerrado) dispondremos de dos buses de 8 miembros cada uno.

Coloquemos el primero de ellos siguiendo el siguiente proceso que resumimos en tres pasos:

- 1. Colocación de bus:** **Place** <ENTER>, **Bus** <ENTER>, situamos el cursor en **6.20, 3.70** y seleccionamos **Begin** <ENTER>. Llevamos hasta **7.60, 3.70** y **End** <ENTER>.
- 2. Colocación de etiqueta:** Seleccionamos **Place** <ENTER>, **Label** <ENTER>, respondemos con **NC[0..7]** a **Label?** y situamos con **Place** en **6.60, 3.70** <ENTER>.
- 3. Colocación de un terminal:** Seleccionamos **Place** <ENTER>, **Module Port** <ENTER>, respondemos con **CONECTA[0..7]** a la pregunta **Module Port Name?** <ENTER>, seleccionamos **Output** <ENTER>, **Left Pointing** <ENTER> y situamos con **Place** <ENTER>.

De la misma forma, conectemos un **bus** al **net** de salida **NA[0..7]** al que etiquetaremos con **NA[0..7]** y conectaremos un **Module Port** de nombre **ABIERTO[0..7]**.

Por último, rellenemos el cajetín y salvemos con el nombre de **RAIZ.SCH**.

El resultado final es el esquema de la figura 3.58.

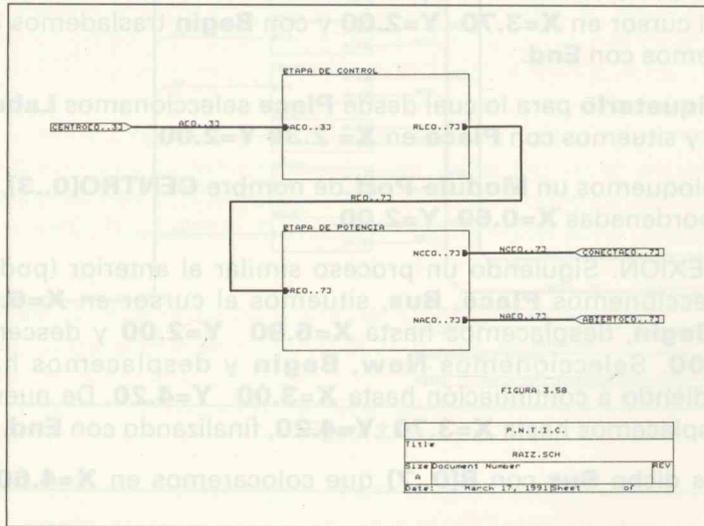


Figura 3.58

## Aplicación práctica de modificación de componentes

Tanto la modificación de componentes (personalización) como la incorporación de otros nuevos, se realiza con la utilidad **Libedit**, de funcionamiento totalmente similar a **Draft**.

Al levantar dicha utilidad aparece una pantalla como la mostrada en la figura 4.1.



Figura 4.1

A la hora de simular el circuito que nos viene sirviendo de ejemplo, con el módulo **VST**, nos vamos a encontrar con la sorpresa de que no existe el componente al 4099. Existen dos soluciones posibles: modificarlo desde **Libedit** o ir un equivalente desde **SOT**. Opción que vamos a trabajar con **SOT**.

## Capítulo 4 El editor de componentes

Hojando un Data Book, encontramos que el **6N74HC259** de TTL, realiza las mismas funciones, pero es necesario modificar la situación de alguno de sus pines con objeto de modificar lo menos posible las conexiones del esquema.

La figura 4.2 muestra las diferencias de patillaje entre los dos circuitos integrados. Las salidas **Q0...Q7** son idénticas, mientras que las entradas **A0...A2**, son equivalentes a las **B0...B2**, cambiadas de posición.



## Aplicación práctica de modificación de componentes

Tanto la modificación de componentes (personalización), como la incorporación de otros nuevos, se realiza con la utilidad **Libedit**, de funcionamiento totalmente similar a **Draft**.

Al invocar dicha utilidad aparece una pantalla como la mostrada en la figura 4.1.

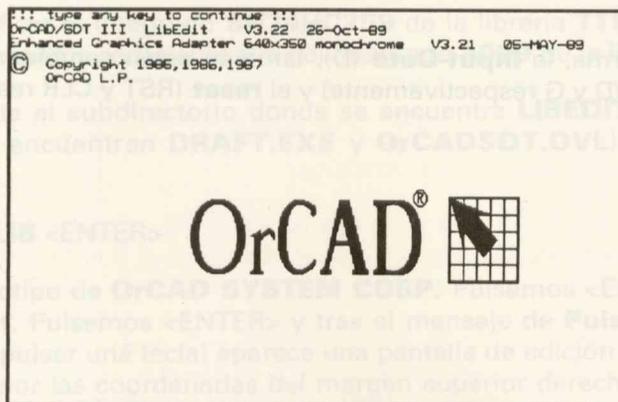


Figura 4.1

A la hora de simular el circuito que nos viene sirviendo de ejemplo, con el módulo **VST**, nos vamos a encontrar con la sorpresa de que no existe el modelo correspondiente al **4099**. Existen dos soluciones posibles: modelizarlo desde **VST** o bien, utilizar un **equivalente** desde **SDT**. Optemos por la segunda opción, ya que nos encontramos trabajando con **SDT**.

Hojeando un Data Book, encontramos que el **SN74HC259** de TTL, realiza las mismas funciones, pero es necesario modificar la situación de alguno de sus **pins** con objeto de modificar lo menos posible las conexiones del esquema.

La figura 4.2 muestra las diferencias de patillaje entre los dos circuitos integrados. Las salidas **Q0..Q7** son idénticas, mientras que las entradas **A0..A2**, son equivalentes a las **S0..S2**, cambiadas de posición.

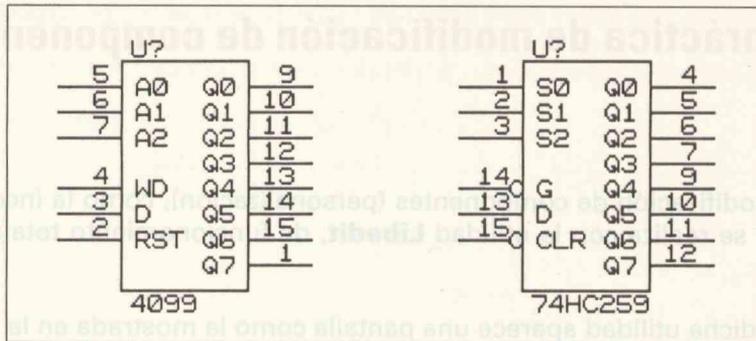


Figura 4.2

De la misma forma, la **Input-Date** (D), también están cambiadas de posición, así como el **enable** (WD y G respectivamente) y el **reset** (RST y CLR respectivamente).



Figura 4.1

## Una sesión con LIBEDIT

**LIBEDIT.EXE** constituye por sí misma una particular utilidad de gran provecho a la hora de **modificar** o **crear** componentes.

La configuración de **LIBEDIT** se realiza de modo similar a la de **DRAFT**, con **LIBEDIT/C**.

Veamos como funciona esta utilidad en un caso práctico.

Vamos a modificar el integrado **SN74HC259** de la librería **TTL**, para obtener una configuración de patillaje semejante a la del integrado **4099** de la librería **CMOS**.

Para ello, desde el subdirectorio donde se encuentra **LIBEDIT.EXE** (debe ser el mismo donde se encuentran **DRAFT.EXE** y **OrCADSDT.OVL**), introduzcamos la siguiente orden:

```
LIBEDIT TTL.LIB <ENTER>
```

Aparece el logotipo de **OrCAD SYSTEM CORP**. Pulsemos <ENTER>. A continuación el **Copyright**. Pulsemos <ENTER> y tras el mensaje de **Pulse cualquier tecla para continuar** (pulsar una tecla) aparece una pantalla de edición con la situación del cursor detectada por las coordenadas del margen superior derecho, que inicialmente se muestran en **+ .0, + .0**.

Pulsemos <ENTER> con lo que se descolgará el menú principal de características similares al de **Draft**.

Los siguientes comandos, realizan una función idéntica a sus homónimos pertenecientes al menú de **Draft**: **Again, Get Part, Jump, Macro, Quit, Tag** y **Zoom**.

Otros comandos como **Conditions, Library** y **Set** incorporan nuevos subcomandos, relativos a las partes de un componente, acompañando a los ya clásicos de funcionamiento similar a los de **Draft**.

Como comandos específicos de **LIBEDIT**, podemos considerar los siguientes: **Body, Export, Import, Name, Origin, Pin, Reference**.

Veamos cómo se emplean en la práctica estos comandos.

El estado inicial es la pantalla de edición con el menú principal desplegado en el margen superior izquierdo. La librería elegida para extraer el componente a modificar ha sido la TTL.

Con el comando **Origin**, podemos modificar las coordenadas iniciales, situándolas en cualquier parte de la pantalla de edición.

Tras cargar **LIBEDIT** y responder con **TTL.LIB** al interrogante **Read Library?** seleccionemos **Get part**. Ante la interrogación del mismo nombre, respondamos introduciendo **259** <ENTER>. Seleccionemos el **74HC259**, de las varias opciones que aparecen. Tras validar con <ENTER>, el integrado se nos mostrará en pantalla, dentro de un rectángulo.

Seleccionemos la opción **Pin** del menú principal. Tras validar, se nos muestra una barra en el margen superior de la pantalla, con una serie de subcomandos que nos van a permitir **añadir, borrar, nombrar, numerar, elegir el tipo, forma, etc.** de los pines del componente que deseamos modificar.

Notar que el puntero del ratón sólo se mueve por el perímetro o cuerpo (**body**) del integrado, no pudiendo accederse a otro lugar.

Situémonos en + **.0, + 1.0** y pulsemos <ENTER>. Queremos sustituir tanto el **nombre**, como el **número de pin**, ya que el **tipo (type)** y la **forma (shape)** pueden conservarse.

Seleccionemos **Name** <ENTER> y borremos con la tecla <Borrar Atrás> la letra **D**, tecleando en su lugar **S0** seguido de <ENTER>. Pulsemos <ENTER> de nuevo y seleccionemos la opción **Pin Number** <ENTER>. Tecleemos "**1**" seguido de <ENTER>, con lo que quedará modificado el primer **pin**.

A continuación, coloquemos el segundo **pin**, correspondiente a la entrada **S1**. Situemos el cursor en + **.0, + 2.0** y validemos con <ENTER>. Se trata ahora de **añadir** un nuevo **pin**, por lo que seleccionaremos el comando **Add**. Tras pulsar <ENTER>, respondamos con **S1** al interrogante **Name?** y con **2** <ENTER> a **Pin Number?**. A continuación, seleccionemos **Input** <ENTER> en respuesta a **Pin Type** y **Line** <ENTER> para **Pin Shape**, con lo que finaliza la colocación de este segundo **pin**.

Para la colocación del **pin 3**, procederemos de forma similar que para el **pin 1**, cambiando el nombre por **S2** y **Pin Number** por "**3**".

Coloquémonos en + **.0, + 4.0**. Vamos a eliminar el **pin** que se encuentra en esta coordenada. Para ello, seleccionemos la opción **Delete** seguida de <ENTER> con lo que el **pin** quedará anulado.

A continuación, situémonos en + .0, + 5.0 donde colocaremos la entrada **G (negada)** equivalente al **enable WD** del 4099. En respuesta a **Name?** introducimos "G" y <ENTER>, mientras que en respuesta a **Pin Number?** introducimos "14" seguido de <ENTER>.

**Type** sigue conservando el valor de **Input** y a **Shape** otorgamos el valor de **Dot**.

Procedamos a colocar la **entrada de datos D** en la posición + .0, + 6.0. Para ello utilicemos la opción **Add** y respondamos con **D, 13, Input, Line** seguidos de <ENTER> a las preguntas que de forma secuencial, caracterizan el **pin**.

Siguiendo un proceso similar coloquemos en + .0, + 7.0 la señal **CLR (negada)** equivalente a la **RST** del 4099. Para ello, seleccionemos **Name** y respondamos con **CLR** seguido de <ENTER>, mientras que a **Pin Number?** responderemos con "15" y <ENTER>. Las opciones **Type** y **Shape** conservan sus valores de **Input** y **Dot**.

Por último, eliminemos con **Delete** el **pin** situado en + .0, + 8.0 Los demás pines, correspondientes todos ellos a las salidas **Q0..Q7**, permanecen inalterados.

## Cambio de referencia

Este atributo de todo componente, puede modificarse haciendo uso de la opción **Reference** presente en el menú principal.

Al seleccionarla y pulsar <ENTER>, aparece el siguiente interrogante: **Initial Reference Designator?**. Podríamos responder con **IC**, por ejemplo, con lo que quedaría sustituida la referencia **U** por **IC**.

## Cambio de nombre

Vamos a nombrar el integrado modificado con el nombre de **LATCH8**. Para ello utilizaremos la opción **Name** del menú principal.

Al seleccionar esta opción, se descuelga un submenú de cuatro opciones que permiten añadir (**Add**), borrar (**Delete**), editar (**Edit**) o cambiar el prefijo (**Prefix**) del componente.

Seleccionemos **Edit** con <ENTER>. Un mensaje nos indica el nombre actual. Pulsemos <ENTER>. Ante el interrogante **Name?** borremos con <Borrar atrás> el nombre **74HC259** e introduzcamos el nuevo nombre **LATCH8** seguido de <ENTER>. Aparece

un nuevo interrogante, **Sheet Path?**, válido para un esquema jerárquico, lo cual no es el caso. Respondamos con <ENTER> y finalicemos con <ESCAPE>. El resultado es el que muestra la figura 4.3

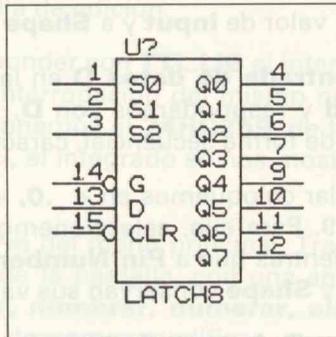


Figura 4.3

## Incorporación del componente modificado a una determinada librería

Podemos distinguir tres posibilidades:

- Incorporación del componente modificado a la misma librería conservando el nombre u otro distinto.
- Incorporación del componente modificado a otra librería distinta.
- Incorporación del componente a una librería personalizada.

### Incorporación a la misma librería

Seleccionemos el comando **Library** del menú principal. Aparece un submenú con las siguientes opciones:

**Update Current**

**List Directory**

**Browse**

**Delete Part**

**Prefix**

La 2.<sup>a</sup> y 3.<sup>a</sup> opciones realizan una función similar a sus homónimas de **Draft**, mientras que **Delete Part** borra cualquier componente de la librería seleccionada y **Prefix** edita el **prefijo de definición**, concepto que recordaremos más adelante. Cuando se selecciona esta opción, aparece un submenú que permite **añadir, borrar, editar** dicho **prefijo**.

Seleccionemos la opción **Update Current (Actualizar)**. El componente queda almacenado en memoria de forma temporal. Si deseamos que se almacene de forma permanente en un archivo, seleccionemos la opción **Quit** del menú principal.

Se presentan dos opciones: actualizar librería (**Update file**) o escribir en un archivo (**Write to file**). Con la primera opción, la librería actual, en nuestro caso la **TTL**, que-

dará con el CI **74HC259** modificado de forma permanente, aunque se realiza una copia de seguridad de la librería (**TTL.BAK**).

Con la segunda opción, podemos personalizar la librería con un nombre cualquiera (por ejemplo, **TTL1.LIB**) con el inconveniente del espacio a ocupar.

## Incorporación a otra librería existente

Puede ocurrir que sea necesario incorporar el componente modificado a otra librería para, por ejemplo, ahorrar memoria. En nuestro ejemplo, podemos incorporar el nuevo **LATCH8**, perteneciente a la librería **TTL.LIB** o a la modificada **TTL1.LIB**, a la **CMOS.LIB**, con lo que ahorraríamos el cargar cualquier librería **TTL** a la hora de editar el esquema **RELE7.SCH**.

Para conseguir ésto, seguiremos los pasos siguientes:

1. Supongamos al componente modificado, integrado en una librería, por ejemplo, **TTL1.LIB**. Editemos y salvemos dicho componente mediante el comando **Export** en un archivo que nombraremos como **LATCH.SRC**. La extensión **.SRC** es opcional. Este archivo se genera en código **ASCII**, pudiendo ser editado por cualquier editor de textos.
2. Inicializemos **LIBEDIT** mediante el comando **Quit**. Respondamos a la pregunta **Read Library?** con el nombre de la librería donde queremos incorporar el componente; en nuestro ejemplo, **CMOS.LIB**.
3. Seleccionemos el comando **Import** desde el menú principal. Responder con **LATCH.SRC** al interrogante **From to file?**. A continuación, sigamos los pasos expuestos en la sección anterior si queremos incorporar el componente a la librería, ya sea de forma temporal o permanente.

## Incorporación a una librería propia

Es la opción más utilizada normalmente. Veamos los pasos necesarios para realizar esta incorporación.

1. Al igual que en el caso anterior, salvar el componente modificado en un archivo, utilizando el comando **Export**. Supongamos que el archivo es **LATCH.SRC**.
2. Con cualquier procesador de texto que pueda producir código ASCII, editemos dicho archivo. El listado de nuestro ejemplo es el siguiente:

```

PREFIX
END
'LATCH8'
{X Size =} 6 {Y Size =} 9 {Parts per Package =} 1
L1 1 IN 'S0'
R1 4 OUT 'Q0'
R2 5 OUT 'Q1'
L3 3 IN 'S2'
R3 6 OUT 'Q2'
R4 7 OUT 'Q3'
L5 14 DOT IN 'G'
R5 9 OUT 'Q4'
R6 10 OUT 'Q5'
L7 15 DOT IN 'CLR'
R7 11 OUT 'Q6'
R8 12 OUT 'Q7'
T0 16 PWR 'VCC'
B0 8 PWR 'GND'
L2 2 IN 'S1'
L6 13 IN 'D'

```

Para convertir este archivo, en un archivo de librería, es necesario añadir el **prefijo de definición** (en la siguiente sección se amplía este concepto).

Como en nuestro caso, sólo existe un componente, bastará con colocar dos líneas al comienzo del fichero, conteniendo las órdenes **PREFIX** y **END** respectivamente. El archivo quedaría como sigue:

```

'LATCH8'
{X Size =} 6 {Y Size =} 9 {Parts per Package =} 1
L1 1 IN 'S0'
R1 4 OUT 'Q0'
R2 5 OUT 'Q1'
L3 3 IN 'S2'
R3 6 OUT 'Q2'
R4 7 OUT 'Q3'
L5 14 DOT IN 'G'
R5 9 OUT 'Q4'
R6 10 OUT 'Q5'
L7 15 DOT IN 'CLR'
R7 11 OUT 'Q6'
R8 12 OUT 'Q7'
T0 16 PWR 'VCC'
B0 8 PWR 'GND'
L2 2 IN 'S1'
L6 13 IN 'D'

```

Capítulo 5  
Utilidades

- Utilizemos el **compilador de librerías, COMPOSER.EXE**, para convertir el archivo **LATCH.SRC** en un archivo de librería, que podemos llamar **LATCH.LIB**. La orden que nos produce esta transformación es:

### **COMPOSER LATCH.SRC LATCH.LIB**

procurando encontrarnos en el subdirectorio donde se halle ubicado el archivo **COMPOSER.EXE**.

Puede ocurrir que sea necesario modificar el contenido de un archivo de librería para, por ejemplo, ahorrar memoria. En tal caso, podemos incorporar el nuevo **LATCH8**, perteneciente a la librería **LATCH.LIB** a la librería **LATCH.LIB** o a la librería **LATCH.LIB** a la librería **LATCH.LIB** a la hora de incorporar a una librería existente una librería nueva.

Para conseguir esto, seguimos los siguientes pasos:

- Supongamos el comando **LIBRARY** en un archivo de librería, por ejemplo **LATCH.LIB**. Este comando indica que se debe incorporar a la librería **LATCH.LIB** el archivo **LATCH.SRC**.

Como en nuestro caso, sólo existe un comando **LIBRARY** en el archivo **LATCH.LIB**, el comando **LIBRARY** en el archivo **LATCH.SRC** debe ser el mismo que el comando **LIBRARY** en el archivo **LATCH.LIB**.

- Inicialmente, el comando **LIBRARY** en el archivo **LATCH.LIB** debe ser el mismo que el comando **LIBRARY** en el archivo **LATCH.SRC**.

- Seleccionemos el comando **LIBRARY** en el archivo **LATCH.SRC** y lo cambiemos por **LIBRARY** en el archivo **LATCH.LIB**. Este comando **LIBRARY** en el archivo **LATCH.SRC** debe ser el mismo que el comando **LIBRARY** en el archivo **LATCH.LIB**.

Es la opción más utilizada para incorporar una librería a otra. Para realizar esta incorporación, se debe utilizar el comando **LIBRARY** en el archivo **LATCH.SRC**.

- Al igual que en el caso anterior, el comando **LIBRARY** en el archivo **LATCH.SRC** debe ser el mismo que el comando **LIBRARY** en el archivo **LATCH.LIB**.

- Con cualquier procesador de texto, puede producirse código ASCII, editando dicho archivo. El código de dicho archivo es el siguiente:

## Generalidades

Tras la confección del esquema, OrCAD SBT/III dispone de una serie de utilidades que se cargan desde el DOS tras abandonar la sesión de edición.

Cuando se invocan estas utilidades hay que tener en cuenta el tipo de estructura del fichero que hayamos confeccionado previamente mediante Draft.

Los tres tipos de estructuras, como ya vimos, son:

- Archivo jerárquico.
- Archivo de bajo nivel (flat file).
- Archivo de una sola hoja (one sheet file).

La aplicación de las utilidades difiere de un tipo a otro. Por ello, todas las utilidades incorporan la posibilidad de seleccionar el tipo de estructura mediante los switches adecuados.

## Capítulo 5 Utilidades



### Generalidades

Tras la confección del esquema, **OrCAD SDT/III** dispone de una serie de utilidades que se cargan desde el **DOS** tras abandonar la sesión de edición.

Cuando se invocan estas utilidades hay que tener en cuenta el tipo de estructura del fichero que hayamos confeccionado previamente mediante **Draft**.

Los tres tipos de estructuras, como ya vimos, son:

- Archivo jerárquico.
- Archivo de bajo nivel (flat file).
- Archivo de una sola hoja (one sheet file).

La aplicación de las utilidades difiere de un tipo a otro. Por ello, todas las utilidades incorporan la posibilidad de seleccionar el tipo de estructura mediante los **switches** adecuados.



## Estructura genérica de una utilidad

Para ejecutar cualquier **utilidad** será necesario situarnos en el directorio correspondiente donde se encuentre el archivo de extensión .EXE que corresponda a dicha **utilidad**.

La estructura genérica de una **utilidad** es la siguiente:

**nombre de la utilidad archivo fuente archivo destino / switches**

La estructura mínima de una **utilidad** corresponde a la secuencia **nombre de la utilidad archivo fuente**, donde el nombre de la utilidad (ANNOTATE, ERC, NETLIST, etc.) va seguido del nombre del archivo correspondiente al esquema confeccionado con **Draft** (extensión .SCH) que recibe el nombre de archivo fuente.

La especificación **archivo destino** es opcional y corresponde al nombre del archivo donde se almacena el esquema al que se ha aplicado la utilidad. La extensión de este archivo, aunque no es obligatoria, suele tener relación con la utilidad que se ha hecho correr. Así por ejemplo, los archivos creados con la utilidad **ANNOTATE** llevarían asociada la extensión **.ANN**, así como los creados con **NETLIST** llevan la extensión **.NET**.

En cuanto a la especificación **switches**, también opcional, hay que distinguir entre **switches** que afectan la mayoría de las utilidades y **switches** particulares de una utilidad.

A continuación se relacionan los **switches** de carácter general, así como la función que desempeña cada uno de ellos.

La especificación de uno o varios **switches** se lleva a cabo anteponiendo a la letra identificativa del **switch**, el carácter **"/"**.

### Switches de carácter general

**/A.** Especifica que el archivo fuente es un archivo con numeración de componentes (**annotation file**).

**/C.** Muestra el menú de configuración de la utilidad que se está utilizando.

/ **D.** Desciende hasta el componente definido como **sheetpath**.

/ **E.** Permite una pausa para intercambio de discos en sistemas que usan un sólo **floppy**.

/ **F.** Provoca que el **archivo fuente** sea leído como un **archivo de texto**, cuando se trate de estructuras de bajo nivel (**flat file**).

/ **H.** Evita la repetición de **hojas** duplicadas en una estructura de jerarquía compleja y en algunas utilidades reduce el tiempo de proceso.

/ **O.** Trata el archivo fuente como un archivo de estructura simple (**one-sheet**), sin posibilidad de descender a niveles jerárquicos más bajos.

/ **Q.** Hace que la utilidad que se ha invocado, corra de forma **resumida**, mostrándose sólo los mensajes de error, no el resto de información debido a otras actividades.

## Repertorio de utilidades

Tras la confección de un esquema, los componentes quedan con una **referencia** en la que aparece un interrogante. Este interrogante es sustituido por un **número de orden** si se ejecuta la utilidad **ANNOTATE**.

### Annotate

Muestra un archivo, ya sea de tipo jerárquico, plano o simple, actualizando automáticamente todas las referencias de los componentes existentes en el esquema. Esto incluye la actualización de la correspondiente numeración de los **pinos** asociados a componentes que tengan varias partes por pastilla.

El resultado de aplicar esta utilidad puede ser colocado en un **archivo de destino** o bien en la misma hoja de trabajo si usamos el **switch / M (mezclar)**.

El formato general de esta utilidad, teniendo en cuenta que las especificaciones entre corchetes son opcionales, es el siguiente:

**ANNOTATE** archivo fuente [archivo destino] [/C] [/D] [/E] [/F] [/H] [/L] [/M] [/O] [/Q] [/U]

#### Switches particulares utilizados

**/L.** Crea un informe listando las últimas referencias asignadas por **ANNOTATE**. Si no se especifica el archivo de destino, el informe es colocado en un fichero de texto con el mismo nombre que el fichero **raíz** y con la extensión **.END**.

**/M.** Provoca la **anotación** en el propio archivo fuente.

**/U.** Cambia las referencias de modo incondicional. En efecto; **ANNOTATE**, actualiza las referencias en el orden en que son colocados los componentes. Si se añade algún componente, su referencia puede resultar fuera de lugar. Esto se soluciona con **ANNOTATE /U**, ya que se actualiza la referencia de **todos** los componentes.

Apliquemos a nuestro esquema **RELE62.SCH** esta utilidad, de la siguiente forma:

## ANNOTATE /U RELE62.SCH /M /L

El resultado es la figura 5.1 correspondiente al esquema que titularemos **RELE7.SCH**.

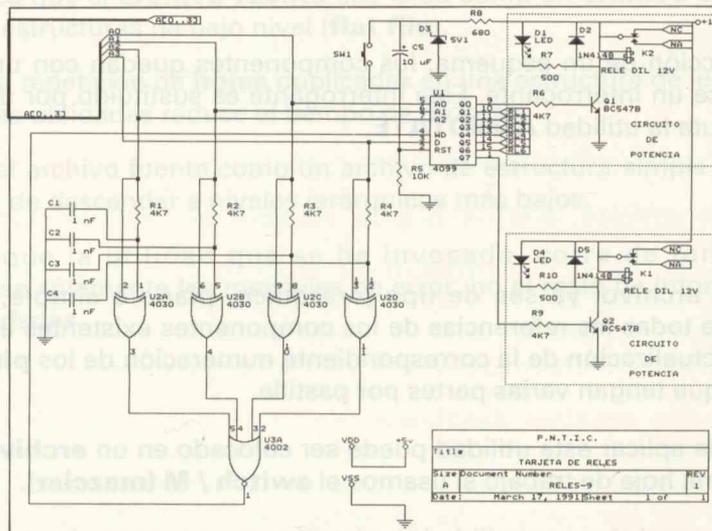


Figura 5.1

## Backanno

Puede ocurrir que en un archivo donde se ha realizado un **ANNOTATE** sea necesario actualizar o cambiar referencias, pero conservando en un archivo las referencias originales. Esto se consigue mediante la utilización de la utilidad **BACKANNO**, cuyo formato es el siguiente:

**BACKANNO archivo fuente was/is [A]/[C]/[D]/[E]/[F]/[O]/[Q]**

donde **was/is** es el nombre de un archivo de texto conteniendo las antiguas y nuevas referencias. A continuación, veamos como se realiza un archivo de este tipo. Para ello, vamos a cambiar de lugar las **referencias** correspondientes a las cuatro puertas **EXOR** del esquema **RELE7.SCH**. El proceso a seguir es el siguiente:

Mediante un editor de texto (aquí se ha utilizado el de las **Utilidades Norton**) confeccionamos un archivo donde consten las antiguas y nuevas referencias. El contenido de dicho archivo se muestra a continuación.

<b>U2A</b>	<b>U2D</b>
<b>U2B</b>	<b>U2C</b>
<b>U2C</b>	<b>U2B</b>
<b>U2D</b>	<b>U2A</b>

Salvemos este archivo tipo **was/is**, con el nombre de **CAMBIO.BAK** y ejecutemos la utilidad **BACKANNO** según el formato siguiente:

### BACKANNO RELE7.SCH CAMBIO.BAK

Si ahora invocamos **Draft** y cargamos el archivo **RELE7.SCH**, veremos cómo las **referencias** de la cuatro puertas **EXOR** han cambiado su situación y además, **también la numeración correspondiente a los pines**, como muestra la figura 5.2.

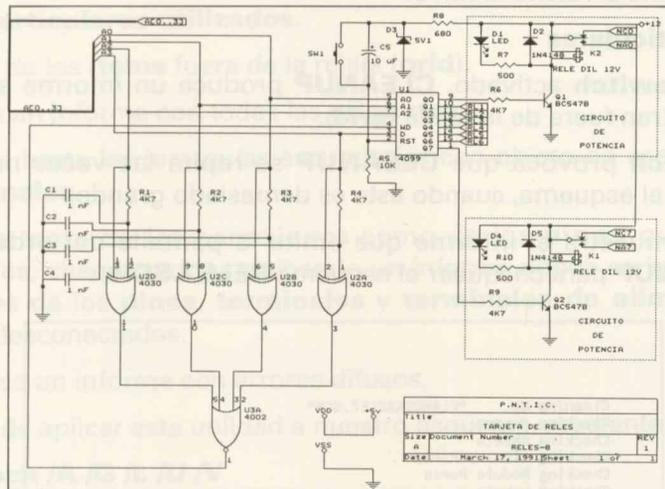


Figura 5.2

Notemos que se ha creado de forma automática un archivo de seguridad **RELE7.BAK**. Con objeto de obtener una secuencia de esquemas, renombramos desde el sistema operativo el archivo **RELE7.SCH** por **RELE78.SCH** y el archivo **RELE7.BAK** por **RELE.SCH**. Para ello, utilícese el comando **RENAME** del DOS.

## Cleanup

Esta utilidad chequea cualquier esquema diseñado con **Draft** y detecta la posible **desconexión** de cualquier **objeto** (conexión, bus, unión, terminal, etc.), así como

objetos **duplicados**, emitiendo mensajes de error por la pantalla del ordenador o a través de un archivo.

Esta utilidad evita la producción de errores posteriores, por lo que resulta obligatoria su ejecución.

El formato es el siguiente:

**CLEANUP** archivo fuente [archivo destino] [/A] [/C] [/D] [/E] [/F] [/G] [/H] [/O] [/Q] [/R]

donde **archivo fuente** puede ser cualquier tipo de archivo (jerárquico, de bajo nivel o simple) y **archivo destino** es opcional, de forma que si no se especifica, los mensajes aparecen en el monitor. Si se especifica un nombre de **archivo destino**, el informe se almacena en dicho archivo.

### Switches particulares

**/G.** Con este **switch** activado, **CLEANUP** produce un informe sobre los **items** que se encuentran fuera de la rejilla (**grid**).

**/R.** Este **switch** provoca que **CLEANUP** se repita las veces necesarias hasta chequear todo el esquema, cuando éste es demasiado grande.

La figura 5.3 muestra el informe que emite la pantalla del ordenador, cuando utilizamos **CLEANUP** para chequear el esquema **RELE7.SCH**.

```
Cleaning Up      "PLANOS\RELE7.SCH"
Checking Header
Checking Sheets
Checking Library Parts
Checking Module Ports
Checking Labels
Checking Power
Checking Bus Entries
Checking Junctions
Checking Wires and Buses
Checking Non-Orthogonal Wires and Buses
Checking Horizontal Wires
Checking Horizontal Buses
Checking Vertical Wires
Checking Vertical Buses
```

Figura 5.3

## Erc

Esta utilidad, cuyo nombre es un acrónimo de las palabras **Electrical Rules Check**, chequea cualquier tipo de esquema realizado por **Draft** para ver si es conforme a las reglas básicas eléctricas, emitiendo el informe de errores correspondiente, a través de la pantalla o almacenándose en un archivo.

Esta utilidad, junto a **CLEANUP** evita la práctica totalidad de los errores que suelen cometerse en la confección de esquemas.

El formato de esta utilidad es el siguiente:

**ERC archivo fuente [archivo destino] [/A] [/C] [/D] [/E] [/F] [/G] [/H] [/L] [/O] [/Q] [/S] [/U] [/V]**

### Switches particulares utilizados.

**/ G.** Informa de los **items** fuera de la rejilla (**grid**).

**/ L.** Produce un informe con todas las **etiquetas**.

**/ S.** Chequea hasta las jerarquías secundarias con objeto de asegurar la conexión de los **terminales**.

**/ U.** Informa tanto de las conexiones como de los **pines** que se encuentran desconectados, chequeando y emitiendo un informe donde se indican las posibles localizaciones de los **pines**, **terminales** y **terminales de alimentación** que se encuentran desconectados.

**/ V.** Especifica un informe con errores difusos.

El resultado de aplicar esta utilidad a nuestro esquema, mediante la orden:

**ERC rele7.sch /A /G /L /U /V**

es el mostrado en la figura 5.4.

```
Time Stamp - 13-JUN-1990  15:53:38
"HOJA\RELE7.SCH"
Electrical Rules Check Report
TARJETA DE RELES   Revised:      May 31, 1990
RELES-7           Revision: 1
P.N.T.I.C.
```

Figura 5.4

## Partlist

Con esta utilidad se crea un informe-resumen de todos los componentes existentes en un esquema de cualquier tipo de estructura o modificados con **ANNOTATE**.

En dicho informe, los componentes están ordenados alfabéticamente por referencia.

De modo opcional, a un informe generado con **PARTLIST** se le puede añadir información sobre el uso de un componente específico, mediante un archivo de texto especial llamado "**include file**" (archivo incluido). El formato de esta utilidad es el siguiente:

**PARTLIST** archivo fuente [archivo destino] [archivo incluido] [/A] [/C] [/D] [/E] [/F] [/I] [/L] [/N] [/O] [/Q] [/S] [/V]

### Switches especiales

- /I.** Especifica que un archivo tipo **include** va a ser añadido al informe.
- /L.** Especifica que la entrada de cada componente puede estar, dentro del informe, con una línea de separación.
- /N.** Provoca que la cabecera que se incluye al ejecutar esta utilidad no se inserte en cada página.
- /S.** Produce un informe a simple espacio.
- /V.** Provoca que los informes de salida sean de formato estándar.

El resultado de aplicar esta utilidad al esquema **RELE7.SCH** es el que se muestra a continuación.

TARJETA DE RELES  
RELES-7  
Bill Of Materials  
Page 1

Revised: May 31, 1990  
Revisión: 1  
June 18, 1990 17:25:16

Item	Quantity	Reference	Part
1	4	C1,C2,C3,C4	1 nF
2	1	C5	10 uF
3	2	D1,D4	LED
4	2	D2,D5	1N4148
5	1	D3	5V1
6	2	K1,K2	RELE DIL 12V
7	2	Q1,Q2	BC547B
8	6	R1,R2,R3,R4,R6,R9	4K7
9	1	R5	10K
10	2	R7,R10	500
11	1	R8	680
12	1	SW1	PULSADOR
13	1	U1	4099
14	1	U2	4030
15	1	U3	4002

## Netlist

Esta utilidad realiza un muestreo de un esquema realizado por **Draft**, ya sea jerárquico, de bajo nivel, simple o anotado, generando un archivo tipo **netlist** en formato seleccionable.

Un archivo **netlist** se caracteriza por tener el siguiente formato:

**NETLIST** archivo fuente [archivo destino] [formato] [/A] [/B] [/C] [/D] [/E] [/F] [/G] [/H] [/I] [/K] [/L] [/M] [/N] [/O] [/P] [/Q] [/S] [/U] [/V]

donde archivo fuente puede ser tanto el **raíz** de un archivo jerárquico, el nombre de un archivo de texto en un archivo de bajo nivel, el nombre de un archivo **simple** o el nombre de un archivo **anotado**. En los tres últimos casos, deberá especificarse el **switch** adecuado (/F, /O y /A respectivamente).

El **archivo destino**, es opcional y puede ser cualquier ruta del DOS seguida del nombre del archivo tipo **netlist** que queramos dar (normalmente la extensión es **.NET**). En el caso de utilizarse un formato especial, es necesario especificar con un **switch** (/S).

**Formato** es el nombre de un tipo especial de formato con el que se quiere generar la **netlist**. Cuando no se especifica el tipo de **formato**, por defecto se genera una **netlist** en formato **EDIF**.

La verdadera potencia de **SDT** consiste precisamente en la facilidad que posee para generar **archivos de netlist** de distintos formatos.

Podemos elegir entre los siguientes:

<b>APPLICONBRAVO</b>	<b>COMPUTERVISION</b>	<b>MENTOR</b>	<b>SCICARDS</b>
<b>APPLICONLEAP</b>	<b>EDIF</b>	<b>MULTIWIRE</b>	<b>SPICE</b>
<b>ALGOREX</b>	<b>EEDESIGNER</b>	<b>OrCADPCB</b>	<b>TANGO</b>
<b>ALTERAADF</b>	<b>FLATEDIF</b>	<b>PCAD</b>	<b>TELESIS</b>
<b>CADNETIX</b>	<b>FUTURENET</b>	<b>PCADNLT</b>	<b>VECTRON</b>
<b>CALAY</b>	<b>HILO</b>	<b>PSPICE</b>	<b>VSTMODEL</b>
<b>CASE</b>	<b>INTELADF</b>	<b>RACALREDAC</b>	<b>WIRELIST</b>
<b>CBDS</b>	<b>INTERGRAPH</b>	<b>SALT</b>	

## Switches particulares

- / B.** Sólo se aplica cuando se quiera generar una **netlist** en formato **WIRELIST**.
- / G.** Con este **switch** activo, **NETLIST** produce una lista de **items** que se encuentran fuera del **grid** y los coloca en un archivo con el mismo nombre de **archivo destino** y extensión **.GRD**.
- / I.** Con este **switch**, **NETLIST** asigna un nombre tipo **net** a todos los **pines**, los desconectados inclusive. Esto resulta muy útil cuando se utiliza un archivo de **netlist** con otro tipo de **software**, como simuladores o **routers** que requieren un listado de todos los **pines**, los desconectados incluidos.
- / K.** Sólo se aplica a **netlists** de formato **FutureNet** en los casos en que hay que especificar atributos de terminales.
- / L.** Se genera un informe con todas las **etiquetas** y **terminales** que se encuentren conectados. **NETLIST** los coloca en un archivo con el mismo nombre que **archivo destino** pero con la extensión **.LAB**.
- / M.** Se utiliza para formatos de **FutureNet** y realiza una función parecida a **/ K**.
- / N.** Sólo tiene efecto para **netlists** generadas en formatos **SPICE**, **FUTURENET**, **IntelADF**, **AlteraADF** y **VSTMODEL**.
- / P.** Provoca que la salida generada por **NETLIST** coloque **números de pin** en lugar de **nombres de pin** en formatos **FutureNet** y **EDIF**.
- / S.** Especifica que la **netlist** que se va a generar va a tener un formato especial diferente de **EDIF** (ver lista de formatos).
- / U.** Produce un informe donde constan los **cables** y **pines** que se encuentran desconectados, almacenándolos en un archivo del mismo nombre que **archivo destino** y extensión **.NC**. También informa de los **pines**, **terminales** y **terminales de alimentación** que pueden estar abiertos.
- / V.** Le dice a **NETLIST** que asigne a los atributos de **power** compatibles con el formato **FutureNet** con los siguientes valores de los **pines** que OrCAD/SDT III asigna a los **powers**:

<b>GND</b>	<b>100</b>
<b>+5V</b>	<b>101</b>
<b>+12V</b>	<b>105</b>
<b>-12V</b>	<b>106</b>
<b>VEE</b>	<b>107</b>

Ejecutemos esta utilidad con el archivo **RELE7.SCH**. Para ello, desde el directorio donde se encuentre **NETLIST.EXE** invoquemos lo siguiente:

**NETLIST RELE7.SCH RELE7.NET /O /G /U**

Se crean dos archivos: uno, en formato **EDIF**, conteniendo la **netlist** propiamente dicha (RELE7.NET) y otro (RELE7.NC), debido al **switch / U**, conteniendo una lista de los **objetos** que presentan algún problema a la hora de realizar un **PCB**.

A continuación se muestra el contenido de ambos archivos.

```
(EDIF rele7_sch
(status
(EDIFVersion 1 1 0)
(EDIFLevel 0)
(Written
(TimeStamp 1990 6 13 15 45 59)
(comment "The ABOVE TimeStamp is local time")
(accounting Program "NETLIST.EXE")
(accounting ProgramVersion " V3.22 26-Oct-89")
(comment "(C) Copyright 1985,1986,1987 OrCAD Systems Corporation ALL RIGHTS
RESERVED.")
)
)
(external DEVICE_LIB)
(external CMOS_LIB)
(external TTL1_LIB)
(design rele7_sch (qualify lib root))
(library lib
(cell root
(status
(Written
(TimeStamp 1990 5 31 18 0 24)
(comment "The ABOVE TimeStamp is local time")
(comment " May 31, 1990")
(comment "Sheet 7 of 8")
```

```

(comment "DOCUMENT NUMBER RELES-7")
(comment "REVISION 1")
(comment "TITLE TARJETA DE RELES")
(comment "P.N.T.I.C.")
)
)
(view NETLIST root_NET
(interface
(define input port VCC)
(define input port GND)
(define input port VDD)
(define output port RL1)
(define output port RL2)
(define output port RL3)
(define output port RL4)
(define output port RL5)
(define output port RL6)
(define input port VSS)
(define input port C_0
(rename C_0 "C 0"))
(define input port C_1
(rename C_1 "C 1"))
(define input port C_2
(rename C_2 "C 2"))
(define input port C_3
(rename C_3 "C 3"))
)
(contents
(define local signal A0)
(define local signal A1)
(define local signal A2)
(define local signal A3)
(instance (qualify CMOS_LIB X_4099) X_4099_NET U1)

```

```

(instance (qualify CMOS_LIB X_4030) X_4030_NET U2A)
(instance (qualify CMOS_LIB X_4030) X_4030_NET U2B)
(instance (qualify CMOS_LIB X_4030) X_4030_NET U2C)
(instance (qualify CMOS_LIB X_4030) X_4030_NET U2D)
(instance (qualify DEVICE_LIB LED) LED_NET D1)
(instance (qualify CMOS_LIB X_4002) X_4002_NET U3A)
(instance (qualify DEVICE_LIB X_4K7) X_4K7_NET R1)
(instance (qualify DEVICE_LIB X_4K7) X_4K7_NET R2)
(instance (qualify DEVICE_LIB X_4K7) X_4K7_NET R3)
(instance (qualify DEVICE_LIB X_4K7) X_4K7_NET R4)
(instance (qualify DEVICE_LIB X_10K) X_10K_NET R5)
(instance (qualify DEVICE_LIB X_4K7) X_4K7_NET R6)
(instance (qualify DEVICE_LIB X_500) X_500_NET R7)
(instance (qualify DEVICE_LIB X_680) X_680_NET R8)
(instance (qualify DEVICE_LIB X_1_NF) X_1_NF_NET C1)
(instance (qualify DEVICE_LIB X_1_NF) X_1_NF_NET C2)
(instance (qualify DEVICE_LIB X_1_NF) X_1_NF_NET C3)
(instance (qualify DEVICE_LIB X_1_NF) X_1_NF_NET C4)
(instance (qualify DEVICE_LIB X_10_UF) X_10_UF_NET C5)
(instance (qualify DEVICE_LIB X_1N4148) X_1N4148_NET D2)
(instance (qualify DEVICE_LIB X_5V1) X_5V1_NET D3)
(instance (qualify DEVICE_LIB BC547B) BC547B_NET Q1)
(instance (qualify DEVICE_LIB PULSADOR) PULSADOR_NET SW1)
(instance (qualify DEVICE_LIB LED) LED_NET D4)
(instance (qualify DEVICE_LIB X_4K7) X_4K7_NET R9)
(instance (qualify DEVICE_LIB X_500) X_500_NET R10)
(instance (qualify DEVICE_LIB X_1N4148) X_1N4148_NET D5)
(instance (qualify DEVICE_LIB BC547B) BC547B_NET Q2)
(instance (qualify DEVICE_LIB RELE_DIL_12V) RELE_DIL_12V_NET K1)
(instance (qualify DEVICE_LIB RELE_DIL_12V) RELE_DIL_12V_NET K2)
(joined
(qualify SW1 X_2)
(qualify R8 X_1)

```

```

)
(joined
(qualify R8 X_2)
VCC
(qualify D4 ANODE)
)
(joined
A0
(qualify U1 A0)
(qualify U2C I1)
(qualify R3 X_1)
)
(joined
A1
(qualify U1 A1)
(qualify U2B I1)
(qualify R2 X_1)
)
(joined
A2
(qualify U1 A2)
(qualify U2A I1)
(qualify R1 X_1)
)
(joined
A3
(qualify U2D I1)
(qualify U1 D)
(qualify R4 X_1)
)
(joined
(qualify K2 COIL2)
(qualify Q1 COLLECTOR)

```

```

) qualify R4 X_2)
(joined C4 X_2)
(qualify C5 X_2)
(qualify U1 RST)
(qualify R5 X_1)
) qualify R1 COIL2)
(joined Q2 COLLECTOR)
(qualify D1 CATHODE)
(qualify R7 X_1)
) qualify D4 CATHODE)
(joined R10 X_1)
(qualify U1 Q0)
(qualify R6 X_1)
) qualify R9 X_2)
(joined Q2 BASE)
(qualify R6 X_2)
(qualify Q1 BASE)
) qualify Q2 EMITTER)
(joined C1 X_1)
(qualify U1 Q1)
RL1 qualify Q1 EMITTER)
) qualify D3 ANODE)
(joined
(qualify U1 Q2)
RL2
) qualify U2A O)
(joined U3A I3)
(qualify U1 Q3)
RL3
) qualify U2B O)
(joined U3A I2)
(qualify U1 WD)
(qualify U3A O)

```

)  
(joined  
RL4  
(qualify U1 Q4)  
)  
(joined  
RL5  
(qualify U1 Q5)  
)  
(joined  
(qualify U1 Q6)  
RL6  
)  
(joined  
(qualify U1 Q7)  
(qualify R9 X\_1)  
)  
(joined  
(qualify R1 X\_2)  
(qualify C1 X\_2)  
(qualify U2A I0)  
)  
(joined  
(qualify R2 X\_2)  
(qualify C2 X\_2)  
(qualify U2B I0)  
)  
(joined  
(qualify R3 X\_2)  
(qualify C3 X\_2)  
(qualify U2C I0)  
)  
(joined

```

(qualify R4 X_2)
(qualify C4 X_2)
(qualify U2D I0)
)
(joined
(qualify K1 COIL2)
(qualify Q2 COLLECTOR)
)
(joined
(qualify D4 CATHODE)
(qualify R10 X_1)
)
(joined
(qualify R9 X_2)
(qualify Q2 BASE)
)
(joined
(qualify Q2 EMITTER)
(qualify C1 X_1)
(qualify R5 X_2)
(qualify Q1 EMITTER)
(qualify D3 ANODE)
GND
)
(joined
(qualify U2A O)
(qualify U3A I3)
)
(joined
(qualify U2B O)
(qualify U3A I2)
)
(joined

```

```

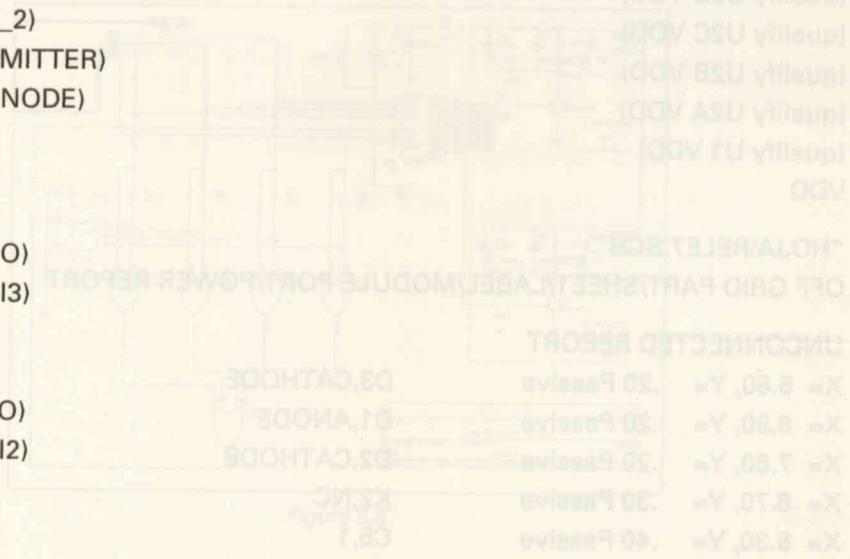
K2.COMMON
K2.NO
K2.COIL1
D2.ANODE
R7.2
SW1.1
C2.1
D5.CATHODE
K1.NO
K1.COMMON
K1.NO
C3.1
K1.COIL1
D5.ANODE
R10.2
C4.1

```

```

(qualify U3C VDD)
(qualify U3A I1)
)
(joined
(qualify U3D O)
(qualify U3A I0)
)
(joined
(qualify U3A VSS)
(qualify U3D VSS)
(qualify U3C VSS)
(qualify U3B VSS)
(qualify U3A VSS)
)
(qualify U1 VSS)

```



(qualify U2C O)  
 (qualify U3A I1)  
 )  
 (joined  
 (qualify U2D O)  
 (qualify U3A I0)  
 )  
 (joined  
 (qualify U3A VSS)  
 (qualify U2D VSS)  
 (qualify U2C VSS)  
 (qualify U2B VSS)  
 (qualify U2A VSS)  
 VSS  
 (qualify U1 VSS)  
 )

(joined  
 (qualify U3A VDD)  
 (qualify U2D VDD)  
 (qualify U2C VDD)  
 (qualify U2B VDD)  
 (qualify U2A VDD)  
 (qualify U1 VDD)  
 VDD

"HOJA\RELE7.SCH"

OFF GRID PART/SHEET/LABEL/MODULE PORT/POWER REPORT

UNCONNECTED REPORT

X= 5.80, Y= .20 Passive	D3,CATHODE
X= 6.90, Y= .20 Passive	D1,ANODE
X= 7.60, Y= .20 Passive	D2,CATHODE
X= 8.70, Y= .30 Passive	K2,NC
X= 5.30, Y= .40 Passive	C5,1

- X= 7.90, Y= .40 Passive
  - X= 8.70, Y= .50 Passive
  - X= 7.90, Y= .60 Passive
  - X= 7.60, Y= .70 Passive
  - X= 7.50, Y= .80 Passive
  - X= 4.90, Y= 1.10 Passive
  - X= .90, Y= 3.10 Passive
  - X= 7.60, Y= 3.10 Passive
  - X= 8.70, Y= 3.20 Passive
  - X= 7.90, Y= 3.30 Passive
  - X= 8.70, Y= 3.40 Passive
  - X= .90, Y= 3.50 Passive
  - X= 7.90, Y= 3.50 Passive
  - X= 7.60, Y= 3.60 Passive
  - X= 7.50, Y= 3.70 Passive
  - X= .90, Y= 3.90 Passive
- K2,COMMON
  - K2,NO
  - K2,COIL1
  - D2,ANODE
  - R7,2
  - SW1,1
  - C2,1
  - D5,CATHODE
  - K1,NC
  - K1,COMMON
  - K1,NO
  - C3,1
  - K1,COIL1
  - D5,ANODE
  - R10,2
  - C4,1

Ante los mensajes contenidos en este archivo, debemos tomar las medidas oportunas para corregir nuestro esquema. Ello se consigue colocando las **uniones** oportunas en las coordenadas que indica el archivo. De este modo obtenemos la figura 5.5.

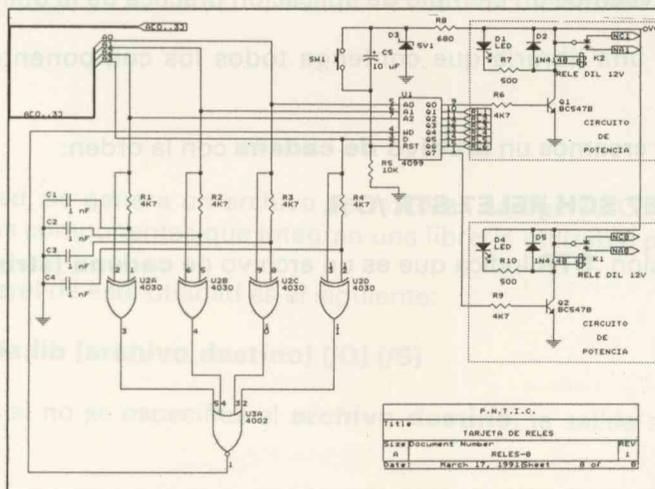


Figura 5.5

## Libarch

Esta utilidad toma todos los componentes que han sido extraídos de las librerías correspondientes durante la confección de un esquema y crea un **archivo fuente** conteniendo tan solo aquellos componentes que intervienen en el esquema.

El formato que emplea esta utilidad es el siguiente:

**LIBARCH archivo fuente [archivo destino] [/A]/[C]/[D]/[F]/[L]/[O]/[Q]/[S]**

### Switches particulares

**/L.** Con este **switch** activado, **LIBARCH** crea un archivo de **cadena** en código ASCII conteniendo el nombre de todos los componentes usados en el esquema, entrecomillados y separados, cada uno, por una línea.

**/S.** Cuando se activa este **switch**, **LIBARCH** construye, a partir del archivo de **cadena**, un archivo **fuentes** de librería, listo para ser usado por el compilador **COMPOSER**.

Esto tiene una aplicación muy interesante, ya que podemos crear una **sol**a librería que contenga todos los componentes que van a intervenir en el esquema, con el consiguiente ahorro de memoria y de tiempo a la hora de cargar las librerías.

A continuación veamos un ejemplo de aplicación práctica de la utilidad **LIBARCH**.

Vamos a crear una librería que contenga todos los componentes del esquema **RELE7.SCH**.

En primer lugar creamos un **archivo de cadena** con la orden:

**LIBARCH RELE7.SCH RELE7.STR /O /L**

donde la extensión **.STR** indica que es un archivo de **cadena (string)**. El listado de este archivo es

```
'4099'  
'4030'  
'LED'  
'GND POWER'  
'4002'
```

'RESISTOR'  
'CAP NP'  
'CAPACITOR POL'  
'DIODE'  
'DIODE ZENER'  
'NPN'  
'SW PUSHBUTTON'  
'RELAY SPDT'

En segundo lugar vamos a crear, a partir del archivo de **cadena RELE7.STR**, un archivo **fuentes** capaz de ser compilado (extensión **.SRC**) por **COMPOSER**. La orden de ejecución será:

**LIBARCH RELE7.STR RELE7.SRC /S**

Por último compilemos el archivo fuente **RELE7.SRC** para obtener el archivo de **librería RELE7.LIB**. Para ello, la orden es la siguiente:

**COMPOSER RELE7.SRC RELE7.LIB**

Desde este momento, cada vez que deseemos utilizar el esquema **RELE7.SCH** para efectuar cualquier modificación, bastará con cargar la librería **RELE7.LIB** que tan sólo ocupa **5KB**.

## Liblist

Con esta utilidad, se genera un archivo de texto en código ASCII que contiene un listado de todos los componentes que integran una librería utilizable por **Draft**.

El formato general de esta utilidad es el siguiente:

**LIBLIST librería.lib [archivo destino] [/Q] [/S]**

Como siempre, si no se especifica el **archivo destino**, la salida se efectúa por la pantalla.

El **switch /S** imprime un informe que muestra el número total de dispositivos que contiene la librería.

En efecto; veamos como podemos efectuar un listado de los componentes que integran la librería **RELE7.LIB** creada anteriormente. La orden a ejecutar es:

### **LIBLIST RELE7.LIB RELE7.TXT**

El listado de **rele7.txt** da como resultado:

RELE7.LIB  
4002  
4030  
4099  
CAP NP  
CAPACITOR POL  
DIODE  
DIODE ZENER  
GND POWER  
LED  
NPN  
RELAY SPDT  
RESISTOR  
SW PUSHBUTTON

Si hubiésemos utilizado el **switch/S** habríamos obtenido un número total de dispositivos empleados de 13.

## **Plotall**

Esta utilidad **plotea** un conjunto de esquemas en un **plotter** o en una impresora.

Cuando se usa esta utilidad, hay que asegurarse de que se ha tenido en cuenta la elección de un **driver de plotter** durante la configuración de **Draft**. Ya vimos en el capítulo 1, las especificaciones de esta configuración (canal, baudios, paridad, etc.).

**PLOTALL** respeta los valores asignados en la opción de configuración **Template Table**. No obstante, dichos valores pueden **escalarse** desde **PLOTALL** con el **switch/S**.

El formato de empleo de esta utilidad es el siguiente:

**PLOTALL archivo fuente [archivo destino]**

**[/A] [/C] [/D] [/E] [/F] [/G] [/H] [/N] [/O] [/P] [/Q] [/R] [/S arg] [/U arg] [/W] [/X arg]  
[/Y arg]**

Veamos el significado de los que son particulares de esta utilidad.

**/ G.** Plotea referencias de **grid** sobre el borde de salida.

**/ N.** Produce una ausencia de relleno, de modo que en aquellos componentes con especificaciones de relleno, éste no se produce durante el ploteo.

**/ P.** Dirige la salida hacia la impresora en lugar de hacia el plotter.

**/ R.** Este **switch** se usa cuando se utiliza un plotter con alimentador accionado por rodillo.

**/ S arg.** Especifica un factor de escala para la salida ploteada.

La siguiente tabla muestra los factores de escala que se pueden usar para plotear hojas de trabajo en diferentes formatos de papel de plotter.

**TAMAÑO DE LA  
HOJA DE TRABAJO**

**TAMAÑO DEL PAPEL DEL PLOTTER**

	<b>A</b>	<b>B</b>	<b>C</b>	<b>D</b>	<b>E</b>
<b>A</b>	<b>1.000</b>	<b>1.347</b>	<b>2.082</b>	<b>2.806</b>	<b>4.351</b>
<b>B</b>	<b>0.638</b>	<b>1.000</b>	<b>1.329</b>	<b>2.083</b>	<b>2.776</b>
<b>C</b>	<b>0.474</b>	<b>0.638</b>	<b>1.000</b>	<b>1.329</b>	<b>2.089</b>
<b>D</b>	<b>0.301</b>	<b>0.472</b>	<b>0.627</b>	<b>1.000</b>	<b>1.311</b>
<b>E</b>	<b>0.224</b>	<b>0.301</b>	<b>0.472</b>	<b>0.627</b>	<b>1.000</b>

Para escalar el tamaño del plotter con la utilidad **PLOTALL**, basta buscar en la tabla el factor de escala correspondiente al tamaño de la hoja de trabajo y el tamaño del papel del plotter que estamos usando.

Como ejemplo vamos a imprimir el esquema **CPU.SCH**, que originalmente se encuentra en formato DIN A-2 (tamaño C), realizando la conversión oportuna según la tabla. El formato de ejecución es el siguiente:

**PLOTALL CPU.SCH /S 0.474 /A /P**

y el resultado queda plasmado en la figura 5.6.

**/ U arg.** Provoca el uso de una plantilla equivalente. Este **switch** hace que **PLOTALL** use valores de **X,Y** equivalentes del tamaño de hoja especificado, como configurados en la plantilla. El valor **arg** identifica el tamaño de hoja que puede ser uno de los siguientes: **A, B, C, D** o **E**. Este **switch** es incompatible con los **switches /X** y **/Y**.

**/ W.** Especifica qué ancho de papel va a ser usado durante el ploteo o la impresión.

**/ X arg.** Puede considerarse como una abcisa relativa. En efecto; si establecemos, por ejemplo, **/ X -4.00** (-4.00 es el argumento) esto provoca un desplazamiento relativo de la abcisa X, de 4 pulgadas a la izquierda del origen del plotter. La especificación de este **switch** lleva consigo la especificación del **switch / Y arg.**

Atención con el origen del plotter. Mientras un tipo de plotter tiene el origen en el centro, otro tipo de plotter lo tiene en una esquina.

**/ Y arg.** Realiza la misma función que el **switch** anterior, pero refiriéndose a ordenadas.

Cuando se usan los **switches /X, /Y** y **/S**, es conveniente utilizarlos en el orden **/S arg /X arg /Y arg.**

### **Drivers para plotter**

**OrCAD/SDT III** incorpora los siguientes **drivers** especiales para plotter:

**DXF.DRV.** Produce un archivo capaz de ser utilizado por **AutoCAD**.

**PSCRIPT.DRV.** Sirve para usar impresoras y plotter con **PostScript**.

## **Printall**

Realiza una función parecida a la de **PLOTALL**. El formato de ejecución de la instrucción es el siguiente:

**PRINTALL archivo fuente [archivo destino] [/A] [/C] [/D] [/E] [/F] [/G] [/H] [/O] [/Q] [/W]**

donde el archivo destino es opcional y si no se especifica, la salida se produce por la impresora.

Entre los **switches** de aplicación particular, podemos citar los siguientes:

**/ H.** Se utiliza para acelerar el tiempo del proceso.

**/ W.** Verifica si existe o no papel en la impresora.

## Crossref

Muestrea un diseño, recoge información sobre todos los componentes usados en dicho diseño y crea un informe de **referencia cruzada** listando la localización de cada componente.

**CROSSREF** produce dos listados de salida. El primero se ordena por referencia y valor de los componentes mientras que el segundo es ordenado por valor y referencia.

Además, **CROSSREF**, presenta una columna llamada **sheetname** donde se lista la hoja en que aparece cada componente.

El formato para hacer uso de esta aplicación es el siguiente:

**CROSSREF archivo fuente [archivo destino] [/A] [/C] [/D] [/F] [/I] [/N] [/O] [/P] [/Q] [/R] [/S] [/U]**

### Switches particulares

**/ I.** Efectúa un chequeo sobre los componentes que tienen la misma referencia.

**/ N.** Ordena la salida por nombres.

**/ P.** Muestra las coordenadas de todos los componentes.

**/ R.** Ordena la salida por referencias.

**/ S.** Produce informes a simple espacio.

**/ U.** Lista las partes no usadas en pastillas con múltiples partes.

A continuación se muestra el resultado de aplicar esta utilidad al esquema **RELE7.SCH**, según el formato:

**CROSSREF RELE7.SCH RELE7.CRF /O /S /R**

## TARJETA DE RELES

Revised: May 31, 1990

## RELES-7

Revision: 1

## Part Cross Reference Listing

June 18, 1990 13:39:18 Page 1

Item	Reference Part	Sheetname	Sheet#	Filename	X	Y
1	C1	1 nF	<<<root>>>	7	RELE7.SCH	1.00, 2.60
2	C2	1 nF	<<<root>>>	7	RELE7.SCH	1.00, 3.00
3	C3	1 nF	<<<root>>>	7	RELE7.SCH	1.00, 3.40
4	C4	1 nF	<<<root>>>	7	RELE7.SCH	1.00, 3.80
5	C5	10 uF	<<<root>>>	7	RELE7.SCH	5.20, .50
6	D1	LED	<<<root>>>	7	RELE7.SCH	6.80, .30
7	D2	1N4148	<<<root>>>	7	RELE7.SCH	7.50, .30
8	D3	5V1	<<<root>>>	7	RELE7.SCH	5.70, .30
9	D4	LED	<<<root>>>	7	RELE7.SCH	6.80, 3.20
10	D5	1N4148	<<<root>>>	7	RELE7.SCH	7.50, 3.20
11	K1	RELE DIL 12V	<<<root>>>	7	RELE7.SCH	8.20, 3.10
12	K2	RELE DIL 12V	<<<root>>>	7	RELE7.SCH	8.20, .20
13	Q1	BC547B	<<<root>>>	7	RELE7.SCH	7.70, 1.20
14	Q2	BC547B	<<<root>>>	7	RELE7.SCH	7.70, 4.10
15	R1	4K7	<<<root>>>	7	RELE7.SCH	1.80, 2.60
16	R2	4K7	<<<root>>>	7	RELE7.SCH	2.80, 2.60
17	R3	4K7	<<<root>>>	7	RELE7.SCH	3.80, 2.60
18	R4	4K7	<<<root>>>	7	RELE7.SCH	4.80, 2.60
19	R5	10K	<<<root>>>	7	RELE7.SCH	5.20, 2.10
20	R6	4K7	<<<root>>>	7	RELE7.SCH	7.00, 1.20
21	R7	500	<<<root>>>	7	RELE7.SCH	7.10, .70
22	R8	680	<<<root>>>	7	RELE7.SCH	6.20, .10
23	R9	4K7	<<<root>>>	7	RELE7.SCH	7.00, 4.10
24	R10	500	<<<root>>>	7	RELE7.SCH	7.10, 3.60
25	SW1	PULSADOR	<<<root>>>	7	RELE7.SCH	4.80, .50
26	U1	4099	<<<root>>>	7	RELE7.SCH	5.70, 1.20
27	U2A	4030	<<<root>>>	7	RELE7.SCH	1.60, 3.80
28	U2B	4030	<<<root>>>	7	RELE7.SCH	2.60, 3.80
29	U2C	4030	<<<root>>>	7	RELE7.SCH	3.60, 3.80
30	U2D	4030	<<<root>>>	7	RELE7.SCH	4.60, 3.80
31	U3A	4002	<<<root>>>	7	RELE7.SCH	3.10, 5.70

## FLdattrib

Esta utilidad cambia el atributo **Visible** de un campo específico dentro de un esquema. El atributo puede ser **visible** o **invisible**, mientras que el campo que afecta al componente, puede ser la **referencia**, el **valor** o cualquiera de los ocho campos que afectan al componente.

El formato de esta utilidad es el siguiente:

**FLDATTRIB archivo fuente campo [/C] [/E] [/F] [/H] [/I] [/O] [/Q] [/U] [/V]**

donde **archivo fuente** es el archivo con que se va a operar, pudiendo admitir los tres tipos de estructura, y hasta un archivo anotado.

**Campo** identifica el **campo** cuyos atributos visibles queremos cambiar. Estos son los siguientes: **R (referencia), V (valor), 1,2,3,4,5,6,7,8 (campos del componente)**.

En el momento de ejecutarse esta utilidad, se produce un archivo de seguridad con el mismo nombre que el archivo fuente.

### Switches especiales

**/ I.** Cambia el campo o los campos especificados en **invisibles**.

**/ U.** Coloca el parámetro de **visibilidad** de modo incondicional, aunque no exista nada en el campo.

**/ V.** Cambia el campo o los campos especificados en **visibles**.

Veamos con un ejemplo de aplicación, de cómo se comporta esta utilidad. El formato de ejecución sería:

```
FLDATRB RELE7.SCH V /O /I
```

donde RELE7.SCH es el archivo fuente, **V** es el campo **valor** y el **switch I**, provoca que el valor de cada componente permanezca invisible. Como indicamos anteriormente, se crea el archivo RELE7.BAK de forma automática.

El resultado de aplicar esta utilidad es el esquema de la figura 5.7, donde podemos constatar la ausencia del campo **valor** del componente.

## Fldstuff

Esta utilidad busca y reemplaza **campos** de componente. Esta utilidad está íntimamente relacionada con la opción **Key Field** del menú de configuración.

**FLDSTUFF** utiliza la entrada **FLDSTUFF Combine for Value/Field** para construir una **cadena de texto** llamada **match string**.

El formato de esta utilidad es el siguiente:

```
FLDSTUFF archivo fuente campo archivo material [/C] [/E] [/F] [/H] [/I] [/K]  
[/N] [/O] [/Q] [/R] [/U] [/V]
```

donde **archivo fuente** es, como de costumbre, cualquiera de los tres tipos de estructura, debiéndose especificar los **switches** correspondientes en cada caso (/F, /H, /O).

**Campo** identifica, al igual que en la utilidad anterior, el **campo** del componente que se va a **materializar** (referencia, valor, 1..8).

**Archivo material** es el nombre de un archivo ASCII que contiene la lista de variables. Por conveniencia se le añade la extensión **.STF**.

### Switches particulares

/ **I**. Especifica qué campo puede ser invisible.

/ **K**. Conserva idéntica **visibilidad** para todos los campos.

/ **N**. No convierte **cadena materiales** en mayúsculas.

/ **R**. Crea un informe donde se muestra toda la actividad que se ha producido durante la ejecución de **FLDSTUFF**. Para ello, es necesario asignar un nombre al archivo que va a contener dicho informe, que debe ir a continuación del nombre del **archivo material**.

/ **U**. Cambia incondicionalmente el campo especificado.

/ **V**. Especifica que el campo **materializado** será visible.

## Extract

Esta utilidad se emplea para extraer archivos **PLD** en código fuente desde un esquema y crear un archivo que pueda ser usado con el módulo **OrCAD/PLD**.

El formato de empleo de esta utilidad es el siguiente:

**EXTRACT archivo fuente [/A] [/C] [/D] [/F] [/O] [/P] [/Q] [/S arg]**

### Switches especiales.

/ **P**. Extrae información de la **PLD**.

/ **S arg**. Sólo extrae información de la **PLD** especificada en **arg**.

## Simple

Convierte una jerarquía compleja en una jerarquía simple. Esta utilidad suele ser usada para generar **netlist** válidas para el módulo **OrCAD/PCB II** a partir de diseños

de jerarquía compleja. También puede ser usada para simplificar diseños complejos en los que se va a usar **notación incremental**.

El formato de empleo de esta utilidad es el siguiente:

**SIMPLE** archivo fuente archivo destino [/C]/[E]/[Q]

## Treelist

Esta utilidad muestrea diseños de estructura jerárquica y muestra el nombre de las **hojas**, los nombres de los archivos asociados a la **hoja de trabajo**, así como la fecha de la última modificación. Esta utilidad es usada para organizar y conservar la pista de **hojas de trabajo** en esquemas jerárquicos.

El formato de empleo normal de esta utilidad es el siguiente:

**TREELIST** archivo fuente [archivo destino] [/C] [/D] [/E] [/F] [/Q]



## Xferovl

Esta utilidad transfiere información sobre la configuración desde una versión a otra del archivo **OrCADSDT.OVL**.

**XFEROVL** sólo trabaja con versiones superiores a la 3.12 de **OrCAD/SDT III**.

Para actualizar el archivo **OrCADSDT.OVL** sigamos los siguientes pasos:

1. Situémonos en el directorio donde se encuentren los archivos .EXE y .OVL de **OrCAD/SDT III**.
2. Renombrar el archivo **OrCADSDT.OVL** para su conservación de seguridad. Para ello tecleemos:

```
REN OrCADSDT.OVL OrCADSDT.OLD <ENTER>
```

3. Actualizar el archivo renombrado con:

```
XFEROVL OrCADSDT.OLD OrCADSDT.OLV <ENTER>
```

4. Ejecutar DRAFT normalmente.



## Requisitos previos

Suponemos configurado OrCAD/VST según los directorios siguientes:

- 1.º LIBRARY, para librerías, previamente descompactadas.
- 2.º DESIGN, para almacenar los archivos .NET creados con ORCAD/SDT III, los ficheros de impulsos (.STM) y de muestreo (.TRC).
- 3.º TEMP, para almacenar archivos temporales por ORCAD/VST.
- 4.º DRIVER, para almacenar el resto de drivers necesarios.

Una vez creados estos subdirectorios y convenientemente llenados por los archivos correspondientes (esta operación se realiza de forma automática en la versión educativa), pasaremos a realizar la netlist correspondiente al archivo creado con OrCAD/SDT III. En nuestro caso es el archivo EJEM.SCH de la figura 6.1. (de carácter teórico) y EJEM.NET, el archivo de netlist correspondiente y debidamente depurado.

Copiemos EJEM.NET en el directorio DESIGN de OrCAD/VST.

## Capítulo 6 Utilización práctica de ORCAD/VST



## Avanzando SIMULATE

# Requisitos previos

Situándonos en el directorio apropiado, basta teclear

Suponemos configurado **OrCAD/VST** según los directorios siguientes:

- 1.° **LIBRARY**, para librerías, previamente descompactadas.
- 2.° **DESIGN**, para almacenar los archivos **.NET** creados con ORCAD/SDT III, los ficheros de impulsos (**.STM**) y de muestreo (**.TRC**).
- 3.° **TEMP**, para almacenar archivos temporales por ORCAD/VST.
- 4.° **DRIVER**, para almacenar el resto de drivers necesarios.

Una vez creados estos subdirectorios y convenientemente llenados por los archivos correspondientes (esta operación se realiza de forma automática en la versión educativa), pasaremos a realizar la **netlist** correspondiente al archivo creado con OrCAD/SDT III. En nuestro caso es el archivo **EJEM.SCH** de la figura 6.1. (de carácter teórico) y **EJEM.NET**, el archivo de **netlist** correspondiente y debidamente **depurado**.

Copiemos **EJEM.NET** en el directorio **DESIGN** de OrCAD/VST.

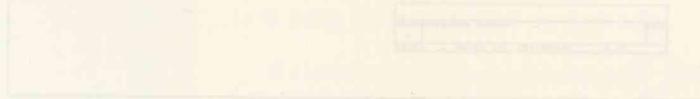


Figura 6.1

Tras unos instantes, aparece el logotipo de OrCAD y a continuación, el Copyright. Tras sendas pulsaciones de tecla, aparece un mensaje solicitando la entrada del archivo de netlist. Si respondemos con <ENTER>, saldremos del programa.

Si no ponemos la extensión **.NET**, **SIMULATE** la pone automáticamente, preguntando previamente si es o no dicha extensión.

A continuación, aparecen de forma fugaz los mensajes "Construyendo modelo de interconexión model" y "Construyendo simulación del modelo". En caso de archivo de netlist erróneo, se produce un mensaje de error y el programa devuelve el control al DOS.



# Arrancando SIMULATE

Situándonos en el directorio apropiado, basta teclear

**SIMULATE** <ENTER>

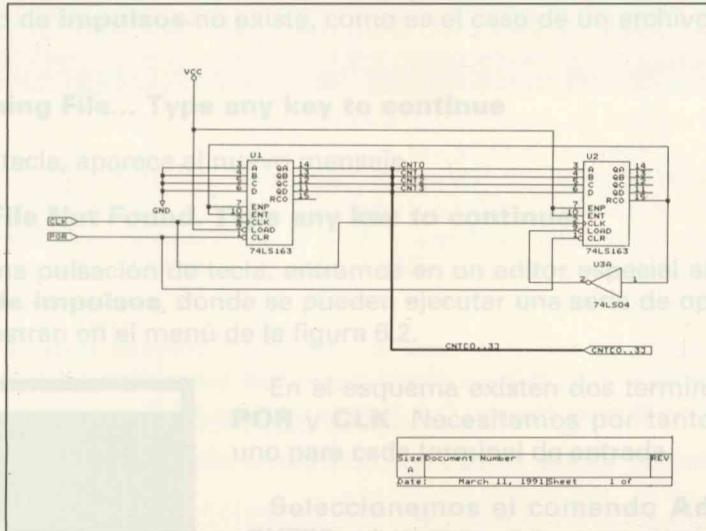


Figura 6.1

Tras unos instantes, aparece el logotipo de OrCAD y a continuación, el Copyright. Tras sendas pulsaciones de tecla, aparece un mensaje solicitando la entrada del archivo de netlist. Si respondemos con <ENTER>, saldremos del programa.

Si no ponemos la extensión **.NET**, **SIMULATE** la pone automáticamente, preguntando previamente si es o no dicha extensión.

A continuación, aparecen de forma fugaz los mensajes "Construyendo modelo de interconexión nodal" y "Construyendo simulación del modelo". En caso de archivo de netlist erróneo, se produce un mensaje de error y el programa devuelve el control al DOS.



## Generación de impulsos

A continuación aparece el mensaje **Enter Stimulus file name?**, que solicita el nombre del archivo de **Impulsos**. Tecleemos el nombre de nuestro ejemplo **EJEM** <ENTER>. OrCAD/VST añade de forma automática la extensión **.STM**.

Si el archivo de **impulsos** no existe, como es el caso de un archivo nuevo, aparece el mensaje

### **Error Opening File... Type any key to continue**

Tras pulsar tecla, aparece el nuevo mensaje

### **Stimulus File Not Found. Type any key to continue.**

Tras la última pulsación de tecla, entramos en un editor especial al que **VST** llama como **editor de impulsos**, donde se pueden ejecutar una serie de operaciones como las que se muestran en el menú de la figura 6.2.

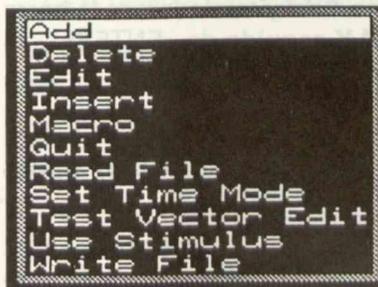


Figura 6.2

co.

En el esquema existen dos terminales de entrada: **POR** y **CLK**. Necesitamos por tanto dos impulsos, uno para cada terminal de entrada.

Seleccionemos el comando **Add**. Tras pulsar <ENTER> dos veces, aparece una pantalla —**stimulus editor**— donde van a ser definidos los impulsos.

Una **barra iluminada**, aparece sobre el ítem **Context**. Esta especificación sólo se utiliza en esquemas de jerarquía por lo que en este caso, quedará en blan-

Desplazemos la **barra iluminada** hasta el campo **Signal Name** presionando la tecla <flecha abajo> o desplazando el ratón.

Para entrar o cambiar el nombre de una señal, hay que usar el subcomando **Edit**. Tras pulsar <ENTER> seleccionamos **Edit Signal Name** con <ENTER> nuevamente.

La barra iluminada se transforma ahora en un pequeño **bloque** sobre el que se halla el cursor.

Introducir el texto **POR** y validar con <ENTER>. A continuación, trasladar la barra iluminada, que aparece nuevamente, hasta el campo **Initial Value**. Aparece un pequeño submenú mostrando los diferentes tipos de valor disponibles.

Seleccionemos el valor **0** con <ENTER> y cuando aparezca de nuevo la barra iluminada presionar <ENTER> nuevamente.

```
Add Delete Macro Return Set
STIMULUS EDITOR
Context
Signal Name : POR
Initial Value: 0
Time         Function
250          1
End Stimulus
```

Figura 6.3

Desplazar la barra iluminada hasta el campo **End Stimulus**. En nuestro ejemplo, consideramos que al cabo de **250** unidades de tiempo el estado de la señal cambia a 1. Para ello, especificar **250** en **Time** y **1** (**o T de Toggle**) en **Fuction**. Con esto, hemos definido el **impulso** correspondiente a la señal **POR**. La figura 6.3 muestra las características de este impulso.

Queremos añadir otro impulso, para lo cual seleccionemos **Return** con <ENTER>. El nombre del impulso es **CLK**. Para añadirlo seleccionemos **Add** y traslademos la barra iluminada al campo **Signal Name**. Con **Edit** tecleemos **CLK** seguido de <ENTER>.

En el campo **Initial Value**, colocamos un **0** como valor inicial.

A continuación, vamos a especificar una transición de **200** unidades de tiempo, para lo cual seleccionamos **Add** nuevamente y ante el interrogante **Time of Function?** respondemos con **200**, seguido de <ENTER>.

Vamos a añadir una nueva transición. Para ello, tras seleccionar **Add** tecleamos **400** seguido de <ENTER> en respuesta al interrogante **Time of Function?** En respuesta a **Function** vamos a generar un lazo con el subcomando **Goto**. Tras pulsar **G** aparece el interrogante **Destination Time of Jump?** al que responderemos con **200**

```
Add Delete Macro Return Set
STIMULUS EDITOR
Context
Signal Name : CLK
Initial Value: 0
Time         Function
200          T
400          JMP 200
End Stimulus
```

Figura 6.4

seguido de <ENTER> con lo que volvemos al inicio de la primera transición. La figura 6.4 muestra la definición de este impulso.

Seleccionar **Return** para volver al menú principal del **editor de impulsos**.

Con el proceso seguido hasta el momento, podíamos dar por finalizada la especificación de impulsos a no ser porque las señales de alimentación (**VCC** y **GND**) presentes en todo esquema, se consideran **impulsos especiales** que necesitan ser especificados, eso sí, al ser señales continuas, no será necesario especificar ningún tipo de transición.

Siguiendo un proceso similar al anteriormente expuesto, utilicemos el comando **Insert** (simplemente para ver su funcionamiento) y traslademos la barra iluminada hasta el campo **Signal Name**. Utilizando **Edit**, teclear **GND** seguido de <ENTER>.

A continuación traslademos la barra iluminada hasta el campo **Initial Value** y seleccionemos un **0**, ya que se trata de una señal de tipo estático. Como no existe transición alguna, la especificación del **impulso GND** queda completada. Seleccionar **Return**.

Sigamos un proceso similar el impulso **VCC**. Teclearemos **VCC** en el campo **Signal Name** y **GND** en el campo **Initial Value**. Seleccionar **Return** para volver al menú principal.

En la versión educacional del programa, estos dos últimos pasos no son necesarios.

El resultado es un archivo, cuya edición se puede efectuar mediante el subcomando **Edit**, se muestra en la figura 6.5.

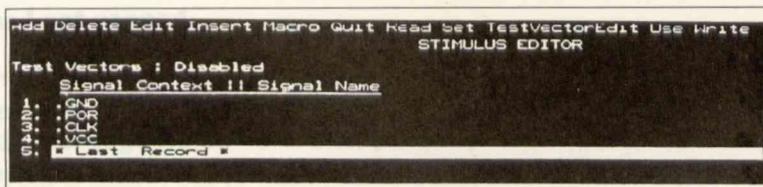


Figura 6.5

Para salvar el archivo de **impulsos**, así creado, seleccionemos el comando **Write**. **Simulate** muestra el nombre (y el path si lo hubiere) del archivo de **impulsos** idéntico al del archivo de **netlist** y da la opción de cambiarlo (**Yes** o **No**).

Para compilar el archivo de **impulsos** y convertirlo en un archivo **reconocible** por **SIMULATE**, utilizaremos la opción Use que detectará los errores que hayan podido ocurrir durante la edición.



## Muestreo

Una vez finalizada la labor correspondiente a la **edición de impulsos**, procedamos a la edición de las señales que se quieran **visualizar** o **muestrear**.

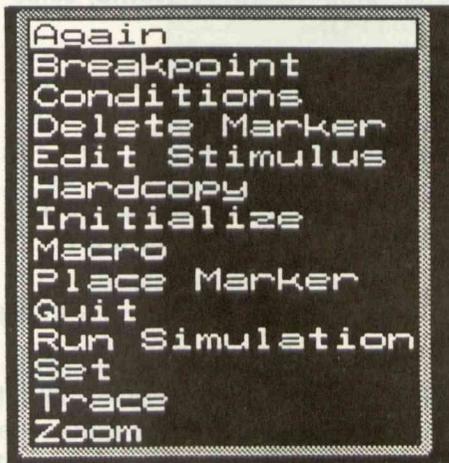


Figura 6.6

Para ello, desde el menú principal (Figura 6.6), seleccionamos el comando **Trace** que va a permitirnos realizar un **muestreo** de las señales, tanto de entrada como de salida, que queremos simular.

Tras seleccionar esta opción, aparece un submenú como el mostrado en la figura 6.7, del cual seleccionamos la opción **Trace Edit**.

Nos encontraremos en el menú principal del editor de muestras (**trace editor**) que, al igual que el editor de impulsos presenta una lista vacía de especificaciones, lo cual se denota

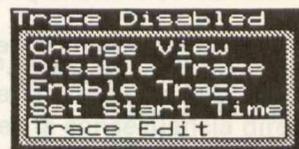


Figura 6.7

por la frase **Last record**.

Seleccionemos el comando **Add** y tras situar la barra iluminada en la posición correspondiente al campo **Display Name**, seleccionemos la opción **Edit** y tecleamos el nombre **RELOJ**, seguido de <ENTER>, para la primera señal. Cualquier otro nombre podría haber sido aceptado ya que no se tiene en cuenta que aparezca o no en el esquema que se desea simular.

De la misma forma, situemos la barra iluminada sobre el campo **Signal Name** y con subcomando **Edit**, tecleamos el nombre de la señal que debe coincidir con la correspondiente al **Module Port** del esquema, en este caso **CLK**. Volver al menú principal del editor con la opción **Return**.

Traslademos la barra iluminada al campo **Type**, que por defecto aparece con la opción **Signal**, una de las varias que se muestran en el submenú de la figura 6.8.

```
Signal
Hex Bus
Octal Bus
Decimal Bus
Binary Bus
Macro
Return
+ = Trace On
- = Trace Off
Macro
Return
```

Figura 6.8

A continuación, situemos la barra iluminada sobre la opción **Trace** a la que pondremos en **ON** y lo mismo haremos con **Display**. Estas dos opciones del menú de edición de **muestra**, pueden cambiarse con las teclas <+> <->.

Pasemos a especificar la siguiente señal que queremos muestrear. Seleccionemos **Add** y tras situar la barra iluminada sobre el campo **Display Name**, seleccionamos **Edit** y tecleemos **POR**.

A continuación, situemos la barra iluminada sobre el campo **Signal Name**. Podíamos introducir sin ningún problema, el nombre de la señal de entrada correspondiente al **Module Port** del esquema (**POR** en este caso), pero también se puede especificar el **nombre de la señal** definiendo el **lugar de ataque**. En nuestro caso, el **pin 1 del integrado U1**. Para especificar esto basta teclear **.U1-1**. Atención al punto que se antepone a **U1**.

Volvamos al menú principal de **SIMULATE** con **Return**.

Hasta ahora hemos definido las dos señales de entrada, pero en un muestreo también hay que especificar las señales de salida. En nuestro ejemplo falta especificar el **bus CNT[0..3]**.

Para ello, sigamos el proceso normal de añadir (**Add**) y colocar la barra iluminada en **Display Name**. A continuación, con **Edit** tecleamos **CNT[0..3]** y validamos con <ENTER>.

Traslademos la barra iluminada al campo **Type** que rellenaremos con la opción **Hexbus** (bus hexadecimal) previa selección de **Edit**.

Los campos **Trace** y **Display** se quedan en **ON**, mientras que el campo **Signal Name** ha sido reemplazado por **Bit Name** debido a la elección de la opción **Bus**.

Traslademos la barra iluminada hasta el campo **Bit Name**, sobre el prompt **Last Item**.

Vamos a añadir cada una de las señales del **bus** asignado un **bit** a cada una de ellas.

Empezaremos por el bit menos significativo (bit 0) al que asociaremos la señal **CNT0** del bus. Para ello utilizemos el comando **Add**, tecleamos **CNT0** y validemos con <ENTER>.

Del mismo modo procederemos para las restantes líneas del bus **CNT1**, **CNT2** y **CNT3** que quedarán asignadas a los bits **1**, **2**, y **3** respectivamente. Tras introducir **CNT** pulsemos la tecla <PgUp> para visualizar los datos introducidos. Corregir si es necesario.

El resultado se muestra en la figura 6.9.

```

Add Delete Edit Insert Macro Return
.....
TRACE EDITOR
Display Name : CNT0..3D
Type         : HexBus
Trace        : ON
Display      : ON
Context      :
Bit          : Name
0           : CNT0
1           : CNT1
2           : CNT2
3           : CNT3
4(000)     : * Last Item *
  
```

Figura 6.9

La especificación del bus **CNT[0..3]** finaliza volviendo al menú principal de **SIMULATE** con el subcomando **Return**.

Una interesante particularidad de **OrCAD/VST** consiste en poder simular las señales presentes en una serie de pines, como si de un bus se tratara, aunque éste no se encuentre especificado como tal en el esquema. Esta ventaja es la consecuencia directa de poder simular señales especificando el número de pin al que se encuentra conectada la señal que se quiere simular.

En efecto, en nuestro esquema, sólo existe un bus (**CNT[0..3]**) compuesto por las cuatro señales de salida **QA..QD** de **U1**.

A continuación, vamos a simular las señales de salida **AQ..AD** de **U2** como si se tratara de un **bus Hexadecimal**.

Para ello, seleccionemos el comando **Add** y situemos la barra iluminada en el campo **Display Name**. Con el subcomando **Edit** introducimos el nombre de este "conjunto de señales" que queremos muestrear: **SALIDAS DE U2**. Validemos con <ENTER>.

Seguidamente, rellenemos el campo **Type** con la opción **HexBus** y dejemos los campos **Trace** y **Display** en **ON**.

Desplazar la barra iluminada hasta la posición marcada como **Last Item** y añadir con **Add**, en orden menor-mayor, las señales siguientes, tal y como se muestra en la figura 6.10.

```

Add Delete Edit Insert Macro Return
TRACE EDITOR
Display Name : SALIDAS U2
Type          : HexBus
Trace         : ON
Display      : ON
Context      :
Bit          Name
0           .U2-14
1           .U2-13
2           .U2-12
3           .U2-11
4           * Last Item *

```

Figura 6.10

El resultado de las señales definidas es el mostrado en la figura 6.11.

```

Add Copy Delete Edit Insert Macro Quit Read Use Write
TRACE EDITOR
Display Name      Type      Trace      Display
1. CLOCK          Signal   ON         ON
2. FOR            Signal   ON         ON
3. CNTC0..3C      HexBus  ON         ON
4. SALIDAS U2     HexBus  ON         ON
5. * Last Record *

```

Figura 6.11

Recordar el presionar **<PgUp>** para ver los datos introducidos y la opción **Return** para volver al menú principal.

Ya en él, seleccionemos el comando **Write** para salvar el archivo creado con este editor.

**SIMULATE** formula la pregunta **Write Trace File?** a lo que respondemos con **EJEM** seguido de **<ENTER>**. Automáticamente se añade la extensión **.TRC**.

De nuevo en el menú principal, seleccionamos el comando **Use** para compilar el archivo **EJEM.TRC**. En caso de existir errores, nos devolverá al editor de muestreo.

## Ejecutando la simulación

Una vez confeccionados y compilados los archivos de **impulsos** y de **muestreo**, hay que pasar a la ejecución de la **simulación**, propiamente dicha.

El comando encargado de efectuar este proceso es **Run Simulation**. Cuando se selecciona aparece el mensaje **Simulation Length?** que nos demanda la duración de la **simulación**.

Tecléemos **500** y validemos con <ENTER>. El resultado es el mostrado en la figura 6.12.

En la presentación de este resultado, el problema reside en que las señales muestreadas no están completas.

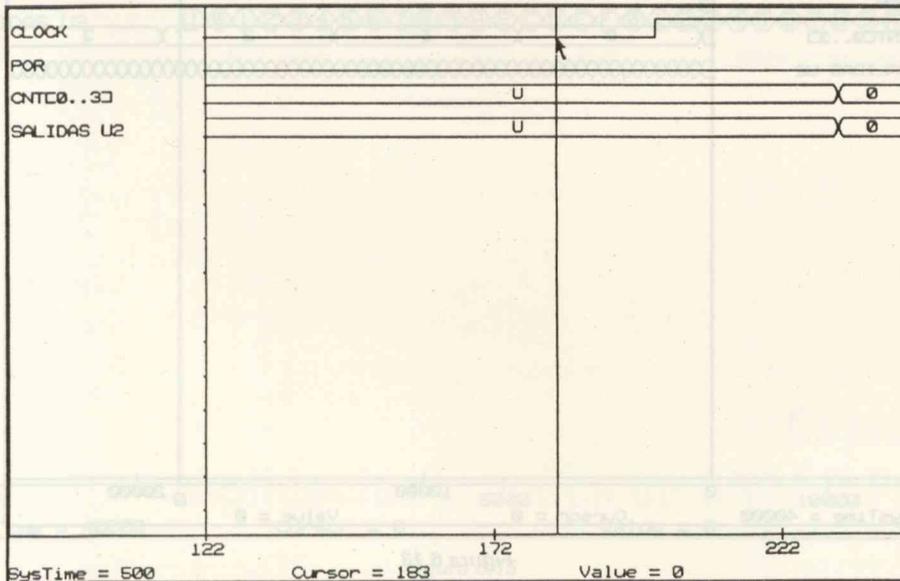


Figura 6.12

**SIMULATE** tiene un reloj interno que puede programarse. Para ello, seleccionemos la opción **Trace** desde el menú principal y de las cuatro opciones que aparecen, seleccionamos la primera: **Change View**.

Aparece el prompt **Trace Delta Time?** al que respondemos con **200**. De esta forma, hemos programado el reloj interno, de forma que la unidad de tiempo que va a utilizar el **simulador** queda multiplicada por **200**.

A continuación, hay que **inicializar** el simulador. Para ello, desde el menú principal, seleccionamos la opción **Initialize**. Tras validar con <ENTER>, aparece un mensaje avisándonos de que el **simulador** va a **arrancar de nuevo desde 0** y demandando nuestra conformidad (**Are you sure?**). Tras seleccionar la opción de **Yes**, el prompt **Systime** que aparece en la parte inferior de la pantalla, se coloca a **0**.

Ahora sólo queda **ejecutar la simulación**, para lo cual seleccionamos el comando **Run Simulation** nuevamente, pero ante la pregunta **Simulation Length?** respondemos con **40000**, ya que hemos aumentado la velocidad del reloj interno.

Tras validar con <ENTER>, aparece una pantalla como la de la figura 6.13.

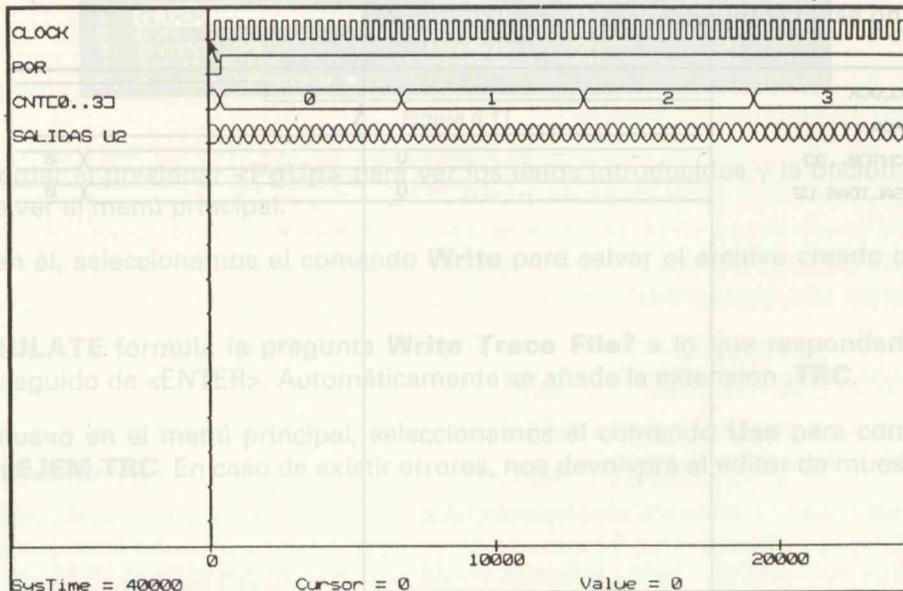


Figura 6.13

Notemos que el **muestreo** correspondiente a **SALIDAS U2**, que recordemos había sido especificado como **bus hexadecimal**, no presenta ninguna **numeración interna** como ocurre con el **bus CNT**.

Ello se debe a que no hay suficiente espacio para que puedan escribirse las numeraciones.

```

Begin at cursor
End at cursor
Center at cursor
Up
Down
Top
Scale
Next page
Previous page
Last page
Initial Display

```

Figura 6.14

El problema se soluciona utilizando el comando **Zoom**. Cuando se selecciona esta opción, aparece un menú como el de la figura 6.14.

Seleccionemos la opción **Scale** y dentro de este submenú, la opción **2**. El resultado obtenido es como el de la figura 6.15, donde se aprecia perfectamente el contaje en hexadecimal que tiene lugar en el **bus**.

**SIMULATE** también incorpora una opción que sirve para medir **intervalos de tiempo**. El comando correspondiente es **Place Marker**.

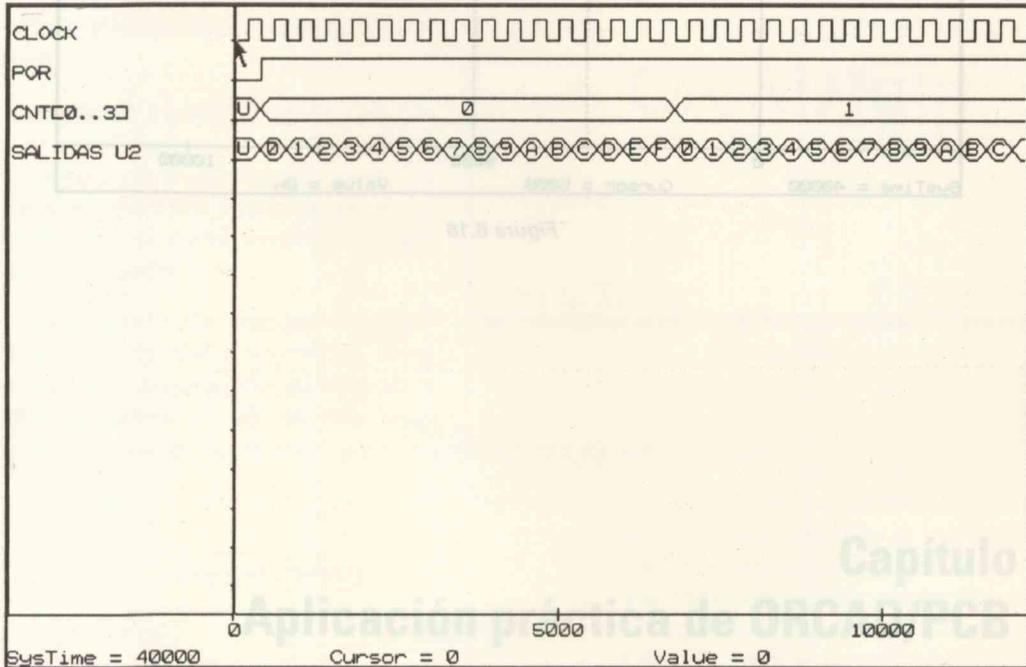


Figura 6.15

Para funcionar con este comando, seleccionar previamente la posición que queramos marcar (por ejemplo **3000**) y, tras presionar <ENTER>, seleccionemos la opción **Place Marker**. Tras validar con <ENTER> y desplazar hasta el **tiempo 5000**, se obtiene un cronograma como el mostrado en la figura 6.16.

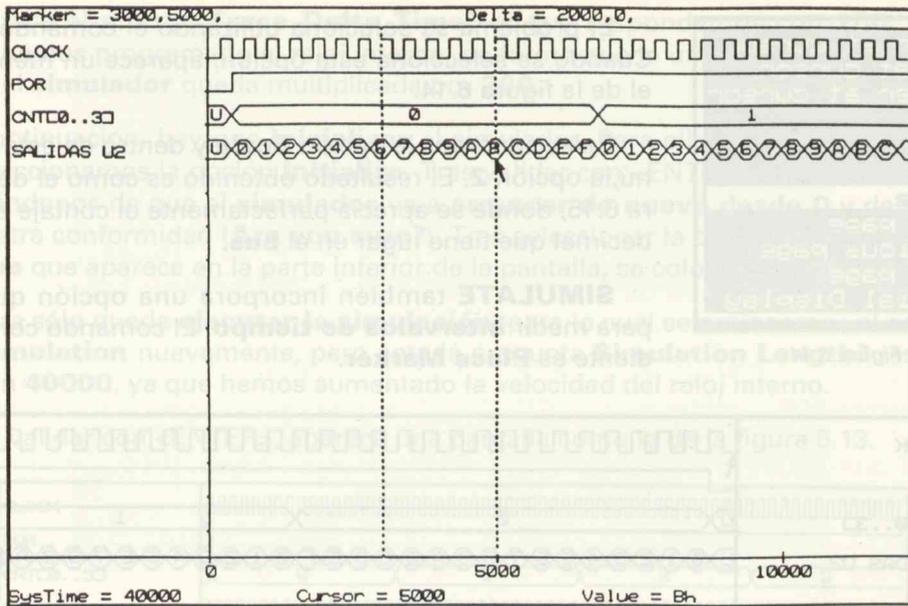


Figura 6.16

## Introducción

El presente trabajo tiene como objetivo principal proporcionar al lector una guía práctica para el uso del módulo OrCAD/PCB II, el cual es el más utilizado dentro del conjunto OrCAD para la realización de los diseños de los circuitos impresos.

Dentro del conjunto OrCAD, el módulo de PCB (acrónimo de PC Board Layout Tools) es utilizado para realizar los diseños de los circuitos impresos.

El objetivo de este trabajo se centra en el aprendizaje del OrCAD PCB siguiendo la realización de un ejercicio práctico.

Para ello, parte de la realización de un

esquema correctamente dibujado y verificado con OrCAD/SDT siguiendo puntualmente todos los pasos necesarios hasta obtener el diseño del circuito impreso totalmente acabado.

No se pretende realizar un estudio profundo del programa, sino facilitar la introducción al mismo a todas aquellas personas interesadas en el diseño electrónico y poco entusiastas de los manuales.



Figura 7.1



Figura 7.2

## Capítulo 7 Aplicación práctica de ORCAD/PCB II

OrCAD es un conjunto de programas complementarios e interrelacionados que abarcan una gran variedad de aplicaciones. En esta sección se describe la relación existente entre los diferentes módulos que forman este programa.

Referente a OrCAD/PCB II, se puede observar que utiliza únicamente la información procedente del módulo OrCAD/SDT.



# Introducción

Dentro del conjunto OrCAD, el módulo de PCB (acrónimo de PC Board Layout Tools) es utilizado para realizar los diseños de los circuitos impresos.

El objetivo de este trabajo se centra en el aprendizaje del OrCAD PCB siguiendo la realización de un ejercicio práctico.

Para ello, parte de la realización de un

esquema correctamente dibujado y verificado con OrCAD/SDT siguiendo puntualmente todos los pasos necesarios hasta obtener el diseño del circuito impreso totalmente acabado.

No se pretende realizar un estudio profundo del programa, sino facilitar la introducción al mismo a todas aquellas personas interesadas en el diseño electrónico y poco entusiastas de los manuales.

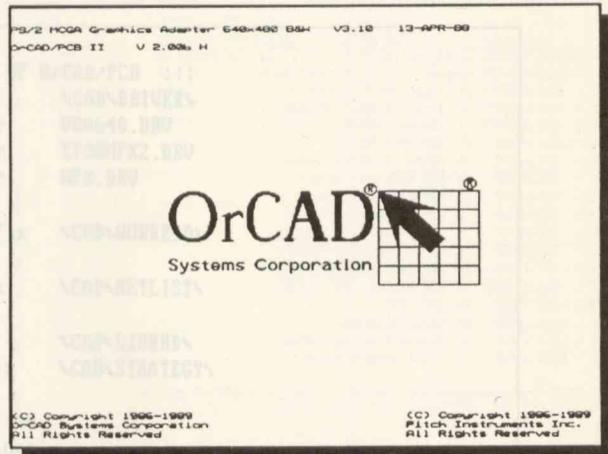


Figura 7.1

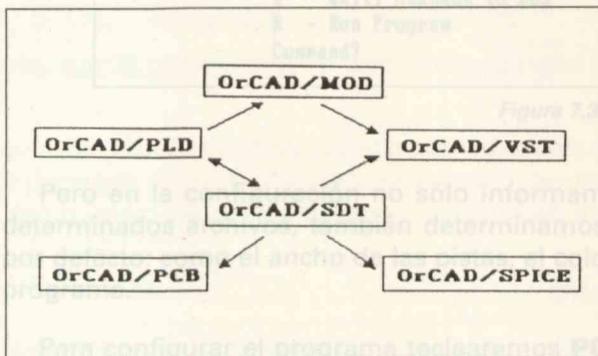


Figura 7.2

OrCAD es un conjunto de programas compatibles e interrelacionados que abarcan los distintos aspectos del diseño electrónico. En la figura 7.2 podemos observar la relación existente entre los diferentes módulos que forman este programa.

Referente a OrCAD/PCB II, se puede observar que utiliza únicamente la información procedente del módulo OrCAD/SDT.

De la **simulación** de los circuitos dibujados con el OrCAD/SDT se encargan dos módulos: El **OrCAD VST** se encarga de la simulación de los circuitos digitales y el **SPICE** de los circuitos analógicos; sin embargo, ambos no están relacionados entre sí.

Dentro del conjunto OrCAD, el módulo de PCB (acrónimo de PC Board Layout Tools) es utilizado para realizar los diseños de los circuitos impresos.



Figura 7.1

OrCAD es un conjunto de programas compatibles e interrelacionados que abarcan los distintos aspectos del diseño electrónico. En la figura 7.2 podemos observar la relación existente entre los diferentes módulos que forman este programa.

Referente a OrCAD/PCB, se puede observar que utiliza únicamente la información procedente del módulo OrCAD/SDT.



Figura 7.2

El objetivo de este trabajo se centra en el aprendizaje del OrCAD PCB siguiendo la realización de un ejercicio práctico.

Para ello, parte de la realización de un

esquema correctamente dibujado y verificado con OrCAD/SDT siguiendo puntualmente todos los pasos necesarios hasta obtener el diseño del circuito impreso totalmente acabado.

No se pretende realizar un estudio profundo del programa, sino facilitar la introducción al mismo a todas aquellas personas interesadas en el diseño electrónico y poco entusiastas de los manuales.

# Configuración

La configuración es la operación inevitable que sigue a la instalación del programa; en ella informamos al programa de la instalación que hemos realizado.

```
OrCAD/PCB :::  
DP - Driver Prefix      \CAD\DRIVER\  
DD - Display Driver    UGA640.DRU  
PD - Printer Driver    EPSONFX2.DRU  
PL - Plotter Driver    HPX.DRU  
  
WP - Work File Prefix  \CAD\WORKBRD\  
  
NP - Netlist Prefix    \CAD\NETLIST\  
  
MP - Module Prefix    \CAD\LIBBRD\  
SP - Strategy Prefix   \CAD\STRATEGY\  
  
MA - Memory Allocation  
DC - Design Conditions  
MC - Met Conditions  
  
CT - Color Table  
PC - Pen Carrousel Configuration  
PW - PhotoPlotter Wheel Configuration  
U  - Update Configuration Information  
Q  - Quit, Abandon to DOS  
R  - Run Program  
Command?
```

Figura 7.3

Pero en la configuración no sólo informamos al programa dónde se encuentran determinados archivos, también determinamos las condiciones de trabajo iniciales o por defecto; como el ancho de las pistas, el color de éstas, etc., cuando entremos en el programa.

Para configurar el programa teclearemos **PCB/C** seguido de "<ENTER>". Al ejecutarse el programa entraremos en el menú de configuración mostrado en la Fig 7.3.

Lo primero que observamos es un grupo de 4 opciones (PD DD PD PL), y se refieren a DRIVER, es decir, ficheros que utiliza el programa para el control del plotter, impresora y placa de gráficos.

### PD - Driver Prefix \CAD\DRIVER\

El comando **PD** sirve para indicar al programa dónde se encuentran los DRIVER, en este caso también se podría haber puesto C:\CAD\DRIVER\.

### DD - Display Driver VGA640.DRV

```

:::Display Driver Configuration:::          Current Selection = VEGA800.DRV
Size Colors Adapter
17 - 640 x 400 16 Sigma 400                SIGMA400.DRV
18 - 640 x 400 16 Tecmar Graphics Master   TGM161.DRV
19 - 640 x 400 1 IBM PS/2 MCGA            MCGA400.DRV
20 - 640 x 400 16 IBM PS/2 VGA            VGA640.DRV
21 - 720 x 200 4 Tecmar Graphics Master   TGM4N.DRV
22 - 720 x 348 1 CPT 9000 Half Screen     CPT9hc.DRV
23 - 720 x 348 1 Hercules Monochrome     HGC2.DRV
24 - 720 x 348 1 Tecmar Graphics Master   TGM2N.DRV
25 - 720 x 400 4 Tecmar Graphics Master   TGM4I.DRV
26 - 720 x 696 1 Tecmar Graphics Master   TGM2I.DRV
27 - 720 x 728 1 CPT 9000 Full screen     CPTFULL.DRV
28 - 768 x 1008 1 MDS Genius / AXTEL 530  GENIUS.DRV
29 - 1024 x 768 16 EIZO B10               GENOALX.DRV
30 - 1024 x 768 16 Genoa SuperVGA        GENOALX.DRV
31 - 1024 x 768 16 Orchid ProDesignerPlus GENOALX.DRV
32 - 1024 x 768 16 Sigma Designs VGA/H   GENOALX.DRV

<<<NOTE>>> More drivers are available using the GENDRIVE utility
M - More
Q - Quit
S - Special (non-listed above)
Selection ->20
  
```

Figura 7.4

Mediante el comando **DD** pasamos al menú de la Fig 7.4, donde podremos seleccionar el DRIVER que corresponda a nuestra tarjeta gráfica. En nuestro ejemplo introducimos el 20 que corresponde a la VGA de 640x480 color y volvemos al menú principal.

En la esquina superior derecha de dicha figura observamos que el driver actualmente seleccionado (current selection) es VEGA800.DRV y que al introducir el 20 será sustituido por VGA640.DRV.

En el supuesto de que no exista el driver apropiado a nuestra placa de gráficos con la utilidad GENDRIVE.EXE podríamos generar nuestro drive en la mayoría de los casos.

Si en la primera página del menú no apareciese nuestro drive pulsando **M** nos irán apareciendo las siguientes páginas.

En el caso que hayamos generado nuestro drive con la utilidad GENDRIVE, el drive no aparecerá en este menú, entonces introduciremos el comando **S** (drive especial) y seguidamente introduciremos el nombre de nuestro drive. Para salir del menú pulsa **Q** (salir).

### PD - Printer Driver EPSONFX2.DRV

Mediante el comando **PD** seleccionamos el drive que corresponda a nuestra impresora o que mejor se adapte a la misma.

Al pulsar PD nos aparece el menú de la Fig 7.5 que corresponde a la primera página de la lista de impresoras que soporta el OrCAD.

En el menú de drivers de impresora podemos observar tres columnas. En la primera aparece el número de selección y el modelo de impresora, en la segunda la resolución de la misma y en la tercera el nombre del **driver** que la controla.

Si no se encontrara nuestro driver entre los primeros pulsando **M** nos irán apareciendo sucesivas páginas.

Puede suceder que no tengamos impresora o que no pensemos usarla con el programa en cuyo caso introducimos **X**.

Dado el tipo de trabajo realizado con el OrCAD PCB la impresora pierde la importancia que tenía en el OrCAD SDT o en el OrCAD VST. Por el contrario, el plotter y fotoplotter resultan fundamentales por su calidad y precisión.

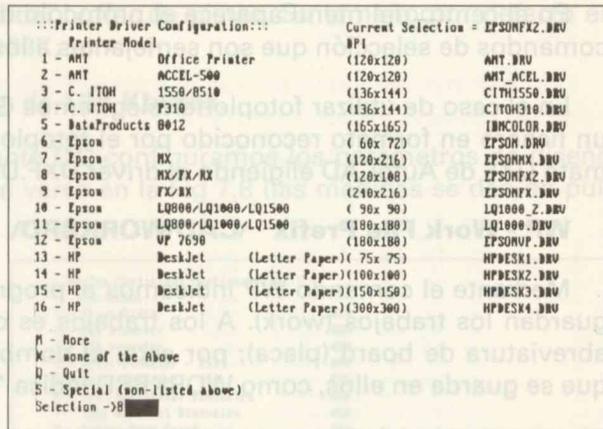


Figura 7.5

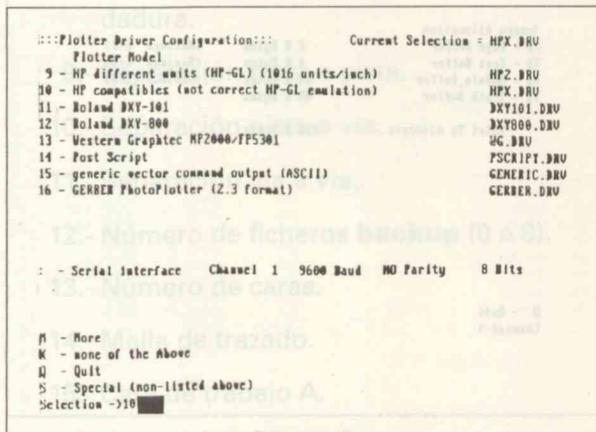


Figura 7.6

## PD - Plotter Driver HPX.DRV

Con **PD** pasamos al menú de selección de driver para plotter, como muestra la Fig 7.6 que al igual que en el caso anterior muestra dos columnas. En la primera vemos el número de selección seguido del modelo de plotter y en la otra columna el driver correspondiente.

En el ángulo superior derecho aparece el drive actualmente seleccionado; en nuestro caso es HPX.DRV que como puede leerse

es un driver para plotters HP compatibles y que no funcionan correctamente con HP.DRV o con HP2.DRV.

En el centro del menú aparece el protocolo de comunicación serie, seguido por los comandos de selección que son semejantes a los ya descritos en otros menús.

En el caso de utilizar fotoplóter elegiremos GERBER.DRV y el OrCAD PCB generará un fichero en formato reconocido por el fotoplóter. También podremos elegir el formato DXF de AutoCAD eligiendo el driver DXF.DRV.

### **WP - Work File Prefix \CADWORKBRD\**

Mediante el comando WP indicamos al programa el subdirectorio (lugar) donde se guardan los trabajos (work). A los trabajos es conveniente añadir la extensión **.BRD** abreviatura de board (placa); por esto, el nombre de los subdirectorios nos indica lo que se guarda en ellos, como WORKBRD indica "trabajos de placas".

**NP - Netlist Prefix \CADNETLIST\** Con el comando NP indicamos al programa donde se encuentran los ficheros generados con la utilidad Netlist del OrCAD SDT y los generados con el subcomando **Report** del OrCAD PCB.

**SP - Strategy Prefix \CAD\STRATEGY\** Mediante este comando indicamos al OrCAD PCB el subdirectorio donde se encuentran los ficheros de estrategias de traza de pistas.

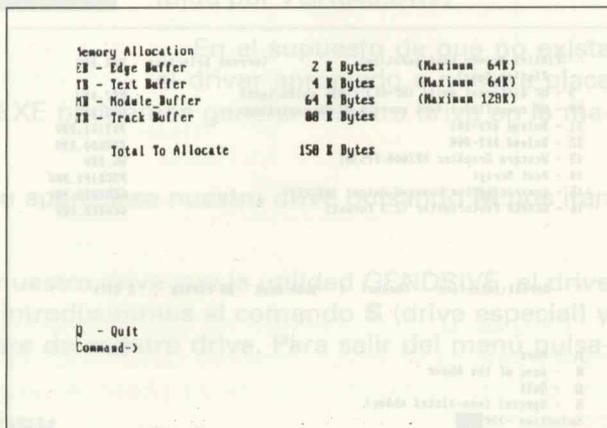
### **MA - Memory Allocation**

Mediante el comando MA configuramos el

tamaño de los cuatro buffers mas importantes que intervienen en el diseño del circuito impreso.

Como puede verse en la Fig 7.7 con **EB** podemos variar el tamaño del buffer de líneas del borde del C.I. Su tamaño inicial es de 2K.

Con **TB** variamos el tamaño del buffer de textos. Su tamaño por defecto es de 4K.



Memory Allocation		
EB - Edge Buffer	2 K Bytes	(Maximum 64K)
TB - Text Buffer	4 K Bytes	(Maximum 64K)
MB - Node Buffer	64 K Bytes	(Maximum 128K)
TR - Track Buffer	80 K Bytes	
Total To Allocate	150 K Bytes	

Q - Quit  
Command->

Figura 7.7

Pulsando **MB** modificamos el tamaño del buffer de módulos (componentes PCB). Su tamaño inicial es de 64K y admite un máximo de 128K.

Con **TR** podemos variar el tamaño del buffer de pistas. Su tamaño por defecto es de 88K bytes y llega a un máximo de 128K.

El **Total de Memoria Asignada** es de **128 Kbytes**.

**DC - Design Conditions** Mediante **DC** configuramos los parámetros de diseño iniciales más importantes como puede verse en la Fig 7.8 (las medidas se dan en pulgadas):

- 1.- Ancho de pista.
- 2.- Diámetro del pad.
- 3.- Diámetro de via.
- 4.- Diámetro del taladro del pad.
- 5.- Diámetro del taladro de la via.
- 6.- Anchura del texto.
- 7.- Altura del texto.
- 8.- Protección máscara de soldadura.

:::Design Conditions Configuration:::		
Parameter		
1 - Track Width	.014	
2 - Pad Diameter	.000	
3 - VIA Diameter	.050	
4 - Drill Diameter - Pad	.040	
5 - Drill Diameter - VIA	.035	
6 - Text Horizontal Dimension	.050	
7 - Text Vertical Dimension	.050	
8 - Solder Mask Guard	.020	
9 - Isolation Track to Track	.013	
10 - Isolation Track to VIA	.013	
11 - Isolation VIA to VIA	.020	
12 - Number of Backup Files	2	
13 - Number of Layers	2	
14 - Routing Grid	.050	
15 - Working Layer A	1	
16 - Working Layer B	2	
17 - Strategy Pass 1	Normal	
18 - Strategy Pass 2	Extensive	
19 - Net Pattern	Tree	
20 - Cursor Style	Short Cross	
Q - Quit		
Selection ->		

Figura 7.8

- 9.- Separación pista a pista.
- 10.- Separación pista a via.
- 11.- Separación via a via.
- 12.- Número de ficheros **backup** (0 a 8).
- 13.- Número de caras.
- 14.- Malla de trazado.
- 15.- Cara de trabajo A.
- 16.- Cara de trabajo B.
- 17.- Estrategia en el primer pase.

18.- Estrategia en el segundo pase.

19.- Net pauta (norma).

20.- Forma del cursor.

Una vez que hayamos seleccionado el nº de la opción, modificaremos el parámetro y con **Q** regresaremos al menú principal.

```

:::Net Conditions Configuration:::
  A      B      C      D      E      F      G      H      I      J      K
  Net Name Width Via  ---Isolation--- Layers ---Strategy--- Net
1 -      .015  .045  .015  .015  .020  1  1  Normal Normal Tree
2 - SIMPLE .020  .045  .010  .015  .020  1  1  One Side Extensive Tree
3 -      .015  .045  .015  .015  .020  1  2  Normal Normal Tree
4 -      .015  .045  .015  .015  .020  1  2  Normal Normal Tree
5 -      .015  .045  .015  .015  .020  1  2  Normal Normal Tree
6 -      .015  .045  .015  .015  .020  1  2  Normal Normal Tree
7 -      .015  .045  .015  .015  .020  1  2  Normal Normal Tree
8 -      .015  .045  .015  .015  .020  1  2  Normal Normal Tree
9 -      .015  .045  .015  .015  .020  1  2  Normal Normal Tree
10 -     .015  .045  .015  .015  .020  1  2  Normal Normal Tree
11 -     .015  .045  .015  .015  .020  1  2  Normal Normal Tree
12 -     .015  .045  .015  .015  .020  1  2  Normal Normal Tree
13 -     .015  .045  .015  .015  .020  1  2  Normal Normal Tree
14 -     .015  .045  .015  .015  .020  1  2  Normal Normal Tree
15 -     .015  .045  .015  .015  .020  1  2  Normal Normal Tree
16 -     .015  .045  .015  .015  .020  1  2  Normal Normal Tree

R - Quit
Command->

```

Figura 7.9

## NC - Net conditions

Teclando **NC** accedemos al menú de la Fig 7.9; en dicho menú podemos asignar a una o varias **nets** las condiciones de diseño adecuadas.

En nuestro ejemplo veremos que la NET "simple" se le asigna las siguientes condiciones de diseño: ancho de pista de 0.20 in, diámetro de vía de 0.45 in, separación mínima pista\_pista 0.01 in, pista\_vía de 0.15 in y vía\_vía 0.02 in; una sola cara; estrategias **one side** y **extensiva**; net pattern **tree** (ramificada).

## CT - Color Table

Accedemos al menú de la Fig 7.10 introduciendo **CT** y en él podemos modificar los colores de lo que aparezca en la hoja de trabajo, como pistas, textos, componentes, etc.

En el menú observamos tres columnas; en la primera aparece el número de selección y el concepto, en la segunda aparece el color seleccionado, y en la tercera vemos un número de selección y su color correspondiente, existiendo un total de 16 colores.

```

::: Color Configuration :::
1 - Grid Bots    LIGHT GRAY      0 - BLACK
2 - Nets Nest    YELLOW          1 - BLUE
3 - Force Vector LIGHT RED        2 - GREEN
4 - Pad (Component) GREEN           3 - CYAN
5 - Pad (Soldier) BLUE            4 - RED
6 - Outline      BLUE            5 - MAGENTA
7 - Edge         MAGENTA        6 - BROWN
8 - Module Name LIGHT GREEN     7 - DARK GRAY
9 - Text         LIGHT MAGENTA  8 - LIGHT GRAY
10 - Anchor      LIGHT BLUE     9 - LIGHT BLUE
11 - Via Through DARK GRAY     10 - LIGHT GREEN
12 - Via Blind   CYAN          11 - LIGHT CYAN
13 - Via Buried MAGENTA       12 - LIGHT RED
14 - Background BLACK          13 - LIGHT MAGENTA
15 - Layer 1    RED           14 - YELLOW
16 - Layer 2    GREEN         15 - WHITE

C - Color select
M - More
Q - Quit
Command->

```

Figura 7.10

Para modificar el color teclearemos **C** e introduciremos el número del concepto a modificar y el número correspondiente del color de la paleta de 16 colores. Por ejemplo, para cambiar el color del texto de magenta claro a marrón teclearemos: **C** luego el nº **9** y después el nº **6**.

Pulsando **M** accedemos a la segunda parte del menú. Para salir pulsamos **Q**.

### PC - Pen Carrousel Configuration

En este menú podemos cambiar los parámetros de ploteado, ver la Fig 7.11.

Podemos ver que hay cuatro columnas; en la primera aparece el número de la plumilla, en la segunda la anchura, en la tercera su velocidad y en la cuarta su aceleración.

Por ejemplo para cambiar la anchura de una plumilla tecleamos **W** y a continuación el nº de la plumilla que necesitamos variar (1ª columna) seguido de <ENTER>; seguidamente introduciremos el nuevo ancho en pulgadas.

La velocidad y aceleración se cambian de igual manera pulsando **V** y **A** respectivamente.

```

:::Pen Carrousel Configuration:::
Pen Width Velocity Acceleration
1 - .010 DEFAULT DEFAULT
2 - .010 DEFAULT DEFAULT
3 - .010 DEFAULT DEFAULT
4 - .010 DEFAULT DEFAULT
5 - .010 DEFAULT DEFAULT
6 - .010 DEFAULT DEFAULT
7 - .010 DEFAULT DEFAULT
8 - .010 DEFAULT DEFAULT
9 - .010 DEFAULT DEFAULT
10 - .010 DEFAULT DEFAULT
11 - .010 DEFAULT DEFAULT
12 - .010 DEFAULT DEFAULT
13 - .010 DEFAULT DEFAULT
14 - .010 DEFAULT DEFAULT
15 - .010 DEFAULT DEFAULT
16 - .010 DEFAULT DEFAULT
V - X Paper Offset 0.000
V - Y Paper Offset 0.000
W - Pen Width select
V - Velocity select
A - Acceleration select
Q - Quit
Command->
  
```

Figura 7.11

```

:::PhotoPlotter Wheel Configuration:::
Aperture Vertical Horizontal Type
1 - 0.000 0.000 LINE
2 - 0.000 0.000 any
3 - 0.000 0.000 any
4 - 0.000 0.000 any
5 - 0.000 0.000 any
6 - 0.000 0.000 any
7 - 0.000 0.000 any
8 - 0.000 0.000 any
9 - 0.000 0.000 any
10 - 0.000 0.000 any
11 - 0.000 0.000 any
12 - 0.000 0.000 any
13 - 0.000 0.000 any
14 - 0.000 0.000 any
15 - 0.000 0.000 any
16 - 0.000 0.000 any
V - Vertical Dimension
H - Horizontal Dimension
T - Type
M - None
L - Load Tool File W - Write Tool Table
Q - Quit C - Clear Tool Table
Command->
  
```

Figura 7.12

### PW - PhotoPlotter Wheel Configuration.

En este menú cambiamos los parámetros de salida para fotoplóter.

Como vemos en la Fig 7.12, la primera de las cuatro columnas nos muestra el nº de la apertura; en la siguiente, la dimensión de la apertura vertical (se cambia con **V**); la siguiente columna muestra el valor de la apertura horizontal (pulsando **H** se cambia); y en la última columna aparece el tipo de apertura, pul-

ra, pulsando **T** aparecen los tipos a seleccionar. Las medidas se dan en pulgadas; siendo el 0 el valor por defecto.

### U - Update Configuration Information.

Pulsando **U** salvamos en el disco los cambios realizados en la configuración. Si salimos del menú de configuración sin salvar los cambios los perderemos al salir del programa.

### Q - Quit, Abandon to Dos.

Si no se han realizado modificaciones de la configuración, pulsando **Q** regresamos al DOS directamente. Si hubiésemos efectuado algún cambio, al pulsar **Q** nos aparece un mensaje en donde se nos pregunta si queremos salvar los cambios (y/n). Pulsando **y** estos se salvan antes de ir al **DOS**; pero si no contestamos afirmativamente éstos se perderán.

### R - Run Program.

Esta opción es similar a la anterior, con la diferencia de que en vez de salir al DOS entramos en el OrCAD/PCB.

Al pulsar **R** aparece el mensaje advirtiéndonos si queremos salvar los cambios (y/n); pulsando **Y** los salvamos permanentemente (en disco) entrando en el programa; pulsando **N** los cambios no se salvan, pero si se reflejan al entrar en el programa, perdiéndose al salir de él.

The screenshot shows a configuration menu with the following options and values:

Option	Value
1 -	1.000
2 -	1.000
3 -	1.000
4 -	1.000
5 -	1.000
6 -	1.000
7 -	1.000
8 -	1.000
9 -	1.000
10 -	1.000
11 -	1.000
12 -	1.000
13 -	1.000
14 -	1.000
15 -	1.000
16 -	1.000
17 -	1.000
18 -	1.000
19 -	1.000
20 -	1.000
21 -	1.000
22 -	1.000
23 -	1.000
24 -	1.000
25 -	1.000
26 -	1.000
27 -	1.000
28 -	1.000
29 -	1.000
30 -	1.000
31 -	1.000
32 -	1.000
33 -	1.000
34 -	1.000
35 -	1.000
36 -	1.000
37 -	1.000
38 -	1.000
39 -	1.000
40 -	1.000
41 -	1.000
42 -	1.000
43 -	1.000
44 -	1.000
45 -	1.000
46 -	1.000
47 -	1.000
48 -	1.000
49 -	1.000
50 -	1.000
51 -	1.000
52 -	1.000
53 -	1.000
54 -	1.000
55 -	1.000
56 -	1.000
57 -	1.000
58 -	1.000
59 -	1.000
60 -	1.000
61 -	1.000
62 -	1.000
63 -	1.000
64 -	1.000
65 -	1.000
66 -	1.000
67 -	1.000
68 -	1.000
69 -	1.000
70 -	1.000
71 -	1.000
72 -	1.000
73 -	1.000
74 -	1.000
75 -	1.000
76 -	1.000
77 -	1.000
78 -	1.000
79 -	1.000
80 -	1.000
81 -	1.000
82 -	1.000
83 -	1.000
84 -	1.000
85 -	1.000
86 -	1.000
87 -	1.000
88 -	1.000
89 -	1.000
90 -	1.000
91 -	1.000
92 -	1.000
93 -	1.000
94 -	1.000
95 -	1.000
96 -	1.000
97 -	1.000
98 -	1.000
99 -	1.000
100 -	1.000

## El comando Set

Al entrar en el programa nos aparece la configuración por defecto que nosotros hemos determinado en el menú de **configuración**. En determinados casos puede interesarnos modificar alguna condición de trabajo desde dentro del programa; es decir, sobre la marcha. Utilizando el comando **SET** del menú general podremos realizar bastantes modificaciones a las condiciones iniciales.

En la figura 7.13 vemos el comando **SET** del menú general si pulsamos el botón izquierdo del ratón o la letra **S** entramos en el menú de SET. como muestra la figura 7.14.

Hay que tener en cuenta que todos los cambios que se realicen desde el menú de SET no son permanentes y se perderán cuando salgamos del programa.

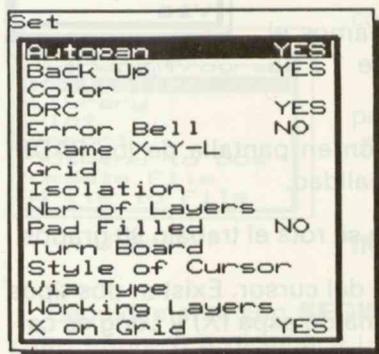


Figura 7.14

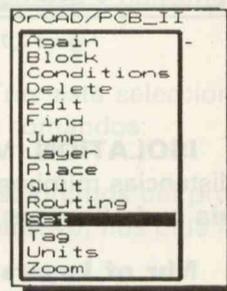


Figura 7.13

En el menú de SET el primer subcomando que nos aparece es **AUTOPAN** y a su derecha podemos ver si se encuentra activo o no (yes/no). En el caso de estar activo nos permitirá pasar de una pantalla a otra cuando nos movamos con el cursor y toquemos un borde de la hoja de trabajo.

El subcomando **BACK UP** nos crea, en el caso de estar activo, unos ficheros de seguridad cuando accedamos al menú de PLACE o ROUTING. El número de copias de seguridad se determina en el menú de configuración.

Mediante el subcomando **COLOR** variamos el color de los **elementos** que aparecen en pantalla (Figuras 7.15 y 7.16), como la malla, los módulos, las caras, los textos, etc.

Si, por ejemplo, quisiéramos variar el color de la malla, ver **figura 15**, tendríamos que acceder en el menú de COLOR a **GRID** (figura 7.16), seguidamente se mostrará un menú de 16 colores a elegir, y seleccionamos el color **Light Grey**. Es necesario, a veces, probar distintos colores hasta encontrar la combinación más apropiada.

Set	
Autopan	YES
Back Up	YES
Color	
DRC	YES
Error Bell	NO
Frame X-Y-L	
Grid	
Isolation	
Nbr of Layers	
Pad Filled	NO
Turn Board	
Style of Cursor	
Via Type	
Working Layers	
X on Grid	YES

Figura 7.15

**ISOLATION.** Mediante este subcomando determinamos las distancias mínimas de **pista a pista, entre pista y via, 2y de via a via** (la via es un paso de cara).

**Nbr of Layers.** Mediante este subcomando determinamos el nº de caras que va a tener el circuito impreso. Admite de 1 a 16 caras.

**Pad Filled.** Este subcomando modifica la presentación en pantalla de los PADS, mostrando los rellenos y con taladro, como serían en la realidad.

**Turn Board.** Cada vez que se ejecuta este subcomando se rota el trabajo 90 grados.

**Style of Cursor.** En este menú modificamos la forma del cursor. Existen dos tipos **Short y Long**; el tipo Short es un cursor pequeño en forma de aspa (X) y Long es cartesiano como el de autoCAD.

**Working Layer.** Como el programa sólo trabaja con dos caras simultáneamente, de las 16 que permite, podemos determinar en este menú cuáles son esas dos caras.

**X on GRID.** Si este subcomando se encuentra activo (yes) el cursor transcurrirá por los puntos de la malla, así como las pistas, y los componentes estarán alineados con ella.

En ciertos momentos puede interesar trabajar fuera de la malla, como en el trazado de pistas entre componentes no normalizados u optimizar la colocación de referencias y valores de componentes, etc.

**DRC** si está activo realiza un chequeo del C. Impreso para detectar las violaciones o errores que pudieran producirse. Al final del diseño podemos generar un informe en ASCII mediante la siguiente secuencia de comandos: **QUIT-REPORT-Design Rule Check.**

**ERROR BELL** es un subcomando que cuando se encuentra activo (yes) el programa nos advertirá de determinadas equivocaciones mediante un pitido.

**FRAME X-Y-L** se utiliza para cambiar de lugar el indicador de cordenadas x-y del cursor en pantalla.

**Grid** sirve para cambiar el paso de la malla de trabajo.

Item
Grid
Background
Module
Edge
Ratsnest
Text
Layer
Via

Figura 7.16

# El comando QUIT

El menú de **QUIT** es uno de los más utilizados y por tanto de gran importancia. También su complejidad es grande por el gran número de subcomandos y menús que engloba.

Cuando se entra en el programa nos aparece el siguiente mensaje en pantalla: **Load Board?** si queremos recuperar un trabajo escribiremos su **nombre** y pulsamos "return", para empezar uno nuevo sólo pulsamos "return".

Desde el **Menú Principal** seleccionamos el comando **QUIT**. Tras esta selección, aparece un nuevo submenú (figura 7.17) que muestra los siguientes comandos:

**ABANDON PROGRAM.** Si ejecutamos este subcomando nos saldremos del programa y el trabajo se perderá. Por eso, antes de salir al sistema operativo, nos pide la confirmación (**NO/YES**).

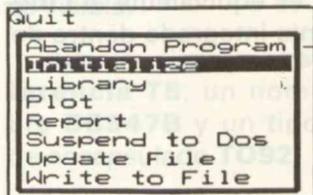


Figura 7.17

**INITIALIZE.** Con este subcomando inicializamos total o parcialmente el programa; en el primer caso se borrará el diseño y en el segundo una parte del mismo.

El proceso de trabajo a seguir es el siguiente: **QUIT - INITIALIZE - USE NETLIST.**

Seguidamente con **BEGIN** abrimos una ventana del tamaño, aproximado, del circuito impreso a diseñar y la cerramos con **END**.

Una vez hayamos definido la zona donde aparecerán los componentes, **teclearemos el nombre del fichero FUENTE.NET** (ver figura 7.18).

Observemos cómo van apareciendo los componentes con sus

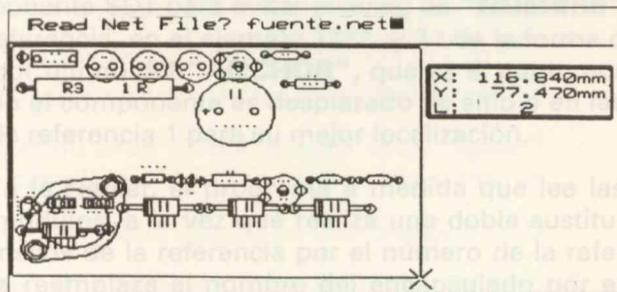


Figura 7.18

referencias y valores. Los componentes que no existan en la librería serán ignorados, pero siempre podemos crearlos y volver a repetir el proceso.

Realizadas las operaciones anteriormente descritas saldremos al **Menú General** para acceder al menú de colocar los componentes **PLACE**.

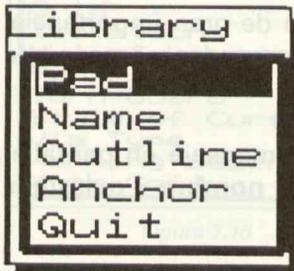
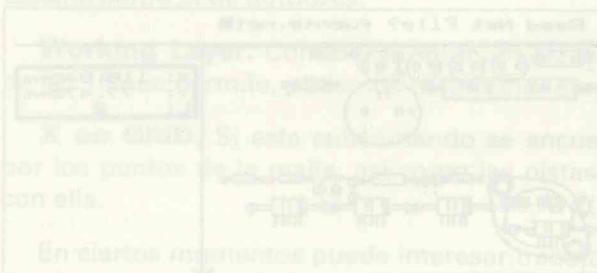


Figura 7.19

En este mismo menú tenemos otros 7 subcomandos como: **LOAD BOARD** que nos permite recuperar un diseño salvado en disco. **CLEAR BOARD** se utiliza para borrar totalmente el diseño. **ERASE ROUTES** borra las pistas del diseño. **RESTORE BACKUP** restaura la última copia de seguridad realizada. Es muy útil cuando por un fallo eléctrico perdemos el trabajo y no lo habíamos salvado o actualizado previamente. **GERBER FILE, HP-GL FILE y DM-PL FILE** son opciones de visualización, que nos permiten ver ficheros almacenados en disco en formato GERBER, HP o DM.

**LIBRARY.** En el menú de library es donde se crean y modifican los componentes o **módulos**; el menú de library consta de los siguientes submenús: **PAD NAME OUTLINE ANCHOR Y QUIT.** Ver fig.19. El menú de **Library** es equivalente al **Libedit.exe** del **OrCAD SDT**, pero con la ventaja de estar totalmente integrado dentro del **PCB.EXE.**



## Diseño de componentes

Desde OrCAD PCB, los componentes se crean o modifican dentro del mismo programa y sin necesidad de abandonar el trabajo que se esté realizando; para ello basta con entrar al menú de **LIBRARY** crear o modificar el componente y salir de **LIBRARY**.

En la Figura 7.20 podemos ver un componente del OrCAD SDT, un transistor, y las partes que lo constituyen. Vemos que consta de un **cuerpo** (símbolo del componente) y que sus terminales (B C E) tienen un número que no es visible.

Además, tiene una **referencia T5**, un nombre **BC547B** y un tipo de encapsulado **TO92**.

Los componentes o **módulos** en el OrCAD PCB (ver Figura 7.21),

constan de un cuerpo "**OUTLINE**"; de unos "**PAD**" cuyo n.º tiene que corresponderse con el terminal homólogo del componente SDT para evitar errores; de "**NOMBRE**" que consta de dos partes: 1.ª de una referencia, en el ejemplo T\*\*\*, y 2.ª de la forma o encapsulado, en el ejemplo **TO92**; y por último del "**ANCHOR**", que es el punto por donde es sujetado por el cursor cuando el componente es desplazado de sitio o en las rotaciones, suele colocarse en el pad de referencia 1 para su mejor localización.

Cuando inicializamos y llamamos a la Netlist, el programa a medida que lee las NET va extrayendo de la librería los módulos, a la vez que realiza una doble sustitución; de una parte reemplaza los asteriscos de la referencia por el número de la referencia del componente SDT y de otra reemplaza el nombre del encapsulado por el nombre del componente SDT, el número **5** y **BC547B** respectivamente.

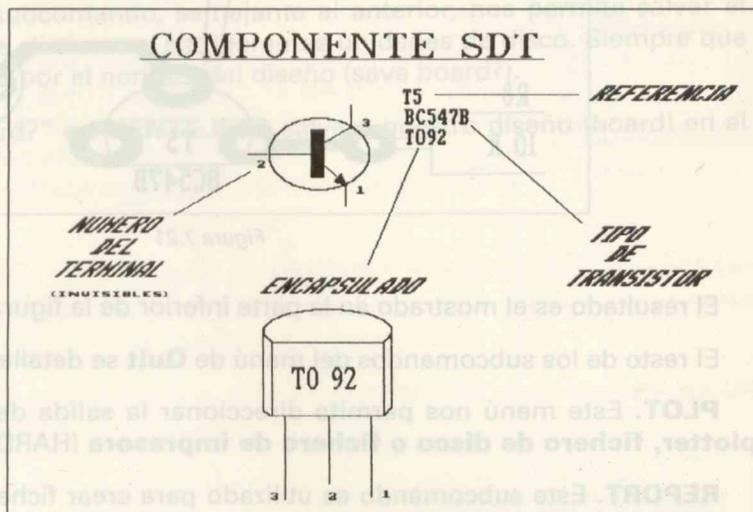


Figura 7.20

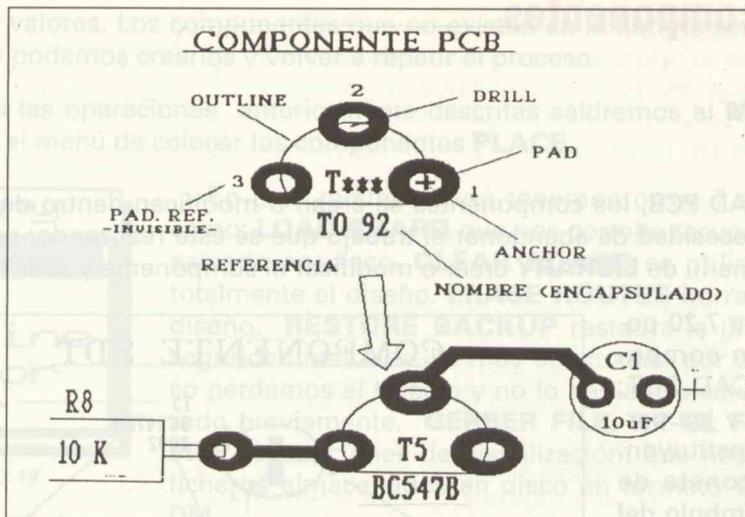


Figura 7.21

El resultado es el mostrado en la parte inferior de la figura 7.21.

El resto de los subcomandos del menú de **Quit** se detalla a continuación.

**PLOT.** Este menú nos permite direccionar la salida de nuestro diseño hacia un **plotter, fichero de disco o fichero de impresora (HARDCOPY)**.

**REPORT.** Este subcomando es utilizado para crear ficheros en ASCII en disco. Estos ficheros nos suministran información referente al diseño que hemos realizado.

Accediendo a este menú nos encontramos con cuatro opciones:

**Netlist** abre un fichero ASCII con el listado de conexiones y en el caso de que ya exista lo actualiza. **Esto ocurre cuando la netlist ha sido modificada con el comando EDIT.**

**Gerber Tool** da un listado de las aperturas para fotoplotter Gerber.

**Design Rule Check (DRC)** abre un fichero que contiene información de los errores o violaciones del diseño realizado.

**Drill Report** abre un fichero informativo referente a los taladros del diseño.

**SUSPEND TO DOS.** Mediante este subcomando podemos **salir al sistema operativo (DOS)** sin abandonar el programa y por tanto sin perder el trabajo en curso. Es

de resaltar que al quedar residente el programa deja muy poca memoria libre y sólo se ejecutaran programas pequeños o comandos DOS residentes.

Para regresar al programa hay que teclear EXIT y pulsar "return".

**UPDATE FILE.** Se utiliza este subcomando para salvar y actualizar el diseño en el disco. Cuando lo utilizemos por vez primera, nos preguntará el nombre del fichero con el que **se salvará** el diseño, **en nuestro ejemplo: FUENTE.BRD**, y en sucesivas veces **se actualizará** automáticamente cada vez que utilizemos el subcomando.

**WRITE TO FILE.** Este subcomando, semejante al anterior, nos permite salvar el mismo trabajo con nombres distintos o en diferentes unidades de disco. Siempre que lo utilizemos nos preguntará por el nombre del diseño (save board?).

Por ejemplo: "Save Board?" **a:FUENTE.BRD** salvará nuestro diseño (board) en el disco de la unidad A.



Figura 7.23

Para mover un componente tenemos que posicionar el cursor mediante el ratón en el componente y pulsar el botón izquierdo. En ese momento el componente quedará comp suspendido del cursor (ver figura 7.25), por el Anchor y así podrá ser trasladado. rotado con **ROTATE**, hasta posicionarlo en otro lugar con **place**.

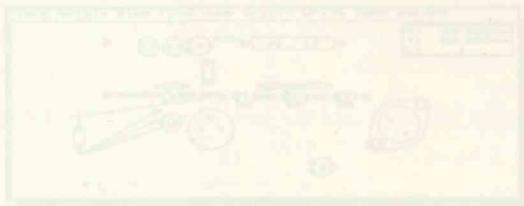


Figura 7.25

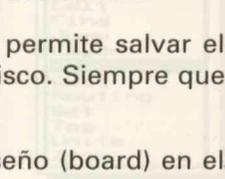


Figura 7.22

Si accedemos al menú de **MODULE** podremos observar un conjunto de subcomandos que nos van a permitir realizar operaciones relacionadas con los módulos (ver figura 7.24).

En primer lugar con el subcomando **MOVE** podremos mover los componentes o módulos.

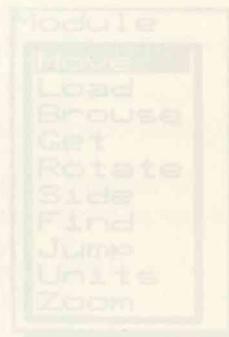


Figura 7.24

Mediante el subcomando **LOAD** podemos traer componentes directamente de la librería al diseño, para ello tendremos que teclear su nombre, lo que implica saberlo de antemano. En nuestro caso tendremos que sacar un dissipador para el



## El menú de PLACE

Es uno de los más importantes por ser, junto con **ROUTING**, de los más utilizados en el diseño de la placa de circuito impreso.

A él se accede desde el menú principal, iluminando la opción correspondiente, tal y como indica la figura 7.22.

Tras seleccionar dicha opción, aparece el submenú mostrado en la figura 7.23, donde existen cuatro opciones con las cuales realizamos la colocación o distribución de los componentes, pondremos algunos componentes especiales (disipadores, etc), delimitaremos zonas, escribiremos textos y trazaremos los bordes del futuro circuito impreso.

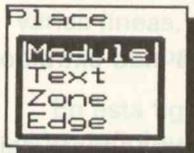


Figura 7.23

Si accedemos al menú de **MODULE** podremos observar un conjunto de subcomandos que nos van a permitir realizar operaciones relacionadas con los módulos (ver figura 7.24).

En primer lugar con el subcomando **MOVE** podremos mover los componentes o módulos.

Para mover un componente tenemos que posicionar el cursor mediante el ratón en el componente y pulsar el botón izquierdo. En ese momento el componente quedará como suspendido del cursor (ver figura 7.25), por el **Anchor** y así podrá ser trasladado, rotado con **ROTATE**, hasta posicionarlo en otro lugar con **place**.

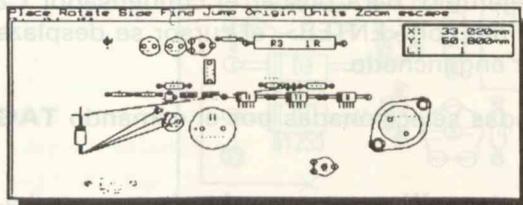


Figura 7.25

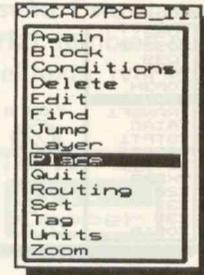


Figura 7.22

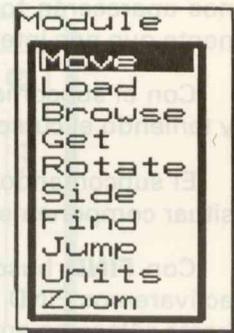


Figura 7.24

Mediante el subcomando **LOAD** podemos traer componentes directamente de la librería al diseño, para ello tendremos que teclear su nombre, lo que implica saberlo de antemano. En nuestro caso tendremos que sacar un disipador para el

transistor de potencia y 4 tornillos para la sujeción de la placa. Estos componentes son considerados especiales por no llevar conexión eléctrica alguna y por tanto no pueden figurar en la netlist generada por **el OrCAD SDT**. Como consecuencia de ello, **al capturar la netlist** con USE NETLIST no aparecerán estos componentes, por lo que habrá que **cargarlos** aparte.

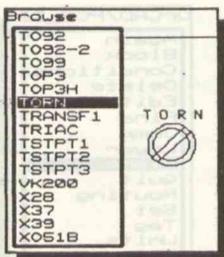


Figura 7.27

Como ejemplo, dentro de nuestro diseño, vamos a cargar un disipador de 56x56, ver **fig.26**, para ello seleccionamos **LOAD** y posteriormente introducimos el nombre del componente: **DS56X562** y nos aparecerá el componente que se ve en la figura 7.26.

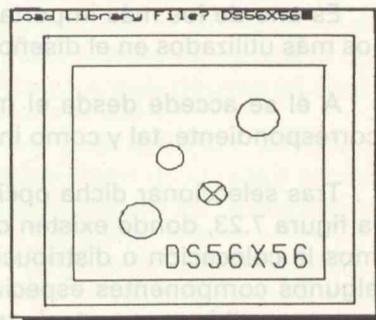


Figura 7.26

Con el subcomando **BROWSE** accedemos directamente a la librería y elegimos el componente de una ventana móvil.

En la figura 7.27, se observa un componente **TORN** (tornillo) que ha sido extraído de la librería a través del comando **BROWSE**.

El subcomando **GET** (coger) del menú de **Modules**, extraemos el componente del diseño cuya **referencia** hayamos introducido. Si no introducimos ninguna referencia nos aparecerán todas en una ventana móvil en donde seleccionamos la del componente que nos interese extraer.

Con el subcomando **ROTATE** podemos girar los componentes 90 grados cada vez y teniendo el cursor como eje de giro.

El subcomando **SIDE** nos permite cambiar de cara los componentes, con el fin de situar componentes en ambos lados del circuito impreso.

Con **FIND** buscamos componentes, por ejemplo: para buscar el condensador C2 activaremos FIND y teclearemos C2. Tras validar con <ENTER>, el cursor se desplaza hasta colocarse encima de C2 pero sin quedar enganchado.

Con **JUMP** salta el cursor a las coordenadas seleccionadas por el comando **TAG** del **Menú Principal**.

Con **UNITS** seleccionamos el tipo de unidades **milímetros o pulgadas** para trabajar con más comodidad. Normalmente se trabaja en **milímetros**.

## Otros comandos de Place

En el menú de la figura 7.23, también tenemos otros subcomandos como **TEXT**, **ZONE** y **EDGE**.

El menú de **TEXT** nos permite poner (**Place**) textos en nuestro diseño. Estos textos pueden ir colocados tanto vertical (**Vert**) como horizontalmente (**Horiz**); también pueden ser reflejados (**mirror**), pudiendo además, elegir el tipo (**Type**) del texto, (ver figura 7.28), en serigrafía (**Silk**) o en cobre (**Copper**).

En el menú de **ZONE** podemos delimitar zonas dentro del diseño mediante los comandos **Begun** (**empezar**), **New** (**nueva**), y **End** (**fin**), con End terminamos de delimitar la zona pero tenemos que especificar cual es el tipo (**Type**) de zona de los tres que hay en el menú: **Copper** (**cobre**), **Forbidden** (**sin pistas**) y **no vías** (**sin vías o pasos de cara**).

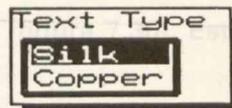


Figura 7.28

En el menú de **EDGE** se delimitan las dimensiones del diseño mediante una o varias líneas, como en nuestro ejemplo, que marcan el **borde** del diseño (ver **fig.29**) esto es importante porque no se podrán trazar pistas fuera de estos límites.

En esta figura también vemos que los componentes han sido colocados en su posición definitiva, incluido el disipador y los tornillos; quedando, el diseño, preparado para la operación posterior que es el **trazado de pistas** en el menú de **ROUTING**.

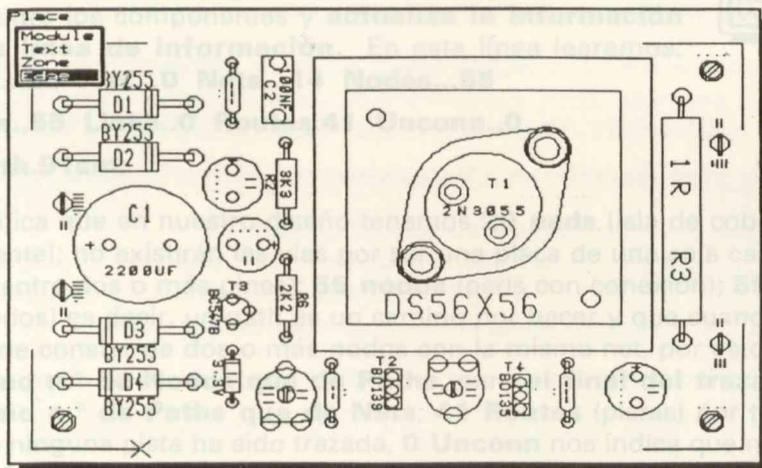


Figura 7.29



## El menú de ROUTING

La operación que sigue a la colocación de componentes, textos y zonas en el diseño, es el **trazado de pistas**; esta operación está condicionada directamente a una buena colocación de componentes. De lo anterior se deduce que una mala colocación de componentes puede hacer imposible el trazado de las pistas.

El menú donde se realiza esta operación es el de **ROUTING** (figura 7.30). Este menú consta, a su vez, de varios comandos y submenús.

Una vez hayamos accedido al menú de **Routing** podemos optar por dos tipos de trazado de pistas: Modo **manual o automático (AUTO ROUTING)**.

### Modo manual

En este modo, el trazado de una pista se realiza siguiendo los siguientes pasos:

- 1.º Entraremos en el menú de **NETLIST** y ejecutamos **COMPILE**, ver **fig.31**, este subcomando realiza la lectura de la NETLIST del diseño y la lectura de las conexiones realizadas entre los componentes y **actualiza la información de la línea de información**. En esta línea leeremos:  
**Pads...20 Vias...0 Nets...14 Nodes...55**  
**Paths..55 Links..0 Routes.41 Unconn..0**  
**Length.91cm.**

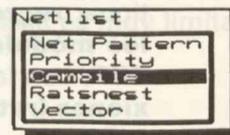


Figura 7.30

Esto significa que en nuestro diseño tenemos **20 pads** (isla de cobre del terminal del componente); no existirán las vías por ser una placa de una sola cara; que hay **14 nets** (unión entre dos o más pines); **55 nodos** (pads con conexión); **55 paths** (camino entre nodos) es decir, un path es un camino por hacer y que cuando se convierte en pista puede constar de dos o más nodos con la misma net, por esto, **al principio hay el mismo n.º de Nodos que de Paths**, pero **al final del trazado de pistas hay el mismo n.º de Paths que de Nets**; **41 Routes** (pistas) por trazar; **0 Links** significa que ninguna pista ha sido trazada, **0 Unconn** nos indica que no hay ninguna pista que no haya podido ser trazada; y finalmente **Length 91cm.** indica que la longitud actual de pistas por trazar es de 91 cm.

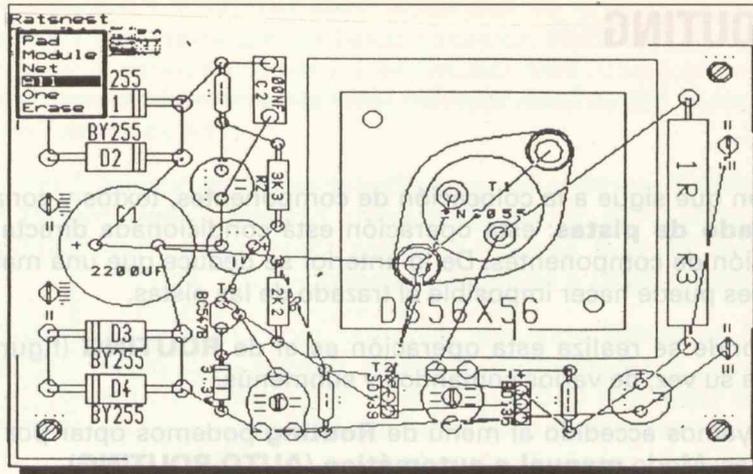


Figura 7.31

2.º Tras ejecutar **compile**, Y para tener una idea de como van unidos los componentes en el diseño, ejecutamos el subcomando **RATSNEST** (nido de ratas), ver **fig.31**, en los puntos del diseño que en se crucen un excesivo número de hilos habrá una densidad de pistas muy alta que será conveniente descongestionar **regresando al menú de PLACE para modificar la colocación de los módulos**. En estos casos, puede ser interesante utilizar el menú de **VECTOR**, con la opción **ALL** podemos ver el **centro de gravedad de las conexiones (ratsnest)** en cuyo punto estaría la colocación óptima del componente.

3.º Regresamos al menú de **ROUTING** y, antes de iniciar el trazado de pistas, nos conviene conocer otros subcomandos de este menú como son: **INQUIRE**, **SHOW**, **HIGHLIGHT**, **WIDTH**, **LAYER PAIR**, **CLEAN UP** y **LAYER**.

Con **INQUIRE** obtenemos información de las pistas y pad.

Con **SHOW** nos muestra la conexión de un pad.

Con **HIGHLIGHT** iluminamos una ruta.

Con **WIDTH** dimensionamos el ancho de las pistas y/o vías.

Con **LAYER** cambiamos de cara (de la 1 a la 2 o viceversa).

Con **LAYER PAIR** determinamos cual es la primera y la última.

Con **CLEAN UP** al ejecutar este subcomando que borran las pistas mal conectadas o interrumpidas.

4.º Para el trazado manual de pistas, ver **fig 7.32**, usaremos el subcomando **BEGIN**. Colocándonos sobre un pad pulsamos **Begin** (con el botón izquierdo del ratón “picar” y si lo indica abajo **pulsar el botón derecho “escape”**) y continuamos trazando la pista (con **begin**) hasta el siguiente **pad**. Para dejar una pista y trazar otra usamos **New (nueva)**. Con **Other** colocamos una vía y cambiamos de cara.

Con **Width** podemos variar la anchura de la pista durante su trazado.

Cambiamos de cara de trabajo con **Layer**.

Es importante acostumbrarse a seleccionar el **Zoom** más apropiado para trabajar con más comodidad ya que este tipo de trabajo puede llevar varias horas.

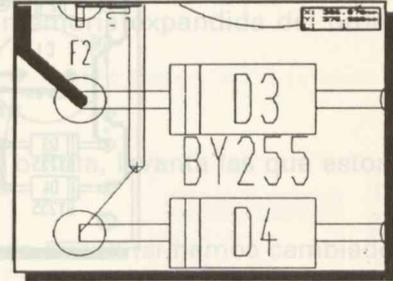


Figura 7.32

## Autorouting

Suele dar buenos resultados cuando se trata de diseños con alta densidad de circuitos integrados, pero en diseños con componentes discretos su eficacia es muy limitada, como en nuestro ejemplo.

Como norma general conviene hacer:

- 1.º **Trazado automático.**
- 2.º **Optimizado automático.**
- 3.º **Corregir manualmente las pistas mal trazadas J'. y las no trazadas.**

En la Figura 7.33 vemos el menú de **AUTO ROUTE** y vemos que **ALL** se encuentra seleccionado para ejecutarse.

Este subcomando se utiliza para realizar el trazado automático de **todo el diseño**, como muestra la Figura 7.34. Con **Pad** sólo traza una pista; con **Module** traza todas las pistas que van a un **módulo**; con **Net** traza sólo una **net**; con **Block** traza las pistas que se encuentren dentro de un bloque; y con **One** (una) realiza cada vez una pista.

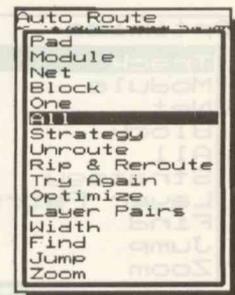


Figura 7.33

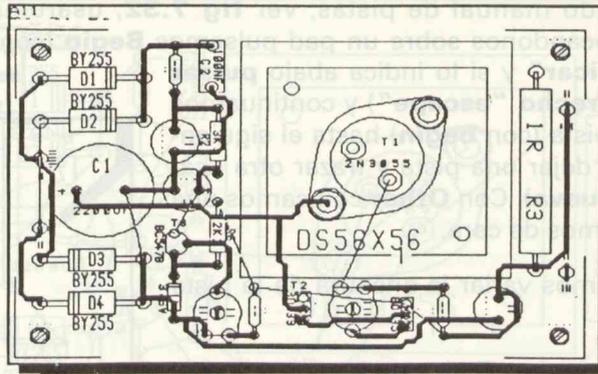


Figura 7.34

## Strategy

En el menú de **STRATEGY** seleccionamos las estrategias que más nos convengan en cada caso particular (ver figura 7.35).

En las estrategias se encuentran los algoritmos de trazado de pistas que el programa utiliza en **modo automático**.

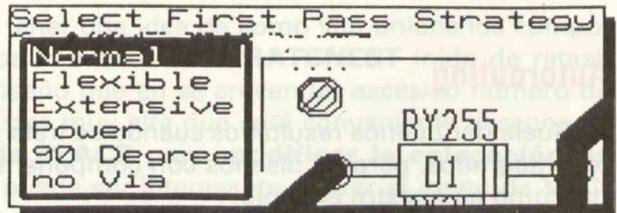


Figura 7.35

En el trazado automático **el programa realiza dos pases y en cada pase utiliza una estrategia distinta**, por esto hay que seleccionar siempre dos de las opciones del menú.



Figura 7.36

El menú de **OPTIMIZE** (figura 7.36), es muy importante porque permite, en un trazado automático, mejorar el recorrido y aspecto de las pistas y reducir el n.º de vías.

Se puede optimizar una pista, **TRACK**; las pistas de un módulo, **MODULE**; una, **NET**; un bloque, **BLOCK**; y todo el diseño, **ALL**.

Con **STRATEGY** (figura 7.37), seleccionamos la estrategia de optimización: **GENERAL** o **MAXIMAL**.

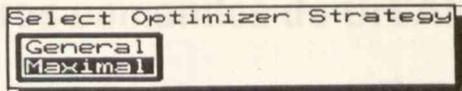


Figura 7.37

El proceso de **optimización**, si el diseño es complejo, puede resultar muy lento, por lo que conviene tener un ordenador rápido. Este programa no utiliza el coprocesador matemático y tampoco usa la memoria expandida del ordenador.

El subcomando **UNROUTE** nos permite **borrar Rutas**.

Con **RIP&REROUTE** le obligamos a hacer una Pista o Ruta, levanta las que estorban e intenta luego trazarlas.

Con **TRY AGAIN** vuelve a intentar trazar una pista de nuevo (si hemos cambiado de **estrategia**).

Al salir de este menú, nuestro diseño estará prácticamente terminado y sólo necesitará mejorar su presentación haciendo una colocación óptima de los textos para facilitar su lectura en la **cara de serigrafía**.

Conditions	
Free Module Buffer	66836 bytes
Free Text Buffer	4086 bytes
Free Track Buffer	50112 bytes
Free System Memory	74086 bytes
Grid	1.270mm / 0.050"

Figura 7.38

**CONDITIONS**. Al igual que en otros módulos de **OrCAD**, este comando informa de la memoria libre de los **buffers** del programa; de la memoria del sistema y del paso de la malla (ver figura 7.38).

Como su nombre indica sirve para borrar.

Al acceder al menú de **Delete** aparecen dos opciones: borrar Objetos o borrar Bloques.

— **OBJETO**. En este submenú podemos borrar segmentos de pista, con **SEGMENT**.

También podemos borrar pistas, con **TRACK**; una **NET** y una zona, con **ZONE**.

— **BLOCK**. Similar al anterior, pero actúa dentro de un bloque (ver figura 7.40). Podemos borrar módulos borrarán también; el quere-

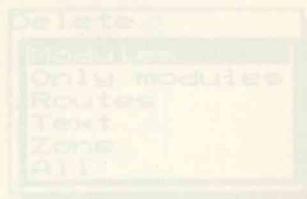


Figura 7.40



## Otros comandos de OrCAD PCB II

En este capítulo se engloban los menús que son similares en el resto de los programas ORCAD como:

**AGAIN.** Repite el último comando ejecutado.

**BLOCK.** Para realizar operaciones con bloques.

Dentro del menú de **Block** (Figura 7.38) podemos **mover o copiar un bloque**, utilizando los comandos **Move** o **Copy** respectivamente.

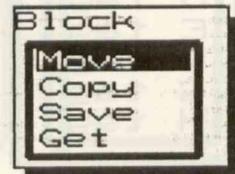


Figura 7.38

Puede resultar muy interesante realizar un bloque con el diseño entero con el fin de duplicarlo, por ejemplo; podemos **salvarlo con SAVE y recuperarlo con GET**. Tanto **Copy** como **Get** pueden actuar sobre parte del diseño (**pistas, módulos, etc**) o en su totalidad.

Conditions		
Free Edge Buffer	2048	bytes
Free Module Buffer	65536	bytes
Free Text Buffer	4095	bytes
Free Track Buffer	90112	bytes
Free System Memory	74096	bytes
Grid	1.270mm /	0.050 "

Figura 7.39

**CONDITIONS.** Al igual que en otros módulos de **ORCAD**, este comando informa de la memoria libre de los **buffers** del programa; de la memoria del sistema y del paso de la malla (ver figura 7.39).

**DELETE.** Es un menú importante dada la frecuencia con que es utilizado.

Como su nombre indica sirve para **borrar**.

Al acceder al menú de **Delete**, aparecen dos opciones: borrar **Objetos** o borrar **Bloques**.

— **OBJETO.** En este submenú podemos borrar **segmentos de pista, con SEGMENT**.

También podemos borrar **pistas, con TRACK**; una **NET** y una **zona, con ZONE**.

— **BLOCK.** Similar al anterior, pero actúa dentro de un bloque (ver figura 7.40). Podemos borrar módulos borrarán también); si **quiere-**

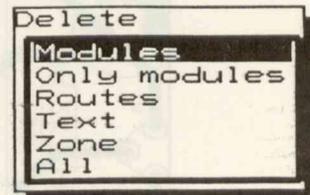


Figura 7.40

mos borrar solo módulos, con **ONLY MODULES**; para borrar las rutas, con **ROUTES**; los textos, con **TEXT**; las zonas, con **ZONES**; y para borrar todo, con **ALL**.

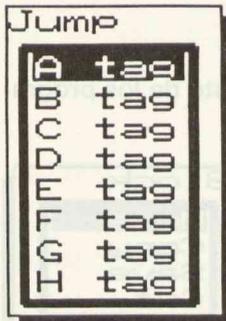


Figura 7.42

**Comando TAG.** Con el comando Tag (figura 7.41), podemos memorizar hasta 8 posiciones en la pantalla (desde **A** hasta **H**).

**Comando JUMP.** Este comando (figura 7.42), nos permite saltar a las posiciones memorizadas con **Tag**.

**ZOOM.** Utilizaremos este menú para ampliar o disminuir la hoja de trabajo.

En el menú de **Zoom** (figura 7.43), hay 5 opciones: **CENTER**, para centrar el diseño; **IN**, para aumentar un paso en la escala.

**OUT**, para decrementar un paso en la escala; **AUTO**, modo automático; **SCALE**, seleccionamos manualmente la escala.

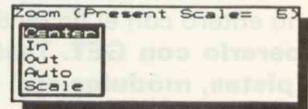


Figura 7.43

**EDIT.** El menú de **Edit** no resulta fácil de explicar, ya que dependiendo de lo que editemos así modificaremos unas cosas u otras. Si editamos una pista podremos modificar su grosor; si es un pad, podremos modificar sus dimensiones, su forma o su net; si es un texto podremos modificar sus dimensiones, su lugar o rotarle, etc.

Cuando movemos textos con **Edit** conviene trabajar fuera de la malla (**x on grid** ....no).

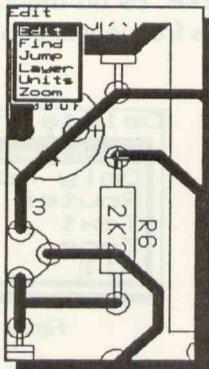


Figura 7.44

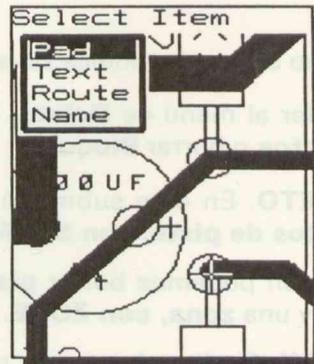


Figura 7.45

La figuras 7.44 y 7.45 muestran el proceso de edición de un pad; para ello se coloca el cursor en el pad de R6 y entramos en el menú de **Edit**, donde seleccionaremos la opción **Netlist** con lo que ya podremos modificar su nombre (N00009), por ejemplo (figura 7.46).

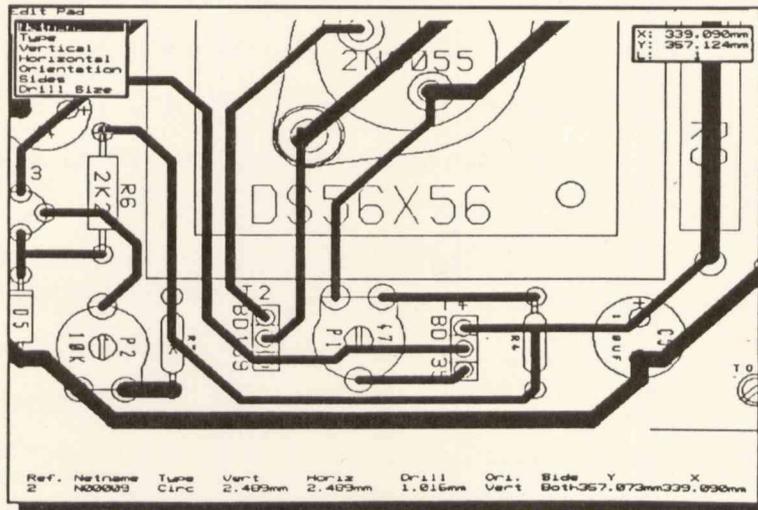


Figura 7.46



# Índice

	<u>Página</u>
INTRODUCCIÓN .....	237
EVOLUCIÓN DE LA ESTRUCTURA DE UN AUTÓMATA PROGRAMABLE .....	239
CARACTERÍSTICAS DE LOS COMPONENTES DE UN AUTÓMATA PROGRAMABLE .....	245
Unidad central de proceso .....	245
Procesador .....	245
Memoria .....	246
Entradas y salidas .....	247
PROGRAMACIÓN .....	249
Listas de instrucciones .....	249
Esquemas de contactos .....	254
Diagrama de funciones .....	254
Gráficos .....	254
EQUIPOS DE PROGRAMACIÓN Y PERIFÉRICOS .....	251
COMUNICACIONES .....	251
VOCABULARIO .....	263
BIBLIOGRAFÍA .....	265
PRÁCTICAS .....	267

## Autómatas programables



# Índice

	<i>Página</i>
INTRODUCCIÓN .....	237
EVOLUCIÓN DE LA ESTRUCTURA DE UN AUTÓMATA PROGRAMABLE .....	239
CARACTERÍSTICAS DE LOS COMPONENTES DE UN AUTÓMATA PROGRAMABLE .....	245
Unidad central de proceso: .....	245
Procesador.....	245
Memoria .....	246
Entradas y salidas .....	247
PROGRAMACIÓN.....	249
Listas de instrucciones .....	250
Esquemas de contactos.....	254
Diagrama de funciones .....	254
Grafcet .....	255
EQUIPOS DE PROGRAMACIÓN Y PERIFÉRICOS .....	259
COMUNICACIONES .....	261
VOCABULARIO.....	263
BIBLIOGRAFÍA.....	265
PRÁCTICAS.....	267



## Introducción

Cuando en un proceso industrial se introduce un dispositivo tecnológico que controla su funcionamiento decimos que está automatizado generándose un sistema denominado **automatismo** que podemos representar así, (figura n.º1):

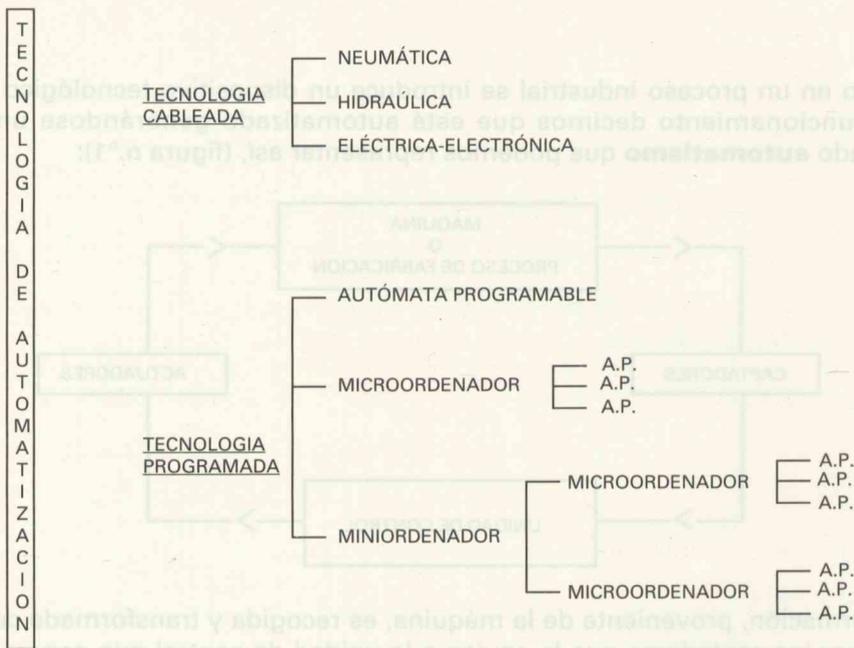


La información, proveniente de la máquina, es recogida y transformada convenientemente por los captadores que la envían a la unidad de control que genera las órdenes correspondientes, éstas se transmiten a la máquina a través de los actuadores o accionadores que las transforman en magnitudes con el nivel de potencia adecuado para interactuar sobre ella.

La constante evolución de la tecnología ha permitido que el nivel de automatización sea cada vez más elevado, hasta llegar al control completo de un proceso de producción.

En la actualidad, debido a la revolución informática, los sistemas productivos interactúan con las demás áreas de una empresa haciendo posible el concepto de Automatización Integrada —**CIM**—, en la que los equipos de control como autómatas programables, equipos de soldadura, robots, controles numéricos etc..., que forman parte de las **islas de automatización**, se integran en un único sistema de control intercambiando información entre ellos y con los sistemas informáticos de las otras áreas a través de **redes de comunicación en área local**, consiguiendo la optimización de la producción en todos los aspectos. Numerosas empresas están efectuando la normalización en el campo de las comunicaciones a través de un Protocolo para la Automatización de la producción denominado **MAP**.

En los procesos de automatización se han utilizado diferentes tecnologías, como podemos observar en el siguiente esquema:



Debido a los inconvenientes que presenta la tecnología cableada, **a mediados de la década de los sesenta**, la General Motors solicitó la colaboración de la Digital Corporation para lograr un sistema de control adaptado al medio industrial, programable en un lenguaje asequible a los operarios y de fácil mantenimiento.

Paralelamente **R.E. Moreley, considerado el padre de los Automatas Programables**, trabajando para la firma Bedford Associates desarrolló un producto que cumplía las especificaciones anteriores. Trabajaba con memoria de ferritas y un procesador hecho con varios circuitos integrados; este dispositivo sustituía a los armarios de relés y se adaptaba al control de procesos secuenciales.

**En la primera década de los setenta** se incorpora la tecnología del microprocesador permitiendo la comunicación del autómata con el computador y aumentando sus prestaciones que se han ido ampliando hasta nuestros días, lo que junto a una considerable reducción de los precios hace que sean utilizados ampliamente en todos los sectores industriales.

# Evolución de la estructura de un autómata programable

## ¿Qué es un Autómata programable?

Es un sistema de control informatizado, desarrollado para controlar procesos industriales.

Para comprenderlo vamos a estudiar cómo ha ido evolucionando su desarrollo.

Cuando en una instalación industrial es necesario automatizar un proceso, en líneas generales, podemos decir que se actúa sobre una o más salidas digitales en función del estado de unas variables binarias de entrada. Esta función la realiza un sistema llamado **controlador lógico**.

Según su forma de operación, estos sistemas pueden ser:

Combinacionales

Secuenciales: Síncronos

Asíncronos

Existen circuitos lógicos realizados en un solo circuito integrado que pueden ser programados para realizar un sistema combinatorial o secuencial concreto. A estos circuitos se les denomina **Dispositivos Lógicos Programables** (PLD o DLP)

Se diseñan mediante computador, utilizando:

i) un software de diseño

ii) un hardware apropiado, que puede ser:

— tarjeta (interface) para el ordenador conectado a un módulo externo.

— periférico independiente conectado vía RS-232C.

## CLASIFICACION DE LOS PLD

Según su función lógica:

Combinacionales

Matrices lógicas programables (PLA)

- Matrices lógicas Y-programables (PAL)
- Matrices lógicas programables (FPAL)
- Matrices de puertas Y o decodificadores programables (PGA)
- Comparadores programables
- Memorias
  - RAM estáticas (SRAM)
  - RAM dinámicas (DRAM)
  - RAM no volátiles (NVRAM)
  - ROM
  - PROM
  - EPROM
    - EAPROM
    - EEPROM
    - UVPROM

Secuenciales      Secuenciadores lógicos programables (PLS)

Mixtos              Matrices de puertas realimentadas (PLM)-programable macro logic-

Según la tecnología

Bipolares

CMOS

En esencia, un **autómata programable** puede considerarse como un dispositivo **controlador lógico programable (PLC)** que ejecuta las ecuaciones lógicas mediante una secuencia de operaciones realizadas por un circuito combinacional programable, denominado **unidad lógica**.

Es decir

$$\text{PLC+UL=AP}$$

El **autómata programable** estaría configurado de la manera siguiente:

La **unidad de control** constituida por:

- un generador de impulsos
- un contador síncrono
- una memoria pasiva (ROM, PROM, EPROM, EEPROM)
- circuito combinacional que genere las señales de control

La **unidad operativa** constituida por:

- unidad de entrada UNE
- unidad lógica UL
- memoria de almacenamiento de resultados parciales
- unidad de salida UNS

Según la diferente concepción del hardware de la unidad operativa, que conlleva un conjunto de instrucciones también diferentes, pueden realizarse múltiples **autómatas programables**.

Al controlar un proceso industrial, muchas veces hay que procesar variables analógicas que, convertidas en digitales, den lugar a variables binarias que representen información numérica.

Por ello, es necesario un sistema electrónico que pueda procesar información numérica. Además, este sistema debe responder rápidamente a los cambios de las variables de entrada.

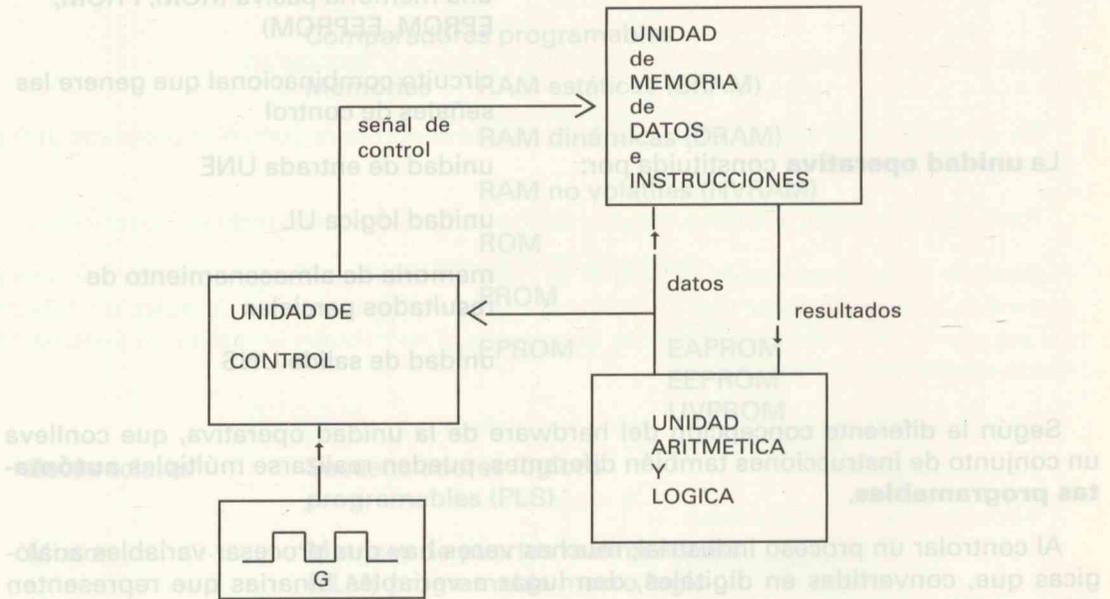
Por estas razones, cuando debido al desarrollo de la microelectrónica nace el **microprocesador** en el que se introduce, en un solo circuito integrado, la **unidad lógica y la unidad de control** de un computador, constituyendo su **CPU**, los fabricantes de autómatas programables empezaron a comercializarlos con microprocesador.

Estos autómatas con capacidad de cálculo numérico pueden tratar variables analógicas, informaciones numéricas y alfanuméricas y están dotados de lenguajes idóneos para resolver problemas de control.

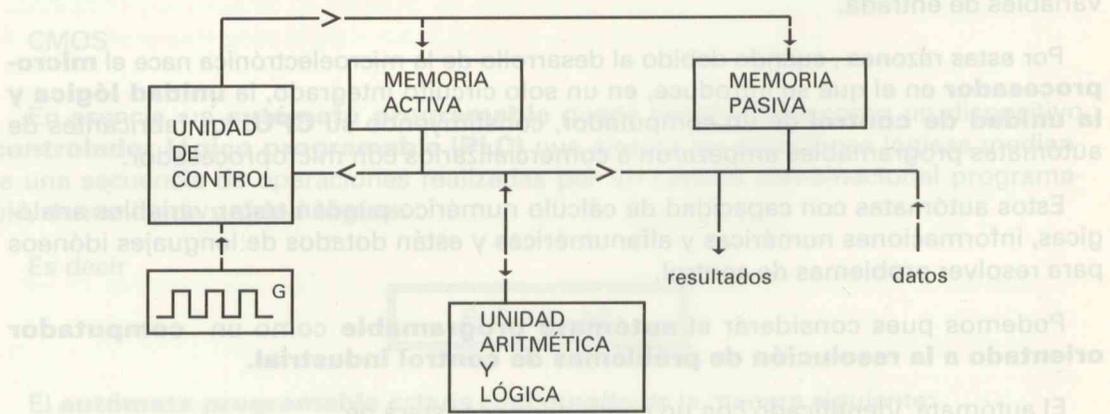
Podemos pues considerar el **autómata programable** como un **computador orientado a la resolución de problemas de control industrial**.

El autómata, identificado con un computador constará de:

Unidad Aritmético Lógica ALU  
 Unidad de Control  
 Memoria de instrucciones y datos

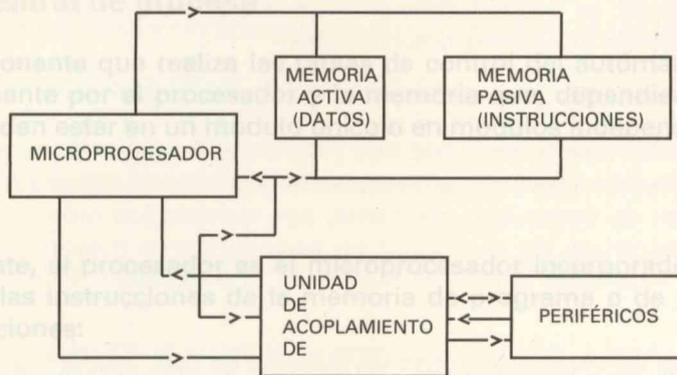


La memoria de datos e instrucciones suele estar físicamente separada en dos bloques, por lo que el diagrama quedaría representado según la figura siguiente:

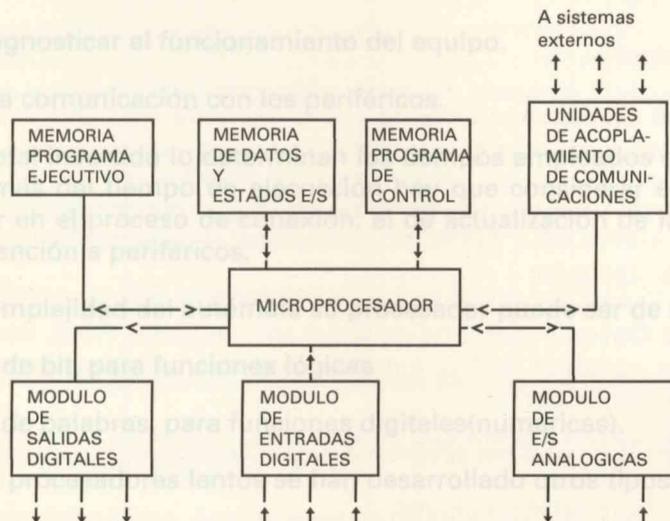


La unidad de control dirige la realización del proceso, y la unidad aritmética-lógica lo ejecuta. Estos dos bloques suelen estar en uno solo constituyendo la CPU. Cuando este bloque está en un solo circuito integrado recibe el nombre de microprocesador, como hemos mencionado más arriba.

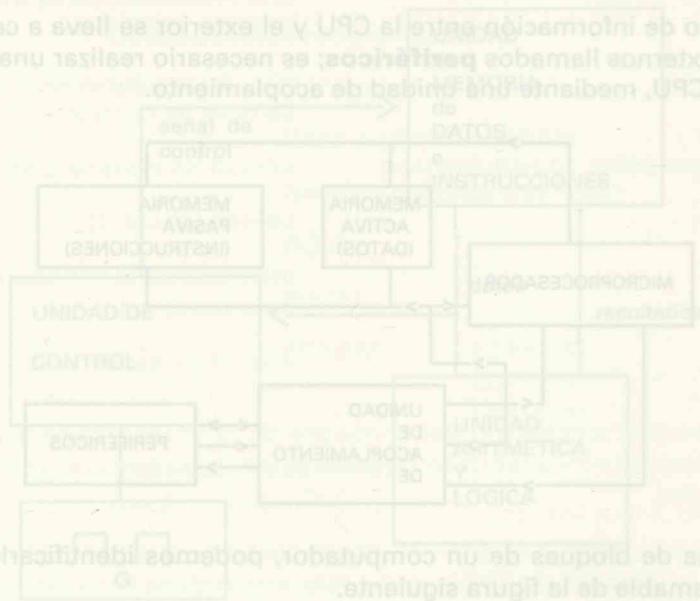
El intercambio de información entre la CPU y el exterior se lleva a cabo a través de unos sistemas externos llamados **periféricos**; es necesario realizar una sincronización entre éstos y la CPU, mediante una unidad de acoplamiento.



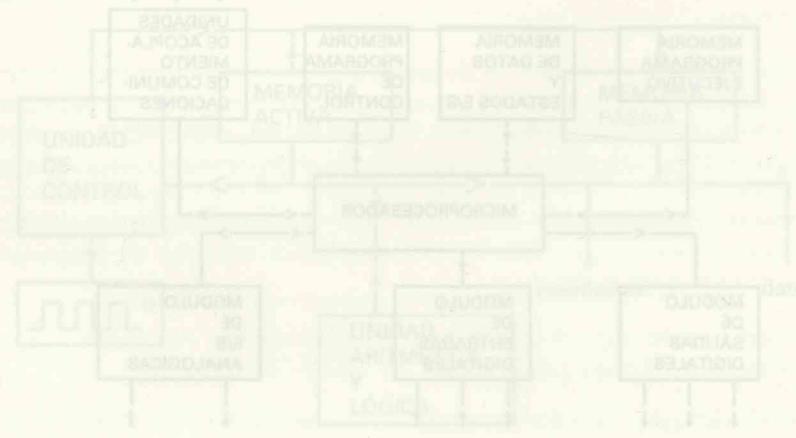
Este diagrama de bloques de un computador, podemos identificarlo con el de un autómatas programable de la figura siguiente.



Las distintas combinaciones que pueden hacerse con estos elementos y la variedad de los mismos hacen que el número de autómatas de diferentes características que existen en el mercado sea tan considerable.



La memoria de datos e instrucciones suele estar físicamente separada en dos bloques, por lo que el diagrama quedaría representado según la figura siguiente:



# Características de los componentes de un autómata programable:

## La unidad central de proceso

Es el componente que realiza las tareas de control del autómata. Está constituido fundamentalmente por el procesador y la memoria que, dependiendo del modelo de autómata, pueden estar en un módulo único o en módulos independientes.

### El procesador

Normalmente, el procesador es el microprocesador incorporado al autómata que, interpretando las instrucciones de la memoria de programa o de usuario, realiza las siguientes funciones:

Adquirir y actualizar los estados de las señales de entradas y salidas, en función del programa de aplicación, resolviendo el conjunto de operaciones de forma secuencial, instrucción tras instrucción, invirtiendo en ello un tiempo denominado de ejecución o de ciclo (scan time) que es una característica importante del autómata.

Vigilar y diagnosticar el funcionamiento del equipo.

Establecer la comunicación con los periféricos.

El tiempo total invertido lo determinan los tiempos empleados en las distintas operaciones. Además del tiempo de ejecución hay que considerar el tiempo de espera del procesador en el proceso de conexión, el de actualización de las E/S al finalizar el ciclo y el de atención a periféricos.

Según la complejidad del autómata su procesador puede ser de distintos tipos:

Procesador de bit, para funciones lógicas

Procesador de palabras, para funciones digitales(numéricas).

Al ser estos procesadores lentos se han desarrollado otros tipos para paliar este inconveniente:

Procesadores de palabras microprogramables para operaciones de palabra con gran velocidad

Procesadores de bits y de palabra con tecnología logic-array o gate-arrays que facilitan la decodificación rápida de las instrucciones programadas en uno o pocos pasos. Con esta tecnología se desarrollan unas estructuras que son bloques de procesadores individuales, cada uno de los cuales elabora un conjunto de órdenes integrados en un gate-array con una unidad de mando.

Multiprocesamiento, es decir, subdivisión de la elaboración de señales por varios procesadores, con las siguientes posibilidades:

Varios procesadores con un coordinador que controle el conjunto.

Sólo un procesador central con módulos periféricos inteligentes que son pequeños microprocesadores con su propio procesador y memoria encargados de resolver tareas específicas de forma independiente, por ejemplo los módulos de posicionamiento que se encargan de posicionar un accionamiento motorizado en una posición preescrita, alcanzada la cual es notificado al procesador central que envía las órdenes oportunas.

Acoplamiento de varios aparatos con un sistema de bus, que actúan según el principio maestro-esclavo .

## Memoria

Hemos de considerar:

**La memoria del Sistema** constituida por la memoria ejecutiva con unidades tipo ROM o EPROM, donde almacena el fabricante los programas ejecutivos (firmware).

**La memoria de Usuario** o memoria de programa de aplicación, con unidades RAM con baterías, UVPROM o EEPROM , donde se almacenan las instrucciones del programa de control aplicado.

**La memoria de Datos** o memoria de señal con unidades RAM, donde se almacenan el estado de las variables de E/S, los estados de las variables internas y las variables numéricas constituida por:

memoria de datos temporal del programa ejecutivo, denominada scratch-pad.

memoria de E/S de variables digitales.

memoria de datos numéricos correspondientes a variables analógicas, procedentes de convertidores analógico-digitales o digitales-analógicos.

memoria de variables internas.

## Entradas y salidas

Las informaciones recogidas del proceso que se ha de controlar se denominan, genéricamente, **entradas** y las acciones de control sobre la máquina **salidas**.

Los dispositivos que generan las señales de entrada corresponden a diferentes tipos de elementos como interruptores, pulsadores, finales de carrera, detectores de posición, sensores, etc, y los dispositivos que dan la potencia conveniente a las señales de salida, generadas por el sistema de control, son relés, contactores, electroválvulas, etc...

Tanto para las entradas y salidas digitales, como para las analógicas, existen unos módulos de acoplamiento.

En equipos compactos el sistema de entradas y salidas, o parte del mismo, forma parte, junto con la unidad central, de un solo rack. En equipos modulares este sistema está constituido por un conjunto de módulos con una estructura de soporte mecánico (bastidores de montaje o racks donde están los conectores del bus de datos), que cumplen las siguientes funciones:

- Adaptar las tensiones de trabajo

- Proporcionar el aislamiento necesario entre los circuitos lógicos y los circuitos de potencia

- Establecer a través del software la identificación de los dispositivos de entradas y salidas con sus direcciones del mapa de memoria entre los que existe una correspondencia especificada por el fabricante

Existen **Módulos de E/S especiales** para resolver funciones de aplicaciones específicas:

- Módulo de entradas de termopar que aceptan señales del orden de mV.

- Módulo controlador de motor paso a paso que genera trenes de impulsos correspondientes a estos motores (0.02 - 1 msg.)

- Módulo servo-controlador para control de posicionamiento, de uno o más ejes.

Módulos de control PID (control proporcional, integral y derivativo) para desarrollar funciones de regulación, cuyo sistema de trabajo corresponde a una realimentación, en lazo cerrado, para mantener o desplazar una variable a un punto de consigna.

Módulos ASCII que permiten el intercambio de información con periféricos, como por ejemplo, pantallas de datos.

Módulos de interconexión serie para E/S remotas, enlazadas mediante un cable de comunicación y, últimamente, mediante fibra óptica.

Módulos especiales llamados inteligentes o de proceso con microprocesador incorporado que les permite tomar decisiones lógicas, tratar datos o procesar instrucciones de funciones complejas, de comunicación...

# Programación

1.º La **automatización de un proceso** se fundamenta en **intercambios de información** entre:

el proceso a automatizar y el sistema de control

el usuario y el sistema de control

2.º Estos intercambios de información necesitan una **codificación** que constituye el **lenguaje de programación**, a través del cual se especifica la secuencia de tratamiento de la información, es decir, lo que debe hacerse con la información que se requiere del sistema de control, según las necesidades del problema de automatización y los recursos del Dispositivo controlador utilizado.

3.º El conjunto de tareas que debe efectuar la Unidad Central de Proceso para resolver una determinada tarea de control, especificadas mediante una secuencia que establece la relación entre las distintas variables lógicas, constituye el **programa**.

Una vez definido, éste se transcribe en forma de **instrucciones** propias del lenguaje de programación empleado y se introduce en la memoria del Autómata, en forma de **códigos binarios** mediante un **equipo de programación** el cual traduce las instrucciones al lenguaje máquina propio de cada Autómata Programable, y que depende de su estructura física (hardware).

Los fabricantes usan para sus equipos distintos tipos de lenguajes:

Lista de instrucciones

Diagrama de funciones

Esquema de contactos

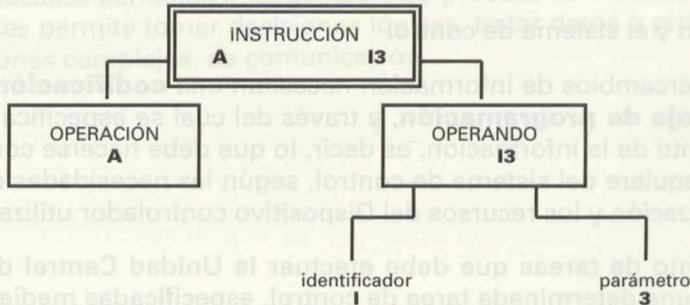
Grafcet

Cada fabricante usa una nomenclatura particular, tanto para las instrucciones como para la denominación de las variables de entrada, salida e internas; y aunque ésta es sencilla y muy parecida en los diferentes tipos de autómatas, esta diversidad dificulta el manejo simultáneo de varios de ellos.

## Lenguaje en lista de instrucciones

Consiste en un conjunto de códigos simbólicos cada uno de los cuales corresponde a una instrucción en lenguaje máquina.

Cada instrucción consta de una parte operacional que describe la función a realizar y un operando que contiene los datos e informaciones necesarios para efectuar dicha función, constituido por el identificador y el parámetro que especifican a qué variables concretas se aplican dichas funciones.



Según la capacidad del autómata podrá interpretar mayor o menor número de instrucciones

### Tipos de instrucciones en un autómata:

De funciones lógicas.

Aritméticas

De tratamiento y manipulación de datos

De temporización y contaje

De control de ciclo de ejecución.

De comunicación.

Otras.

### Funciones lógicas

Permiten resolver el tratamiento de la información que corresponde a procesos lógicos combinacionales o secuenciales mediante la implementación de las funciones lógicas correspondientes:

Función AND (Y) o puerta serie.

Función OR o puerta paralelo.

Función NAND (No Y) o puerta serie negada.

Función NOR (No OR) o puerta paralelo negada.

Función paréntesis:

( abrir paréntesis

) cerrar paréntesis

)N cerrar paréntesis negado

### Instrucciones aritméticas

Corresponden estas instrucciones a las operaciones básicas de:

Suma

Resta

Multiplicación

División.

En ellas se especifica una variable numérica, destino de la operación efectuada entre otras dos variables numéricas (registros)

### Instrucciones de tratamiento de datos

Permiten manejar la información que corresponde a variables numéricas o contenidos de los registros de datos que, en función de la secuencia que se esté ejecutando, tendrán diferentes valores:

#### Operación de Carga

Cargar un valor constante en una palabra

Cargar secuencias de 8 bits

Cargar secuencias de 16 bits

Cargar entradas analógicas

### Operación de Transferencia

- Transferir una secuencia de 8 bits
- Transferir una secuencia de 16 bits

### Operación de Comparación

- Mayor que GT - great than -
- Menor que LT - less than -
- Igual que EQ - equal -

### Instrucciones de temporización y contaje

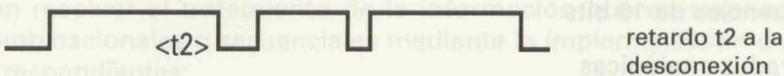
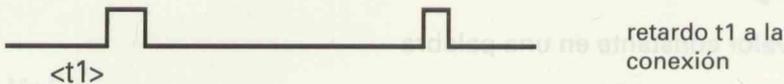
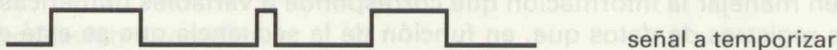
Para que el autómata posea estas instrucciones es necesario que incluya en su hardware temporizadores y contadores:

#### De Temporización

Permiten que una variable temporizada adquiera el estado de una variable de control, según un tiempo preestablecido, llamado valor de consigna del temporizador.

La elección del tiempo de consigna puede hacerse, previamente, asignando un valor al temporizador elegido o bien con una instrucción en la que se especifica dicho tiempo.

La temporización puede efectuarse con retardo a la conexión o a la desconexión.



## De Contaje

Permiten que una función consiga que una variable de salida adquiera el estado 1, cuando hayan tenido lugar un número preseleccionado de transiciones de 0 a 1 del estado de la variable de contaje.

Para contabilizar las transiciones se necesita una variable de control —validación y puesta a cero— y que ésta permanezca a uno.

Puede realizarse la función UP/DOWN contador/decontador y, en este caso, se necesitan dos variables, la de control y la de contaje y decontaje.

## Instrucciones de control

Estas instrucciones, mediante tomas de decisión, influyen en la ejecución de las demás. Su existencia, aunque no es imprescindible en un autómata, facilita su programación.

Permiten alterar la ejecución secuencial de las instrucciones del programa mediante saltos condicionales; su abandono temporal para realizar subrutinas; la optimización del tiempo de ejecución del programa ejecutando algunas instrucciones sólo cuando se requieren; etc...

## Instrucciones de comunicación

Permiten el intercambio de información entre dos o más autómatas enlazados por un canal de comunicaciones serie.

Puede ser un enlace entre pocos autómatas en cuya memoria de datos o registros de trabajo tengan zonas libres reservadas, a tal efecto, o bien enlaces mediante red de comunicaciones en los que el acceso a la memoria de estados y de datos es prácticamente libre.

Tipos de instrucciones:

1. De lectura de datos en un procesador
1. De escritura de datos en otro procesador

En estas instrucciones hay que especificar el número de datos a transmitir, la velocidad de transmisión, etc..

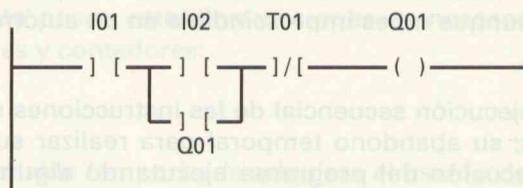
## Lenguaje en esquema de contactos (Ladder Diagram)

Es un lenguaje en el que se representa gráficamente mediante símbolos el proceso de control a realizar.

Los símbolos utilizados para representar contactos, solenoides, enlaces, etc., corresponden a las normas para la representación de esquemas eléctricos DIN 40713-6 y NEMA. Se identifican con la dirección de una variable de entrada o salida y al combinarlos, adecuadamente, definen el estado de una variable de salida.

Las funciones se expresan mediante combinaciones de los símbolos y mediante el uso de expresiones literales de las mismas, incorporadas en el esquema de contactos.

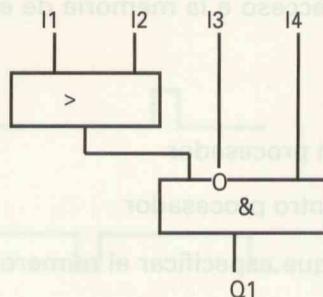
Para la programación de secuencias de cierta complejidad se introducen instrucciones que usan el formato de bloques que permiten agrupar en una sola instrucción varias funciones; en el bloque se especifican las variables y parámetros correspondientes.



Cuando la entrada I2 o la salida Q1 (paralelo) y la entrada I1 (serie) están en estado 1, y el temporizador T1 (serie) está en estado 0, la señal se dirige hacia la salida Q1.

## Lenguaje en diagrama de funciones

Es un lenguaje simbólico en el que se representan las combinaciones entre variables mediante símbolos lógicos normalizados, según normas ISO



Cuando las entradas I1 o (paralelo) la I2 están en estado 1 y (serie) las entradas I3 e I4 están, respectivamente, en estado 0 y 1 se activa la salida Q1, es decir:

$$(I1+I2).I3.I4=Q1$$

## Lenguaje de Grafcet -Graphe de Comande Etape-Transicion-

Fue desarrollado por la AFCET, Asociación Francesa para la Cibernética Económica y Técnica e impulsado y generalizado su uso a través de la ADEPA, Agencia Nacional para el desarrollo de la Producción Automatizada.

Es un diagrama funcional, que permite una representación gráfica de las secuencias que debe efectuar la unidad de control, independientemente de cómo se programen en el autómatas que se utilice para su ejecución.

Considera el proceso a controlar —máquina o sistema— constituido por:

- un órgano operativo que es el proceso físico a automatizar
- un órgano de control, con el que el usuario mantiene un intercambio de información, que gobierna las secuencias del proceso.

Los principales elementos de definición y expresión gráfica del lenguaje Grafcet son:

**ETAPAS**, que son los estados estables en los que se descompone el proceso (representados por un cuadrado numerado en su interior; la etapa inicial se representa con doble línea). A cada etapa se asocian una o varias **ACCIONES**, representadas por un rectángulo.

**TRANSICIONES**, que son las condiciones lógicas de evolución del sistema que permiten pasar de una etapa a la siguiente.

**ENLACES**, que son las uniones entre etapas y transiciones con una dirección significativa.

Las condiciones que deben cumplirse son:

- 1.º Para que se ejecute una acción debe estar activa la etapa a la que esté asociada.
- 2.º Para que una etapa esté activa debe estarlo la etapa precedente y debe verificarse la condición lógica de la transición correspondiente.

3.º Siempre que se ejecuta una transición se activa la etapa siguiente y se desactiva la precedente.

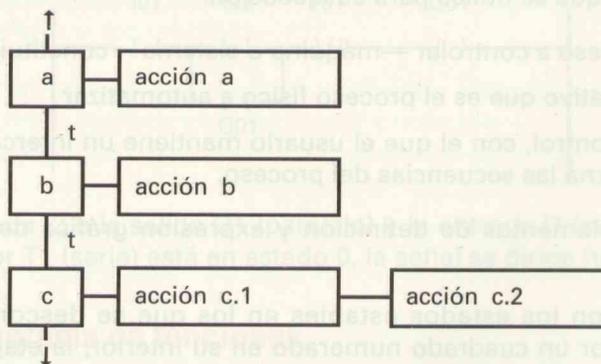
### Tipos de acciones:

**Acciones continuas:** la acción se efectúa directamente si la etapa a la que va asociada está activa.

**Acciones condicionales:** para que la acción se efectúe, además de estar activa la etapa a la que está asociada, debe cumplirse una condición lógica suplementaria; si ésta es una temporización, la acción se llama temporizada.

**Acciones puntuales:** son acciones que sólo se activan la primera vez que en el proceso se activa su etapa asociada; si con posterioridad ésta vuelve a activarse, la acción ya no se repite.

### DIAGRAMA FUNCIONAL



La acción **a** está asociada a la etapa **a**; **t** es la condición de transición para pasar a la etapa **b** que lleva asociada la acción **b**. La etapa **c** tiene asociadas dos acciones.

Cuando se utiliza un diagrama Grafcet, se realiza en dos niveles:

- 1.º nivel en el que se describen las acciones a desarrollar en cada etapa, y las condiciones de paso a la siguiente.
- 2.º nivel en el que se desarrollan las especificaciones del nivel anterior describiendo los dispositivos que realizan las acciones, y los que informan sobre las condiciones de transición.

Este último diagrama puede expresarse mediante ecuaciones lógicas que pueden traducirse en un lenguaje de programación. Algunos fabricantes tienen implementado

el Grafcet en sus equipos de programación; a través de un software específico se puede introducir un diagrama Grafcet cualquiera, simular el proceso visualizando estados y traducir el diagrama, introducido en las instrucciones correspondientes, con el lenguaje propio del autómeta.

En la actualidad se está desarrollando un nuevo método, superior al Grafcet, denominado GEMMA.

**Unidades de programación:** es el computador que ofrece la mayoría de las operaciones de programación con técnicas de instrucciones, de mandatos y numérico, que tienen un pequeño display para mostrar el programa instrucción por instrucción o incluso un pequeño grupo de instrucciones. El uso de estos equipos es imprescindible para los operarios en la industria por su facilidad de uso y mantenimiento.

**Consolas de programación:** con teclado y pantalla CRT (Tubo de Rayos Catódicos) que suelen tener un display de archivo de programas en cinta o disco y teclado a través del cual se ingresan los programas.

**Terminales de programación:** con entrada de datos y visualización de gráficos mediante técnicas de vídeo, teclado por computador CAD (lámpa óptica, tarjetas digitalizadoras, etc.).

**Ordenador que, en la actualidad, está siendo el equipo de programación más utilizado debido a la generalización de su uso y a su imprescindible potencia. Se establece la comunicación con el autómeta, vía serie RS-232C o RS-422, con el protocolo de comunicación que especifica el fabricante.**

#### h) Características del software:

Cada fabricante ha desarrollado para sus autómetas un software específico, que normalmente ofrece las siguientes posibilidades:

1. Trabajo on-line, sin conectar el autómeta, con un menú de opciones que nos permita:
  - 1.1 Escribir un programa nuevo, o bien recuperar uno salvado previamente, utilizando los dispositivos de lenguaje.
  - 1.2 Documentario, introduciendo comentarios al programa.
  - 1.3 Archivarlo, salvándolo en alguna.
  - 1.4 Imprimirlo.



# Equipos de programación y periféricos

Para introducir en la memoria del autómatas las instrucciones que definen las secuencias correspondientes al programa de control se emplean los equipos de programación:

**Unidades de programación** tipo calculadora que ofrecen la mayoría de las operaciones de programación con teclado de instrucciones, de mandatos y numérico, que tienen un pequeño display para visualizar el programa instrucción por instrucción o incluso un pequeño grupo de instrucciones. El uso de estos equipos es imprescindible para los operarios en la industria por su facilidad de uso y desplazamiento.

**Consolas de programación** con teclado y pantalla CRT (Tubo de Rayos Catódicos) que suelen tener un dispositivo de archivo de programas en cinta o disquete y salida a impresora.

**Terminales de programación** con entrada de datos y visualización de gráficos mediante técnicas de diseño asistido por computador CAD (lápiz óptico, tarjetas digitalizadoras...).

**Ordenador** que, en la actualidad, está siendo el equipo de programación más utilizado debido a la generalización de su uso y a su indiscutible potencia. Se establece la comunicación con el autómatas, vía serie RS-232C o RS-422, con el protocolo de comunicación adecuado, según especificación del fabricante.

## i) Características del software:

Cada fabricante ha desarrollado para sus autómatas un software específico, que normalmente ofrece las siguientes posibilidades:

1. Trabajo en off-line, sin conectar el autómatas, con un menú de opciones que nos permite:
  - 1.1 Escribir un programa nuevo, o bien recuperar uno salvado previamente, utilizando los distintos tipos de lenguajes.
  - 1.2 Documentarlo, introduciendo comentarios al programa
  - 1.3 Archivarlo, salvándolo en disquete.
  - 1.4 Imprimirlo

Esta posibilidad de trabajo es muy útil ya que permite realizar el programa, modificarlo, comprobar su funcionamiento por simulación —chequeo—, hacer la documentación pertinente y, con posterioridad, transferir el programa al autómatas para su ejecución.

2. Trabajo en on-line, con el autómatas conectado. También podemos transferir este programa al ordenador para realizar cualquiera de las posibilidades antes comentadas.

Para sistemas de automatización muy desarrollados existe un software modular específico para cada tarea: mando, aritmética, tratamiento de valores de medida, regulación, etc..

## Equipos periféricos

Son los equipos auxiliares del sistema de control:

Módulos de tratamiento de datos para elaborar gráficos, informes, introducir parámetros de control... Se programan en basic, o en ensamblador, y pueden conectarse además de al autómatas, a impresoras, pantallas CRT...

Unidades de test y diálogo que permiten la visualización y modificación de parámetros de control, visualización de mensajes de alarma ...

Impresoras

Instrumentación electrónica

Lectores de códigos de barras

Teclados y displays alfanuméricos

Módulos de entradas y salidas remotas

etc...

Algunos autómatas pueden llevar incorporados un analizador lógico que permite analizar la evolución simultánea de hasta 16 señales.

## Comunicaciones. Redes de autómatas

Normalmente, la comunicación del autómata con las unidades de programación, o los periféricos, se efectúa mediante enlace serie, es decir bit a bit, siendo la información un conjunto de códigos ASCII.

La comunicación puede realizarse en una dirección ( por ejemplo, con la impresora ) o en las dos direcciones, pudiendo, en este caso, ser primero en un sentido, y luego, en otro (semi-dúplex), o simultáneo (dúplex total).

Los protocolos de comunicación están normalizados, siendo las normas EIA las más difundidas.

### Redes de autómatas

Se puede estructurar un sistema de automatización constituido por un conjunto de autómatas conectados entre sí, al que se denomina **red local**.

Trabaja según el principio "maestro-esclavo": un autómata se define como maestro y se le acopla un procesador de comunicaciones que organiza el tráfico de datos por el cable, o conductor de bus, que lleva conectado y que comunica mediante el amplificador de bus a los componentes de la red a los que, cíclicamente, va llamando para el intercambio de información.

Esta red puede estar constituida por unos 30 esclavos y un maestro interconectados por un mismo bus. El maestro intercambia información con el procesador central, a través del procesador de comunicaciones, mediante unos módulos funcionales.

Existen otras redes locales más desarrolladas que admiten unos 100 interlocutores (autómatas, computador) en las que el bus de conexión se realiza mediante cables de fibra óptica con las ventajas que ello reporta en cuanto a inmunidad al ruido electromagnético, evitando con ello los problemas de inducciones, ahorro de cableado, facilidad de conexión...

Cada fabricante desarrolla un hardware para el sistema de bus distribuido que permita la interconexión de los distintos elementos de un sistema de automatización en red local.

Como decíamos en la introducción de este trabajo la tendencia actual de los sistemas de producción es lograr una fabricación integrada mediante computador precisando para ello la integración de las distintas islas de automatización. Ello lleva consigo la necesidad de desarrollar y estandarizar las técnicas de comunicación entre los distintos equipos, tanto en hardware como en software.

Ahora bien, las distintas características de los equipos y las distintas necesidades de tratamiento de la información de las diversas áreas de una empresa hace que, en principio, se demanden diferentes prestaciones a la red de comunicaciones (diferentes tiempos de respuesta, velocidad de transmisión...), lo que dificulta la normalización de ésta.

No obstante, existen dos principios normalizados para el acoplamiento de sistemas individuales por un bus:

El principio Ethernet (CSMA/CD) desarrollado en un principio para el sector comercial pero que hoy día también se aplica en sectores industriales. El mecanismo de acoplamiento se basa en que cada participante puede recibir y emitir, en cualquier momento controlando el propio sistema, las eventuales colisiones.

El principio Token-Passing, en el que el principio de emisión se transpasa de un sistema a otro, según el principio del maestro flotante.

En la actualidad se está desarrollando el protocolo MAP para conseguir la normalización a todos los niveles de una red a la que puedan ser conectados todos los sistemas de control de una empresa.

## Vocabulario técnico

**Baudio:** bit/sg.

Otras unidades son el Kbaudio y el Mbaudio.

La velocidad de transmisión de datos prefijada para los autómatas es 110, 300, 600, 1200, 2400, 9600 o 19.200 baudios.

**CAD:** Computer aided (o assisted) design. Diseño asistido por ordenador.

**CAE:** Computer aided engineering. Ingeniería asistida por ordenador.

**CAM:** Computer aided manufacturing. Fabricación asistida por ordenador.

**CEI:** Normas para la representación de circuitos lógicos

**CIM:** Computer integrated manufacturing: Automatización integrada.

**CNC:** Computer numerically control: Control numérico.

**CPU:** Unidad Central de Proceso.

**CRT:** Tubo de rayos catódicos.

**EAPROM:** Electrically alterable programmable ROM: memoria de sólo lectura alterable eléctricamente en la que puede modificarse el contenido de algunas posiciones seleccionadas de memoria.

**EEPROM:** Electrically erasable programmable read only memory memoria de sólo lectura que puede borrarse con impulsos eléctricos.

**EPROM:** Erasable programmable read only memory. Memoria de sólo lectura programable y borrrable.

**EIA:** Normas para la interconexión de periféricos.

**Firmware:** Conjunto de programas ejecutivos para que el microprocesador realice las tareas propias de procesador del autómata.

**Gate-array:** Matriz de puertas.

**Logic-array:** Matriz lógica.

**Ladder-diagram:** Esquema de contactos.

**MAP:** Protocolo para la automatización de la producción.

**Memoria scratch-pad:** espacio de memoria de almacenamiento temporal intermedio que utilizan los programas ejecutivos.

**NEMA:** Normas para representación de circuitos eléctricos.

**PID:** Módulos de Control PID (Control Proporcional, Integral y Derivativo).

**PLC:** Controlador lógico programable.

**PLD:** Dispositivo lógico programable.

**PLS:** Secuenciador lógico programable.

**PROM:** Programmable read only memory. Memoria programable de sólo lectura.

**Rack:** Bastidor que proporciona un soporte mecánico a los equipos modulares.

**RAM:** Random access memory, memoria de acceso aleatorio.

**ROM:** Read only memory, memoria de sólo lectura.

**Scan:** Ciclo de ejecución o escrutación del programa.

**Scan time:** Tiempo de ejecución o tiempo de ciclo.

**Transmisión de datos half-duplex o full-duplex:** Transmisión de datos

en un sentido o en ambos.

**Transmitters:** Convertidores de señales. Los usuales tienen los siguientes rangos de trabajo: 0 a 5V., 0 a 10V., -10 a +10V., 4 a 20mA.

**Thumbwheel switches:** codificadores rotativos.

**UVPROM:** Ultraviolet reprogrammable read only memory. Memoria de sólo lectura que puede borrarse con radiación ultravioleta.

**Via RS-232C:** Norma de interconexión de periféricos, con enlace serie elaborada por la EIA. La velocidad está limitada a 20 Kbaudios y la distancia de transmisión a 15 m.

**Watch dog:** Vigilancia de ciclo.

## Bibliografía utilizada

- 1.º AUTÓMATAS PROGRAMABLES. Albert Mayol i Badía. Ed. Marcombo
- 2.º AUTÓMATAS PROGRAMABLES. Andre Simon. Ed. Paraninfo
- 3.º CONTROLADORES LÓGICOS Y AUTÓMATAS PROGRAMABLES. Enrique Mandado y otros. Ed. Marcombo.
- 4.º Catálogos de casas comerciales:
  - AEG
  - KLÖCNER MOELLER
  - OMRON
  - SIEMENS
  - SOTEMAT
  - etc....

NOTA: En la actualidad existen otras publicaciones en castellano:

AUTÓMATAS PROGRAMABLES INDUSTRIALES. Arquitectura y aplicaciones. G. Michel. Ed. Marcombo.

AUTÓMATAS PROGRAMABLES EN LA ENSEÑANZA. Alejandro Porras y A.P. Montanero. Ed. Paraninfo



# Prácticas

## Equipo: AUTÓMATA MODICON A020PLUS de AEG

### Características

**Alimentación:** 24V c.c.

**Procesador:** procesador de bit y de palabra 8031

### Memorias:

### de señal o de datos:

- tipo RAM de 2 Kbytes, no protegida contra fallos de tensión
- tipo NVRAM, accesoria para almacenar los datos de la RAM en casos de fallos de tensión
- datos posibles:
  - entradas digitales (bit) I1... I16
  - entradas analógicas (palabra) IWA1... IWA4
  - salidas digitales (bit) Q1... Q16
  - salidas analógicas (palabra) QWA1
  - marcas (bits)
    - M1...M120 de aplicación libre
    - M121..M128 reservadas
  - marcas (palabras)
    - MW1...MW50 de aplicación libre
    - MW51..MW60 protegidas
  - temporizadores T1... T8 con cadencia de 100ms
  - T9... T16 con cadencia de 25ms

contadores C1... C16 con un margen de 1 a 65535 impulsos  
valores constantes V0.... V1023

### **de programa de aplicación o memoria de usuario:**

- tipo EEPROM de 2 Kbytes, asegurada contra fallos de tensión y con capacidad de 896 instrucciones que ocupan igual cantidad de memoria.  
En ella se memorizan los valores teóricos de temporizadores y contadores.
- tipo EEPROM para archivo del programa de aplicación.

### **de sistema:**

- tipo EPROM para el programa ejecutivo (firmware).

### **Entradas:**

16 binarias (I1..I16) con:

señalización por led  
nivel de señal de 24V

4 analógicas (IWA1..IWA4) con:

nivel de entrada de 0 a 10V  
resolución de 8 bits, equivalencia  
de 10V -> 255 (11111111)

### **Salidas:**

16 binarias (Q1..Q16):

contactos de relé, con tensión  
admisible de 60V en c.c. y 220V en  
c.a.

1 analógica (QWA1):

con tensión de salida de 0 a 10V  
resolución 8 bits y equivalencia  
de 10V -> 255 (11111111)

### **Comunicación:**

Via RS-232C  
velocidad ajustable hasta 9600 baudios (bits/sg).

formato: 1 bit de arranque  
7 bits de datos  
1 bit de paridad  
1 bit de parada

## Tiempos:

Un valor típico de ejecución de programa en este autómata es de 30 ms.

## Software:A020FUPE

### Utilización:

1.º Cargamos el **Sistema Operativo**

2.º Cargamos **A020FUPE**

Nos aparece el menú principal :

on-line  
off-line  
preselección

3.º Elegimos la opción **off-line**, pulsando F2

Encontramos el siguiente menú:

escribir o modificar  
documentar  
archivar  
comentar

1 Elegimos la opción **archivar**, en la que podemos:

leer del disco  
escribir en disco  
borrar archivos  
hacer un listado de archivos

2 Recuperamos un programa archivado con el nombre TEMPORIZ

Observamos que hay archivos .AWL —programa—  
.DOC —comentarios—  
.UEB —cabeceras de línea—

Cargamos los tres archivos correspondientes a nuestro ejemplo.

3 Volvemos al menú off-line, pulsando F9 y observamos el programa, para lo que entramos en la opción **escribir o modificar**, pulsando F1 y posteriormente F2.

Podemos analizar el funcionamiento del programa, con la opción de **test o chequeo**, (pulsar F5), y dar valores previamente a los temporizadores que intervienen, (pulsar F3).

4 Volvemos al menú off-line y, pulsando F4, elegimos la opción de **comentarios al programa**; con F2, podemos hacer comentarios a los diferentes operandos y, con F3, observar los ya hechos.

5 Volvemos al menú off-line y, con F2, elegimos la opción **documentar el programa**, mediante la cual podemos obtener el diagrama de funciones correspondiente, rellenar una página de referencia al programa, así como imprimir éste con los comentarios respectivos y su diagrama de funciones.

4.º Volvemos al menú principal y elegimos la opción **on-line**.

El autómata debe estar alimentado

Nos aparece el menú on-line:

operación en on-line

transferir un programa del ordenador al autómata

transferir un programa del autómata al ordenador

cambiar la velocidad de trabajo

observar el funcionamiento del programa mediante el

diagrama de funciones, con el indicador de estatus.

### Realizamos las siguientes aplicaciones:

1 Comprobamos la práctica efectuada anteriormente en off-line, transfiriéndola al autómata, para lo que previamente cargaremos cada archivo de nuestro disco de trabajo desde la opción off-line y después pulsaremos F3 desde la opción on-line.

2

Realizamos prácticas operando en on-line (pulsamos F1).

En esta opción hemos de hacer las siguientes consideraciones acerca de las funciones de uso:

**FUS**, indicación que aparece cuando hemos establecido comunicación con el autómata y cuando tenemos que introducir alguna función.

**HE**, paramos el programa grabado en la memoria de programa.

**STN**, borramos el programa que esté en la memoria. Nos pide confirmación del borrado.

**M**, iniciación del programa. Al teclearlo nos aparece la indicación de dirección ADR.

A continuación introducimos el programa, que siempre terminaremos de la siguiente forma:

**PE**, última instrucción

**E**, final de la introducción del programa. Al teclearlo nos aparece FUS.

**PREC**, emisión de distintivos de prueba para chequear el programa. En el arranque y durante la marcha del programa se comprueba el programa de la EEPROM con el distintivo de pruebas localizado en un margen de dicha memoria. En caso de error aparece la indicación **FUS\***

**FX**, indicación de la causa del error en el programa

**AV**, para introducir o modificar valores de contadores y temporizadores.

Con **S**, —set, valor teórico— damos los valores convenientes y nos pide confirmación para grabar esos datos en la memoria EEPROM de programa.

Con **A**, —actual, valor real— leemos los valores que respecto de los consignados con set, tienen en un instante los contadores y temporizadores

**AS**, para observar si están o no activados los diferentes operandos.

**S**, para ejecutar el programa.

## Aplicaciones

### Programas de puertas lógicas

#### PUERTA1

```
A I1
AN I2
= M1
A I3
AN I4
A I5
= M2
A M1
O M2
= Q1
PE
E
```

Este programa responde a la ecuación lógica:

$$(I1 * I2) + (I3 * I4 * I5) = Q1$$

es decir, el paralelo de la serie de las entradas I1 e I2 negada con la serie de las entradas I3, I4 negada e I5; utilizamos las marcas M1 y M2 para reconocer las series.

#### PUERTA2

```
A I1
AN I2
O(
AN I1
A I2
)
= Q10
PE
E
```

Con este programa realizamos una puerta OR exclusiva. Hemos utilizado paréntesis para facilitar la programación.

### PUERTA3

AN I1  
AN I2  
O(  
A I1  
A I2  
)  
= Q10  
PE  
E

Con este programa realizamos una puerta OR exclusiva negada.

### PUERTA4

A I1  
A(  
A I3  
O I2  
)  
A I4  
A(  
A I7  
O I6  
O I5  
)  
A I8  
A(  
A I12  
O I11  
O I10  
O I9  
)  
A I13  
= Q10  
PE  
E

Este programa corresponde a un circuito combinacional cuya función es:

$$I1*(I3+I2)*I4*(I7+I6+I5)*I8*(I12+I11+I10+I9)*I13=Q10$$

## PUERTA5

A I5  
O Q10  
AN I6  
A I7  
AN Q2  
= Q10  
PE  
E

Quando activamos I5 e I7, se activa la salida Q10. Si desactivamos I5 manteniendo I7 permanece Q10; en el instante que pulsamos I6 se desactiva la salida

## PUERTA6: Codificador Decimal/BCD

A I1  
O I3  
O I5  
O I7  
O I9  
= Q1  
A I2  
O I3  
O I6  
O I7  
= Q2  
A I4  
O I5  
O I6  
O I7  
= Q3  
A I8  
O I9  
= Q4  
PE

Si vamos activando las entradas I1, I2, I3, etc., asociadas a los números decimales 1, 2, 3, etc.; respectivamente, observaremos a la salida los valores correspondientes en código BCD

## Programas de tratamiento de datos

### Programas de CARGA Y TRANSFERENCIA

#### CARGA:

```
L V128  
TBW Q1  
PE  
E
```

Cargamos en el acumulador el valor 128 que corresponde a la secuencia 10000000 y lo transferimos a partir de la salida Q1.

En nuestro ejemplo se activa la salida Q8.

El valor constante V puede ser cualquiera comprendido entre 0 y 1023.

#### CAR1

```
LD MW51  
= MW1  
TBB Q1  
PE  
E
```

Cargamos la marca MW51 que corresponde al reloj interno (de 0 a 59 segundos), la asociamos a una palabra MW1 y la transferimos a partir de la salida Q1.

Probar cambiando la salida de la tercera instrucción, por ejemplo Q6, para visualizar los impulsos de conteo desde Q6 a Q11.

#### CAR2

```
LD IWA1  
= MW3  
TBB Q1  
PE  
E
```

Cargamos la entrada analógica IWA1 con un valor determinado, que puede oscilar entre 0 y 255 correspondientes a 0 y 10 voltios respectivamente, por ejemplo 64; asociamos este valor a la palabra MW3 y lo transferimos a partir de la salida Q1.

### **CAR3**

LBB I1  
= MW1  
TBB Q1  
PE  
E

Cargamos una secuencia de ocho bits desde la entrada I1 a la I8, por ejemplo, con la secuencia 11010101 (la entrada a partir de la que cargamos la secuencia corresponde al bit de menor peso), la asociamos a una palabra MW1 y la transferimos a partir de la salida Q1.

Al comprobarlo en on-line puedo observar el valor decimal de la secuencia, en nuestro ejemplo 213, asociado a la marca MW1

Probar cambiando la entrada en la primera instrucción y la salida a partir de la cual se transfiere la secuencia.

Hacer un programa para cargar una secuencia de 16 bits

### **SET DOMINANTE**

A I1  
RL Q7  
AI2  
SL Q7  
PE  
E

Cuando pulsamos I2 se activa Q7 que permanece activada hasta que pulsemos I1, aunque hayamos anulado I2. Si activamos simultáneamente las dos entradas I1 e I2, la salida Q7 permanece activada.

### **RESET DOMINANTE**

A I1  
SL Q5  
A I2  
RL Q5  
PE  
E

Cuando pulsamos I1, se efectúa el set de Q5; esta salida permanece activada hasta que pulsemos I2. Si pulsamos simultáneamente las dos entradas I1 e I2, la salida Q5 permanece desactivada.

## Programas de Comparaciones

### COMP1

```
LBB I1
= MW5
LD MW5
EQ V10
= Q12
PE
E
```

Cargamos una secuencia de ocho bits a partir de I1; si activamos I2 e I4, como la secuencia 00001010 corresponde al valor 10, observaremos que se activa Q12.

Si cargamos otra secuencia distinta, por ejemplo, la 00000101, activando las entradas I1 e I3, Q12 no se activaría.

En on-line podemos observar el valor decimal de la secuencia cargada a la entrada en el contenido de la palabra asociada MW5, utilizando la función AS.

### COMP2

```
LBB I1
= MW5
LD MW5
GT V10
= Q12
PE
E
```

Si activamos las entradas I2 e I3, es decir cargamos la secuencia 00000110 que corresponde al valor decimal 6, la asociamos a la palabra MW5 y comparamos su contenido con el valor 10, comprobaremos que Q12 permanece inactiva; si cargamos la secuencia 010000010 activando I2 e I7, veremos la salida activada ya que esta secuencia corresponde al valor 66.

### COMP3

```
LD IWA1
= MW5
LD MW5
GT V125
= Q16
PE
E
```

Cargamos la entrada analógica IWA1 con un valor determinado que asociamos a la palabra MW5 cuyo contenido comparamos con el valor 125; probaremos con distintos valores de la entrada para comprobar el funcionamiento correcto del programa.

En on-line podemos leer el contenido de la palabra MW5, mediante la función AS

### COMP4

```
LD IWA2
= MW7
LD MW7
GT V100
= Q13
A Q13
SL Q15
A I1
RL Q15
PE
E
```

Si la secuencia contenida en la palabra MW7 corresponde a un valor de la entrada analógica superior a 100, se activará Q13, la cual hace un set de la salida Q15; si activamos la entrada binaria I1 se reseteará Q15.

### COMPRESO

El programa siguiente puede ilustrar la automatización de un sistema calefactor que queremos se active al llegar a una temperatura T1 y se desactive al sobrepasar otra T2, o bien de un compresor que queremos que conecte al alcanzar una presión de X bares y que desconecte cuando ésta sea de Y bares, etc...

LD IWA1	carga en el acumulador una entrada analógica
EQ V150	si es igual a la constante 150
= M1	asóciala a la marca M1
LD IWA1	carga en el acumulador la entrada analógica
LT V150	si es menor que el valor 150
= M2	asóciala a la marca M2
A M1	si está activada cualquiera de las marcas, por
O M2	cumplirse alguna de las condiciones anteriores
= Q5	activa la salida Q5
LD IWA1	carga la entrada analógica
EQ V200	si es igual a la constante 200
= M3	asóciala a la marca M3
LD IWA1	carga la entrada analógica
GT V200	si su valor es mayor que 200
= M4	asóciala a la marca M4
A M3	si se cumple alguna de las dos condiciones
O M4	anteriores, es decir el paralelo de las marcas
RL Q5	desactiva la salida Q5
PE	
E	

Podemos modificar el programa anterior introduciendo unas alarmas, mediante el activado de unas salidas, como aviso de que la entrada sea menor o igual que 125 y mayor o igual que 225

LD IWA1	carga la entrada analógica
EQ V125	si es igual a la constante 125
= M5	asóciala a la marca M5
LD IWA1	carga la entrada analógica
LT V125	si es menor que 125
= M6	asóciala a la marca M6
A M5	si se cumple alguna de estas dos
O M6	condiciones
= Q6	activa Q6
LD IWA1	carga la entrada analógica
EQ V225	si es igual a 225
= M7	asóciala a la marca M7
LD IWA1	carga la entrada analógica
GT V225	si es mayor que 225
= M8	asóciala a la marca M8
A M7	si se cumple alguna de las dos

O M8                    condiciones anteriores  
 = Q7                    activa la salida Q7  
 PE  
 E

### COMP5

LD V13  
 GT CAW1  
 = Q1  
 LD V15  
 GT CAW1  
 = Q2  
 LD V33  
 GT CAW1  
 =Q3  
 PE  
 E

Si chequeamos este programa en off-line, damos al contador C1 un valor, por ejemplo 14, y observaremos que se activan las salidas Q2 y Q3. Si comprobamos directamente en on-line el programa, daremos el valor de consigna CAW1 con la función AV.

## Programas de funciones aritméticas

### SUMA

LD V100  
 ADD V123  
 = MW12  
 PE  
 E

En on-line podemos comprobar que se ha realizado la suma, viendo el contenido de la palabra MW12 utilizando la función **AS**.

### RESTA

LD V1022  
 SUB V1000

```
=MW12
= Q3
PE
E
```

Cuando realizamos la resta A-B, siendo  $B > A$ , el resultado corresponde a la diferencia entre el valor máximo de la constante (65536) y el resultado de B-A.

### RESTA1

```
LD V500
SUB V600
= MW22
= Q7
PE
E
```

### PRODUCTO

```
LD V200
MUL V7
= MW10
TBW Q1
PE
E
```

Cargamos en el acumulador el valor 200, lo multiplicamos por 7 y el resultado se lo asociamos a la palabra MW10. Añadimos la instrucción TBW Q1 (transferir la secuencia de 16 bits) para observar el resultado.

### DIVISIÓN

```
LD V1000
DIV V50
= MW8
PE
E
```

Al igual que en el programa anterior, podemos transferir el resultado a partir de una salida para visualizar éste.

## Programas con TEMPORIZADORES

### TEMP1: Retardo a la conexión

```
A I1
= T1
A T1
= Q10
PE
E
```

La entrada I1 activa al temporizador T1 y transcurrido el tiempo asignado a T1 se activa Q10.

Hacemos una modificación al anterior:

### TEMP2.

```
A I1
= T1
= Q13
A T1
= Q10
PE
E
```

Al activar I1 se activa Q13 y el temporizador T1, al cabo del tiempo consignado en éste se activa Q10

### TEMP3

```
A I1
= T1
A T1
= Q10
A I2
= Q16
= T2
A T2
AN I1
RL Q10
PE
E
```

La entrada I1 activa al temporizador T1 el cual activa la salida Q10; al introducir la entrada I2 se activa la salida Q16 y el temporizador T2. Cuando eliminamos la entrada I1 y transcurre el tiempo asignado al segundo temporizador se resetea la salida Q10.

#### TEMP4

```

A I3
O I4
O Q1
A I1
A I2
AN T1
= Q1
A Q1
= T1
A Q1
= Q11
AN Q1
= Q12
PE
E
    
```

Con este programa se autorretiene la salida Q1 durante un tiempo T1. Cuando lo ejecutamos, observamos que se activa la salida Q12; al activar I3 o I4 se desactiva Q12, activándose Q1 y Q11. Aunque eliminemos las entradas I3 o I4, permanece activada la salida Q1 durante el tiempo asignado a T1, pasado el cual vuelve a activarse la salida Q12. Mientras esté activa Q1 lo estará Q11.

Podemos hacer una modificación del programa anterior haciendo que la salida Q1 se retenga durante el tiempo T1, una vez que un contador haya contado un número de pulsos determinado, o bien, como en el programa anterior, se haya activado una entrada.

#### TEMDES

```

A I1
= Q1
AN I1
= T1
A T1
    
```

```

= Q10
PE
E

```

Con este programa hacemos un retardo a la desconexión; al desactivar I1 y, transcurrido el tiempo consignado en T1, se activa la salida Q10.

### TEMPORIZ

```

A I1
= T1
A T1
SL Q10
AN I1
= T2
A T2
RL Q10
PE

```

Este es un programa de retardo a la conexión y a la desconexión; en efecto, al activar T1, mediante la entrada I1, producimos un retardo a la conexión, ya que para que se produzca el set (activado) de Q10, debe transcurrir el tiempo consignado. Con el temporizador T2, que activamos al desactivar I1, conseguimos un retardo a la desconexión, ya que el reset (desactivado) de la salida Q10, sólo se producirá transcurrido el tiempo indicado en T2, después del desactivado de la entrada I1.

### TEMCONT

```

A I5
O Q1
= Z1
A I3
= P1
A C1
O I4
O Q1
A I1
A I2
AN T1
= Q1
A Q1

```

```

= Q11
AN Q1
= Q12
A Q1
= T1
PE
E

```

Cuando haya salida Q1, o bien demos un pulso en I5, ponemos a cero el contador. Cuando activamos I1 e I2 podemos pulsar la entrada I4 y el funcionamiento será el indicado antes, o bien dar un número de pulsos consignado en el valor de C1 y observaremos un mismo comportamiento.

### Programa para observar las marcas internas especiales del autómeta

#### MARCAS

```

A M125
AN I2
A I1
AN I3
O(
AM 126
AN I1
A I2
AN I3
)
O(
AM 127
AN I2
A I3
AN I1
)
O I4
= Q10
PE
E

```

En la salida Q10 observamos los pulsos de 1.25, 2.50 y 5 Hz correspondientes a las marcas M125, M126 y M127 cuando introducimos las entradas I1, I2 e I3 respectivamente.

## ALARMA

```
LD MW51
= MW1
LD MW1
GT V30
= Q8
PE
E
PREC
```

Cargamos en el acumulador el valor del reloj interno y lo asociamos a la palabra MW1, a través de la cual se activa Q8 cuando el reloj tiene un valor superior a 30.

Con la función **S** ejecuto el programa observando que la salida se activa al ser la marca MW51 mayor que 30 y se desactiva en caso contrario.

Para efectuar la comprobación utilizamos la función **AS** y vemos en la dirección MW1 el valor almacenado.

## Programas de contadores

**CONT1.** Contador hacia atrás.

```
A I3
= Z1
A I1
= P1
A C1
= Q2
PE
E
PREC
```

Cuando damos un pulso con I3 borramos el contador permitiéndole que inicie su función; introducimos la entrada contadora I1 que activa P1 para que pueda contar el contador. Cuando el número de pulsos efectuados sea igual al número consignado en C1, se activa Q2.

## CONTADOR

```
A M1
= Z1
```

```

A I1
= P1
A C1
SL M1
A C2
A M127
RL M1
A C2
A M127
= Z2
A M1
A M127
= Q10
A Q10
= P2
PE
E

```

Cuando en la entrada I1 damos el número de pulsos especificado en el contador C1, observamos que se activa la salida Q11, dando el número de pulsos indicado en C2.

Con la marca M1 se pone a cero el contador C1, y con la marca M127, el C2, una vez que se ha activado la salida Q10 el número de pulsos indicado en C2.

Podemos introducir una modificación en este programa para visualizar cuándo se han dado los pulsos convenientes con I1; basta añadir la instrucción que corresponde a la activación de una salida, Q3 por ejemplo, que nos lo indique y que nos sirve para borrar el contador C1.

### CONTADO1

```

A Q3
= Z1
A I1
= P1
A C1
SL Q3
A C2
A M127
RL Q3
A C2

```

ALARMIA A M127  
= Z2  
A Q3  
A M127  
= Q10  
A Q10  
= P2  
PE

## CONTAD2

A C2  
= Z2  
A I1  
= P2  
A Q10  
JI 10  
A C2  
SL Q10  
JI 12  
A C2  
RL Q10  
PE  
E

Cuando damos con I1 el número de pulsos indicado en C1, se activa la salida Q10; al repetir el mismo número de pulsos de entrada se desactiva la salida.

## DIVISOR DE FRECUENCIA

A C5  
= Z5  
A I13  
= P5  
A Q9  
JI 10  
A C5  
SL Q9  
JI 12  
A C5  
RL Q9

PE  
E  
PREC

Introducimos con la entrada I13 una serie de pulsos; cada vez que el número de éstos coincida con el número asignado al contador C5, se activa la salida Q9. Al introducir de nuevo el mismo número de pulsos, se desactiva Q9 y así sucesivamente. Podemos ajustar el divisor al valor que nos interese sólo modificando con **AV** el valor de C5.

Con las dos primeras instrucciones ponemos el contador a cero, cuando haya contabilizado el número de pulsos prefijado; con la tercera, ordenamos la entrada de pulsos como entrada del contador; con la cuarta, permitimos que el contador cuente, activando la salida o desactivándola.

Podemos usar las marcas internas M125, M126 o M127 que nos dan pulsos de 1.25, 2.50 y 5 HFz, respectivamente, como entrada.

### CONTADO3

A I4  
A T1  
= M1  
A M1  
= Z1  
A M1  
= Z2  
A M1  
= Z3  
A I1  
= P1  
A C1  
= Q1  
A I2  
= P2  
A C2  
= Q2  
A I3  
= P3  
A C3  
= Q3



```

A T1
A M125
= Q8
A Q1
A Q2
A Q3
= Q6
A Q6
= T1
PE

```

Con este programa, una vez consignados los valores a los tres contadores y al temporizador, si está I4 activa cuando demos los pulsos correspondientes a I1, I2 e I3, y se activen Q1, Q2, Q3 y por último Q6 para indicar que han terminado su trabajo los tres contadores, transcurrido el tiempo T1, se desactivan las salidas. Pero si la entrada I4 está desactivada una vez activas las cuatro salidas y transcurrido el tiempo T1, la salida Q8 se mantiene parpadeante hasta que pulsemos I4.

La acción puede comenzar desconectando previamente la I4.

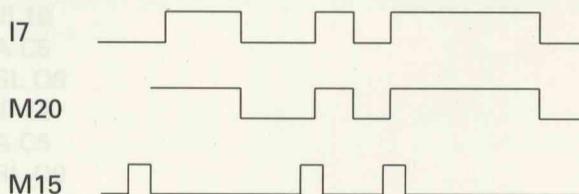
**FLANCO:** Este programa permite detectar cuándo la entrada cambia de 0->1, es decir, el reconocimiento del flanco de subida

```

A I7
AN M20
= M15
A I7
= M20
PE
E

```

Cuando ejecutamos este programa, si inicialmente I7 está desactivada y la ponemos a 1, observamos que la marca M15 se activa. Como al estar presente I7 se activa la marca M20 ésta desactiva a M15 indicándonos, por tanto, el paso de 0 a 1 de la entrada.



Para observar mejor el funcionamiento de este programa, al probarlo en on-line, podemos introducir unas instrucciones que nos activen una salida, en función de la marca correspondiente.

A I7  
 AN M20  
 = M15  
 A I7  
 = M20  
 A M15  
 SL Q1  
 AN M15  
 = Q2  
 A M20  
 SL Q7  
 AN M20  
 =Q8  
 PE  
 E  
 PREC

**UP/DOWN:** Contador ascendente y descendente.

LBB Q1           carga la salida actual  
 A I1               compruebo si se introducen  
 AN M3            pulsos con un reconocimiento  
 = M2             de flanco (instrucciones 2 a 6)  
 A I1  
 = M3  
 AN M2            si no hay nuevo impulso  
 JI 13             salta a la dirección 13  
 AN I3             si quiero decontar  
 JI 12             salta a la dirección 12  
 ADD V2           suma 2  
 SUB V1           resta 1  
 AN I2             si no quiero borrar  
 JI 16             salta a la dirección 16  
 LD V0             carga 0 en el acumulador  
 TBB Q1           transfiérelo a la salida  
 PE  
 E  
 PREC

Este contador necesita: una señal de entrada de pulsos I1  
 una señal de borrado de pulsos I2  
 una señal de control para contar o  
 decontar I3

- 1.º cargo la salida a la entrada para conocer el estado de "conteo."
- 2.º si hay un nuevo impulso (M2=1) y:  
 quiero contar (I3=1), sumo 2 y resto 1, es decir, sumo 1.  
 quiero decontar (I3=0), resto 1.
- 3.º si quiero borrar (I2=1), cargo en el acumulador 0 y lo transfiero a la salida.
- 4.º si no quiero borrar (I2=0), transfiero el contenido del acumulador a la salida.

## REGISTRO

AN I8  
 JI 10  
 =N Q11  
 =N Q12  
 =N Q13  
 =N Q14  
 =N Q15  
 =N Q16  
 JI 29  
 A I6  
 AN M15  
 = M105  
 A I6  
 = M15  
 AN M105  
 JI 29  
 A Q15  
 = Q16  
 A Q14  
 = Q15  
 A Q13  
 = Q14  
 A Q12

= Q13  
 A Q11  
 = Q12  
 A I7  
 = Q11  
 PE  
 E  
 PREC

para programar este registro necesitamos:

- una señal de entrada de impulsos I6.
- una señal de borrado I8.
- una señal de validación I7.

Si doy un pulso con I7, al activar la entrada de I6 se activa Q11.

Los sucesivos pulsos dados con I6 se van transfiriendo de Q11 a Q16.

Si mantenemos activa I7, el registro es de almacenamiento.

Con las instrucciones 10 a 14, de reconocimiento de flanco, identificamos los pulsos de entrada.

Si activamos I8, borramos el contenido del registro, es decir, desactivamos las salidas.

## Programas de control de ciclo de ejecución

### SALTOS

A I1  
 JI 5  
 LD V200  
 JI 6  
 LD V100  
 = TSW1  
 PE  
 E

Si está activada la entrada I1, salta a la dirección 5, carga el acumulador con el valor 100 y le asigna ese valor al temporizador. Si I1 está desactivada, sigue en la instrucción 3.<sup>a</sup> y carga el acumulador con el valor 200, salta a la dirección 6 y le asigna ese valor al temporizador.

## **SALTO2.** Programa con instrucciones PE intermedias

```
AN I1
JI 6
A I6
= Q7
PE
= T1
A T1
= Q6
PE
E
```

Si no activamos I1, al cabo del tiempo consignado en T1 se activa Q6; si tenemos activada I1, cuando activemos I6, se activará Q7.

Podemos observar tanto en on-line como off-line las posibles combinaciones de los estados I1 e I6.

### **Simulación del movimiento de un motor**

Los programas siguientes los aplicaremos al funcionamiento simulado de un motor eléctrico, utilizando para ello un módulo de simulación alimentado con 5V. que tiene dos puertas de accionamiento A y B.

Cuando se aplica un 1 lógico a la puerta A el motor gira en sentido antihorario —hacia la derecha—, movimiento que simulamos con el encendido de los leds. Cuando el 1 lógico es aplicado a la puerta B, el giro es en sentido horario —hacia la izquierda—.

La velocidad de giro puede modificarse mediante unas resistencias de ajuste accesibles desde la parte frontal.

**MOTOR1:** El motor gira a la derecha un tiempo T1

```
A I1
= Q1
A I2
= Q3
= T1
A T1
RL Q3
PE
E
```

En este programa al aplicar I1 se activa Q1, con lo que observamos que el motor está conectado; con I2 aplicamos tensión (introducimos un 1 ) en A, permitiendo el giro a la derecha del motor. Cuando haya transcurrido el tiempo consignado en el temporizador T1, el motor se para (se desactiva Q3).

**MOTOR2:** El motor gira a la izquierda un tiempo T2

A I1  
 = Q1  
 A I2  
 = Q5  
 = T2  
 A T2  
 RL Q5  
 PE  
 E

Con la entrada I1, conectamos el motor, Q1 se activa. Con I2 aplicamos un 1 lógico en B, se activa Q5 y el motor gira hacia la izquierda el tiempo consignado con T2.

**MOTOR3:** El motor gira a la derecha un tiempo T1 y a continuación gira a la izquierda un tiempo T2

A I1	A I1
= Q1	= Q1
A I2	A I2
= Q3	= Q3
= T1	= T1
= T2	
A T1	A T1
RL Q3	RL Q3
A T2	A I3
= Q5	= Q5
= T3	= T3
A T3	A T3
RL Q5	RL Q5
PE	PE
E	E

Al introducir la entrada I1 se conecta el motor; con I2 activamos Q3 y T1 con lo que conseguimos que gire a la derecha el tiempo consignado en el temporizador.

Para que el motor empiece a girar hacia la izquierda, transcurrido el tiempo anterior, podemos introducir el temporizador T2 que activa la salida Q5 y el temporizador T3 para establecer el tiempo de giro, o bien, sustituir estas instrucciones por otras que correspondan a la aplicación de una entrada I3 que active Q5 y el temporizador T3. Cuando transcurre el tiempo consignado en este último, se para el motor.

Los tiempos asignados pueden ser:

T1=150  
T2=150  
T3=300

OPERACIONES	OPERANDOS	EFECTO
A	I, Q, M, T, C	Puerta serie
O		Puerta paralelo
AN		Puerta serie negada
ON		Puerta paralelo negada
A(		Puerta serie abrir paréntesis
O(		Puerta paralelo abrir paréntesis
)		Cerrar paréntesis
)N		Cerrar paréntesis negado
=	Q, M, QWA, MW TAW, TSW, CAW, CSW	Asignación positiva
=N	Q, M	Asignación negada
SL	"	Set
RL	"	Reset
OPERACIÓN	OPERANDOS	EFECTO
=T		Temporizador
=Z		Borrado del contador
=P		Entrada de impulsos a contar
LD	QWA, IWA, V, MW TAW, TSW, CAW, CSW	Carga en el acumulador

<b>LBB</b>	<b>I, Q, M</b>	<b>Carga 8 bits</b>
<b>LBW</b>	<b>"</b>	<b>Carga 16 bits</b>
<b>TBB</b>	<b>Q, M</b>	<b>Transfiere una secuencia de 8 bits</b>
<b>TBW</b>	<b>"</b>	<b>Transfiere una secuencia de 16 bits</b>
<b>GT</b>	<b>QWA, IWA, V, MW</b>	<b>Mayor que</b>
<b>EQ</b>	<b>TAW, TSW, CAW, CSW</b>	<b>Igual que</b>
<b>LT</b>	<b>"</b>	<b>Menor que</b>
<b>OPERACIÓN</b>	<b>OPERANDOS</b>	<b>EFECTO</b>
<b>ADD</b>	<b>QWA, IWA, V, MW</b>	<b>Sumar</b>
<b>SUB</b>	<b>TAW, TSW, CAW, CSW</b>	<b>Restar</b>
<b>MUL</b>	<b>"</b>	<b>Multiplicar</b>
<b>DIV</b>	<b>"</b>	<b>Dividir</b>
<b>JI</b>	<b>N.º de dirección de</b>	<b>Salto</b>
<b>PE</b>		<b>Fin de programa</b>
<b>M125</b>		<b>Pulso de 1,25 Hz</b>
<b>M126</b>		<b>Pulso de 2,50 Hz</b>
<b>M127</b>		<b>Pulso de 5 Hz</b>
<b>M128</b>		<b>Señal a 1</b>
<b>MW51</b>		<b>Segundos (0-59)</b>
<b>MW52</b>		<b>Minutos (0-59)</b>
<b>MW53</b>		<b>Horas (0-23)</b>
<b>MW54</b>		<b>Días (0-65.353)</b>



# Índice

	<i>Página</i>
INTRODUCCIÓN A LA ROBÓTICA .....	305
¿QUÉ SE ENTIENDE POR ROBÓTICA? .....	305
Robótica industrial .....	305
Robótica exploración .....	307
Robótica de electromedicina .....	308
Robótica didáctica .....	308
EL ROBOT .....	309
Definición de robot .....	309
Características del robot .....	309
Partes fundamentales del robot .....	310
Funcionamiento del robot industrial .....	311
MODELO DE UN ROBOT .....	313
Arquitectura .....	313
Grados de libertad .....	313
Movilidad de un robot .....	313
Parte mecánica .....	315
Parte electrónica .....	316
Parte de control .....	316
LOS MOTORES Y ACCIONADORES DE LOS ROBOTS .....	317
Los accionadores hidráulicos .....	317

## Introducción a la robótica



# Índice

	Página
INTRODUCCIÓN A LA ROBÓTICA .....	305
¿QUÉ SE ENTIENDE POR ROBÓTICA? .....	305
Robótica industrial .....	305
Robótica exploración .....	307
Robótica de electromedicina .....	308
Robótica didáctica .....	308
EL ROBOT .....	309
Definición de robot .....	309
Características del robot .....	309
Partes fundamentales del robot .....	310
Funcionamiento del robot industrial .....	311
MODELO DE UN ROBOT .....	313
Arquitectura .....	313
Grados de libertad .....	313
Movilidad de un robot .....	313
Parte mecánica .....	315
Parte electrónica .....	316
Parte de control .....	316
LOS MOTORES Y ACCIONADORES DE LOS ROBOTS .....	317
Los accionadores hidráulicos .....	317

	<u>Página</u>
Los accionadores neumáticos .....	317
Los accionadores eléctricos .....	317
LOS CAPTADORES Y SENSORES .....	319
Captadores dinámicos .....	319
Captadores de desplazamiento o posición .....	319
Captadores de velocidad .....	319
Captadores de aceleración .....	320
Captadores cinemáticos .....	320
Captadores de esfuerzo .....	320
Sensores de entorno .....	320
Sensores de proximidad .....	321
Sensores táctiles .....	321
Sensores de visión .....	321
EL CONTROL DEL ROBOT .....	323
Los servosistemas .....	323
El control de los servosistemas .....	323
Con motores eléctricos de C. C. ....	323
Con motores eléctricos de corriente alterna .....	323
El control cinemático .....	324
El control dinámico .....	324
TRAYECTORIAS DE POSICIONAMIENTO .....	325
Modo de aprendizaje del robot .....	325
Método de aprendizaje manual .....	326
Trayectorias tipo .....	326

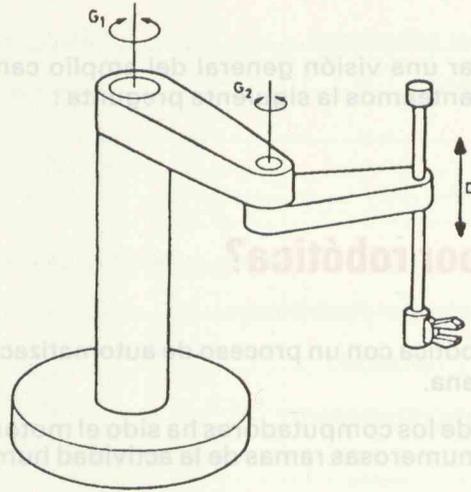
Control de trayectorias por ordenador .....	328
Con motores paso a paso .....	328
Con motores de C/C.....	329
Generación de trayectorias .....	329
<b>SIMULACIÓN</b> .....	333
Parámetros mecánicos y electro-electrónicos.....	333
Reguladores de posición y velocidad .....	333
Puntos inicial y final de desplazamiento.....	333
Tipos de trayectorias .....	333
Tiempo de simulación .....	334
Pruebas de simulación .....	334
<b>UN ROBOT DIDÁCTICO</b> .....	337
Descripción mecánica.....	337
Especificaciones eléctricas.....	338



(FIGURA 1) Robot industrial (tipo S.C.A.R.A.)







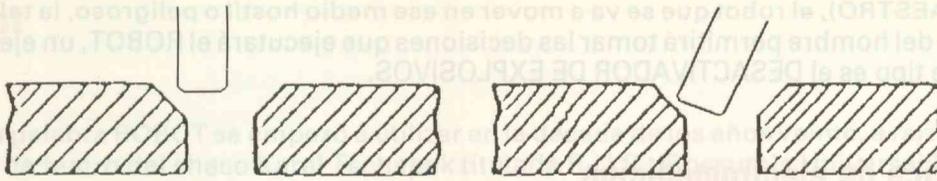
(FIGURA 2: Estructura de un Robot industrial tipo S.C.A.R.A.)

Los Sistemas Flexibles de Producción, que han incorporado robots, fueron desbancando a los modos tradicionales de producción, por presentar:

- Un producto acabado de mejor calidad.
- Una mayor fiabilidad en el mantenimiento de esta calidad.
- Mayor flexibilidad en la adaptación de la producción a la demanda.
- Mayor potencial de reconfiguración de la unidad de producción al pasar de fabricar un producto a otro similar.

### Robots de ensamblado

Dentro de la robótica industrial, estos robots se caracterizan por su elevada precisión (repetitividad), aceleraciones muy fuertes (que a veces superan la de la gravedad) y velocidades elevadas, su estructura suele ser lo más simple que permita la máxima carga prevista en la pinza, con el fin de reducir la inercia.



(FIGURA 3: Achaflanado en montajes de ensamblado)

## Robótica de exploración

Cuando se deben realizar trabajos en lugares hostiles o peligrosos para el hombre, esta parte de la robótica los soluciona sin la intervención humana in situ.

Se suele utilizar de forma general en industrias molestas, nocivas y peligrosas, y en particular en:

- Submarinismo.
- Centrales nucleares.
- Industrias peligrosas o insalubres.
- Viajes espaciales.

La podemos dividir en dos grandes grupos:

### Robótica programada

La tarea a realizar se programa previamente, aunque hoy en día se diseñan robots que le permiten conocer su entorno, se utilizan para operaciones muy elementales como toma de muestras en la Luna, inspección puntual de válvulas de conducción en el medio submarino, traslado de piezas a temperaturas elevadas, etc.

Aunque el robot programado resulta autónomo, a veces no resulta operativo en circunstancias en las que se le obliga a tomar decisiones no programadas, es en estos casos cuando se hace uso de la robótica controlada.

### Robótica controlada

Consiste en controlar a distancia, desde un PUESTO DE MANDO

(MAESTRO), el robot que se va a mover en ese medio hostil o peligroso, la telepresencia del hombre permitirá tomar las decisiones que ejecutará el ROBOT, un ejemplo de este tipo es el DESACTIVADOR DE EXPLOSIVOS.

## Robótica de electromedicina

Un campo de gran desarrollo es el de la robótica médica, que ha permitido mejorar las condiciones de vida de parapléjicos, tetrapléjicos y amputados, mediante:

- órtesis (estructuras motorizadas)
- telétesis (control a distancia de órtesis)

## Robótica didáctica

La implantación en nuestra sociedad de las Nuevas Tecnologías, concretamente la robótica, ha llevado a los Sistemas de Enseñanza de vanguardia a contemplarla en sus diferentes niveles.

## El robot

La palabra ROBOT se empezó a utilizar en la década de los años veinte, a raíz de una obra de teatro del checo Karel Tschapek titulada R.U.R. (Rossum's Universal Robot). Esta obra pone en juego pequeños seres artificiales antropomorfos, que responden perfectamente a las órdenes de su maestro. Estos seres llevan el nombre de robot, del checo "robota", término idéntico al ruso y que significa "trabajo forzado".

El término ROBOT se utiliza en la mayoría de las lenguas, lo que confirma su reciente creación, si bien su objetivo es tan viejo como la existencia del hombre: es un sistema que reemplaza al hombre en todo aquello que no quiere o no puede hacer él mismo, pero de lo que tiene necesidad o interés, sin presentarle ninguna competencia a la hora de tomar cualquier decisión.

### Definición de robot

No hay acuerdo para definir el ROBOT, pero las propuestas más recientes son las del:

- a) Oxford English Dictionary, que lo define como:  
"un aparato mecánico que se parece y hace el trabajo del ser humano".
- b) Robot Institute America  
"un manipulador reprogramable y multifuncional concebido para transportar materiales, piezas, herramientas o sistemas especializados, con movimientos variados y programados, con la finalidad de ejecutar tareas diversas".
- c) Diccionario de la Real Academia Española  
"ingenio electrónico que puede ejecutar automáticamente operaciones o movimientos muy varios"

Observando las aplicaciones de la robótica, se entiende porqué todavía no ha sido aceptada ninguna definición.

### Características del robot

- a) Versatilidad:  
Es la capacidad de modificar su estructura para desarrollar tareas diversificadas (flexibilidad).

b) Autoadaptabilidad al entorno:

Propiedad ligada a los sentidos artificiales, que le permiten al robot ser consciente de su entorno (p.e. salvar obstáculos en su trayectoria).

c) Repetitividad:

Está relacionada con la precisión con la que un robot alcanza el posicionamiento inicial después de haber desarrollado un desplazamiento.

Nº de grados de libertad		
Tipo de accionamiento	$a$	Eléctrico con motores de c/c.
	$\beta$	
	$r$	Eléctrico con motor paso a paso
	$z$	Neumático, opcionalmente eléctrico c/c.
Repetitividad de posicionamiento		0.025 milímetros
Velocidad angular max.		225 grados /segundo
Velocidad lineal max.		2 metros/segundo
Capacidad carga max.		20 kilogramos
Límite de actuación de las articulaciones	$a$	-90 grados ; 110 grados
	$\beta$	0 grados ; 160 grados
	$r$	360 grados
	$z$	150 milímetros (máximo)
Peso del robot		75 kilogramos

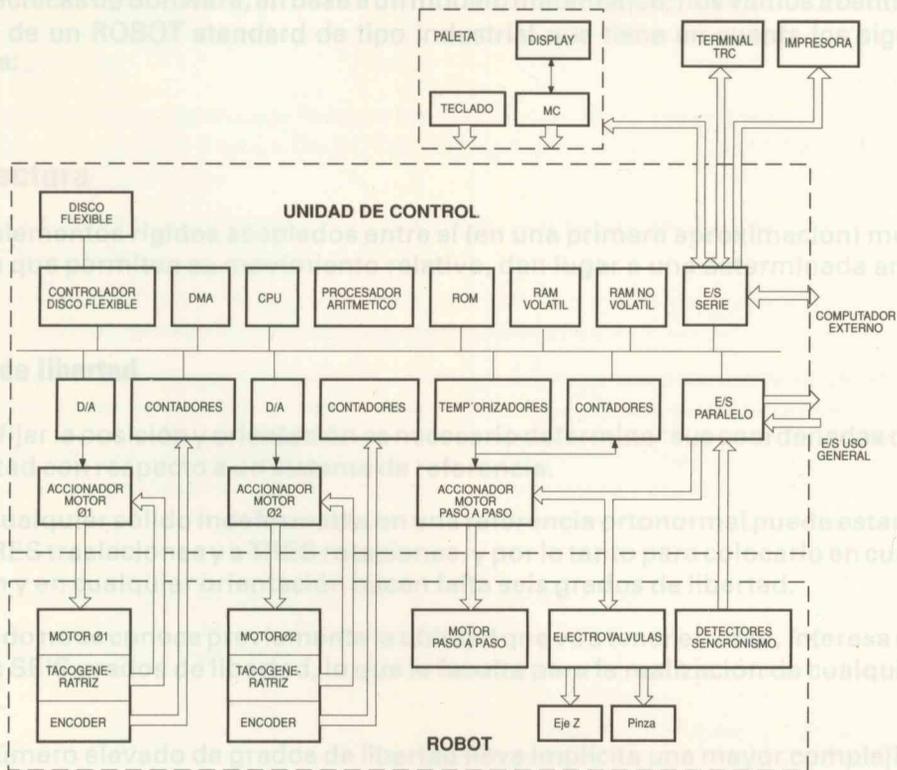
(FIGURA 4: Cuadro de características de un robot del tipo S.C.A.R.A.)

## Partes fundamentales del robot

a) El sistema mecánico articulado, con sus barras, reductores, planetarios, etc.

b) El sistema eléctrico-electrónico, con sus motores, amplificadores, convertidores, etc.

- c) El sistema de control, con sus reguladores, captadores, decodificadores (encoders), etc.
- d) La unidad central de proceso, con sus autómatas, microprocesadores, coprocesadores matemáticos, etc.



(FIGURA 5: Diagrama de bloques de un sistema Robot)

## Funcionamiento del robot industrial

Podemos resumir el funcionamiento de un robot como la respuesta a una tarea programada, a través de un algoritmo de control, que proporciona unas señales que amplificadas suficientemente activan los accionadores de potencia y desplazan al robot siguiendo una trayectoria controlada por la unidad central de proceso, que es informada en cada instante del entorno que rodea al sistema.





## Modelo de un robot

Para diseñar, analizar y controlar un ROBOT, se desarrollan SIMULADORES, utilizando técnicas de Software, en base a un modelo matemático; nos vamos a centrar en el modelo de un ROBOT standard de tipo industrial que tiene en cuenta los siguientes aspectos:

### Arquitectura

Los elementos rígidos acoplados entre sí (en una primera aproximación) mediante uniones que permiten su movimiento relativo, dan lugar a una determinada arquitectura.

### Grados de libertad

Para fijar la posición y orientación es necesario determinar sus coordenadas o grados de libertad con respecto a un sistema de referencia.

Así cualquier sólido indeformable en una referencia ortonormal puede estar sometido a TRES traslaciones y a TRES rotaciones, y por lo tanto para colocarlo en cualquier posición y en cualquier orientación hacen falta seis grados de libertad.

Cuando no se conoce previamente la utilidad que va a tener el robot, interesa dotarle de estos SEIS grados de libertad, lo que le faculta para la realización de cualquier tipo de tarea.

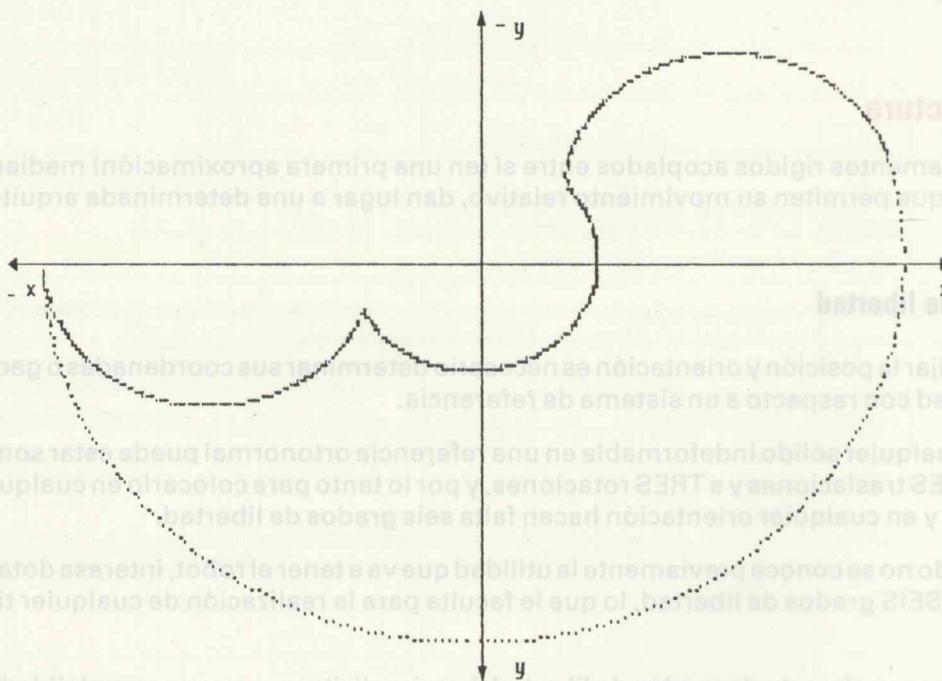
Un número elevado de grados de libertad lleva implícita una mayor complejidad en el algoritmo de control y generación de trayectorias y por tanto requiere un equipo más potente de unidad central de proceso para trabajar en tiempo real.

### Movilidad de un robot

Conviene dejar claro que el número de articulaciones de un robot no coincide con el número de grados de libertad, la estructura del robot puede tener muchas articulaciones, sin embargo el máximo número de grados de libertad será de seis.

Hoy día existen sistemas robotizados que pueden controlar hasta 12 ejes, los propios del robot conjuntamente con otros integrados en el sistema.

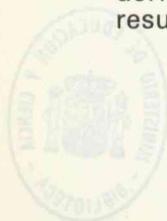
Un robot que trabaja en un plano vertical con tres articulaciones sólo tiene dos grados de libertad, sin embargo la movilidad que le proporciona la tercera articulación, posibilitará el que se generen y controlen trayectorias de una forma más sencilla.



(FIGURA 6: Campo de acción del robot)

Si bien el modelo matemático se complica, se utiliza esta articulación para generar algoritmos más sencillos de procesar y generar trayectorias más rápidas.

En cualquier caso, siempre que el robot genere trayectorias partiendo de unos datos, hay que tender a que la tarea se realice con el robot de mínimo número de grados de libertad, de esta forma el sistema de ecuaciones de estado que define el modelo matemático del robot será más sencillo de procesar, y además conseguir una precisión determinada resultará mecánicamente más fácil.





## Parte electro-electrónica

Además de los actuadores, en el caso de motores eléctricos, son componentes de esta parte, los amplificadores de potencia, los convertidores digitales/analógicos, los amplificadores de la señal de error y otros específicos del robot concreto de que se trate.

El modelo matemático correspondiente a esta parte tendrá en cuenta la resistencia y coeficiente de autoinducción de cada motor, así como la ganancia de los amplificadores de señal error y de potencia con los correspondientes limitadores de tensión y de potencia.

## Parte de control

El servosistema de control de los robots, en líneas generales, debe aunar las siguientes propiedades:

- a) Captar con precisión y rapidez las variables a controlar, generalmente posición y velocidad.
- b) Disponer de un sistema de reguladores que reconduzcan al sistema-robot hacia las variables de referencia con la rapidez necesaria.

La medida de posición se realiza por medio de resistencias que varían linealmente con el giro, a través de discos ranurados en su periferia (encoders) con dispositivos optoelectrónicos y contadores de impulsos.

La medida de velocidad angular de un eje determinado se realiza por medio de tacómetros que se caracterizan por generar una tensión continua proporcional a la velocidad de giro de dicho eje.

Los reguladores, también llamados correctores, están formados, en la mayoría de los casos, por circuitos analógicos basados en amplificadores operacionales, que con muy poca potencia trabajan con la señal de error existente (diferencia entre las magnitudes de la variable de referencia a controlar y la real instantánea) en cada periodo de muestreo que suele tomar el valor standard de 20 milisegundos.

Además de la precisión todo sistema de control se caracteriza por su estabilidad y la respuesta en régimen transitorio.

# Los motores y accionadores de los robots

Existe una gran variedad de accionadores empleados en robótica, dependiendo de la potencia a desarrollar por la articulación y de las características de la aplicación, se utilizan:

## Los accionadores hidráulicos

Están basados en la aplicación de la mecánica de fluidos a la robótica industrial, se componen en la mayoría de las ocasiones de motores hidráulicos lineales de simple o de doble efecto, aunque los hay rotativos con cilindrada axial, todos controlados por servoválvulas muy precisas.

Trabajan con aceites que proporcionan rozamientos viscosos muy bajos.

Se caracterizan por ser muy silenciosos debido a su rigidez hidráulica, pero están siendo sustituidos por accionadores eléctricos de mantenimiento más económico.

## Los accionadores neumáticos

Cuando el robot además de mover grandes pesos, se necesite que los realice con gran rapidez, podemos pensar en la alternativa que ofrecen estos dispositivos.

El funcionamiento de estos accionadores es muy similar al de los hidráulicos, en estos el fluido es el aire.

Necesitan de grandes compresores para el suministro de energía neumática y son instalados en la sala de motores para mantener los niveles de ruido permisibles, como consecuencia de los escapes de aire inevitables.

## Los accionadores eléctricos

Los motores de continua, fácilmente controlables, están corrientemente implementados en gran parte de la robótica industrial y en la totalidad de los sistemas que utilizan lazos cerrados de realimentación para su control.

La linealidad entre el par motor en el eje y el voltaje de inducido, con excitación constante, es la mejor de sus características de control.

En los casos en que no sea necesario generar trayectorias (p.e. alternativa por obstrucción de algún elemento), y en aplicaciones que no requieran control, se utilizan frecuentemente los motores paso a paso. Se caracterizan por girar un ángulo fijo al aplicarles un impulso eléctrico, de tal forma que para girar cualquier articulación un determinado ángulo bastará aplicar a sus bobinados un tren de impulsos con un número muy concreto de ellos que será determinado por un circuito "contador".

Estos motores resultan muy baratos, su coste se incrementa al reducirse el ángulo fijo o paso que lo caracteriza ya que de éste depende la precisión.

Todos los accionadores eléctricos son por lo general más limpios que el resto y tienen un mantenimiento más económico.

De todas formas, la disponibilidad de suministro de energía en cualquier punto, así como el desarrollo de las nuevas técnicas de diseño de motores eléctricos, están predominando en toda la gama de actuadores de la robótica.

Una gran parte de los futuros desarrollos, incorporan como actuadores motores de corriente alterna, ya que las dificultades de control, hoy en día, se han superado y la robustez y el bajo coste de mantenimiento los hace insustituibles.

El grupo accionador eléctrico consta fundamentalmente de cuatro partes, la reductora, el motor, el freno y los elementos captadores para su regulación.

# Los captadores y sensores

Se tiende a llamar captadores a los elementos que captan la información que permite el control del robot, sin tener en cuenta los parámetros del entorno que le rodea.

Los sensores proporcionan al robot información sobre el entorno en que se encuentra.

## Captadores dinámicos

Existe una gran variedad de ellos, pero tienen una finalidad común, la de medir una o varias magnitudes dinámicas.

Forman parte del grupo accionador de cada articulación y se clasifican en:

### Captadores de desplazamiento o posición

Los podemos dividir en dos grandes grupos, de movimiento de traslación o de giro.

a) Los de traslación pueden ser de:

- Potenciómetro lineal
- Bobina con núcleo variable

b) Los de giro los subdividimos en otros dos grupos:

1. Digitales:

- Ópticos (Dedodificadores (Encoders) - Detectores de paso)
- Magnéticos

2. Analógicos:

- Potenciómetros (Mono o multivuelta)
- Resolvers

### Captadores de velocidad

Los más utilizados son los generadores tacométricos.

También se utiliza la técnica de medida de ranuras de encoder por unidad de tiempo.

## **Captadores de aceleración**

Utilizando el computador como parte integrante del sistema de control, se determina la aceleración insertando las señales provenientes de los captadores de velocidad, mediante los circuitos de acoplamiento adecuados y realizando la operación matemática de derivar por uno de los muchos procedimientos existentes.

Si no se dispone de computador será suficiente con dar entrada a estas señales a circuitos diferenciadores muy sencillos de construir con amplificadores operacionales.

Otros acelerómetros comerciales están basados en dispositivos centrífugos.

El empleo de circuitos codificadores y decodificadores suele ser práctica habitual en muchos de estos captadores.

## **Captadores cinemáticos**

Se suelen utilizar para medir esfuerzos y más concretamente presiones, de esta forma se puede llegar a determinar y controlar las presiones de apriete de las garras y manos del robot.

## **Captadores de esfuerzo**

Los que últimamente se están imponiendo son los de cristal de cuarzo, basados en fenómenos piezoeléctricos, se aprovecha el voltaje entre las caras opuestas al cristal y que es proporcional a la presión a que se somete al cuarzo.

Ya están cayendo en desuso las galgas extensiométricas que miden el incremento de resistencia eléctrica de un conductor debido a la deformación que provoca el esfuerzo en una determinada dirección.

## **Sensores de entorno**

Los robots dotados con este tipo de sensores, conocen algunas de las variables del mundo que les rodea y de esta forma pueden realizar algunas tareas específicas que los robots provistos únicamente de captadores no podrían realizar.

## **Sensores de proximidad**

Informan al robot de la presencia o ausencia de objetos en el campo de acción a estudiar, están basados en :

- a) Ultrasonidos
- b) Radiación luminosa

## **Sensores táctiles**

Se utilizan en la pinza del robot para controlar y medir la presión de apriete.

Son microinterruptores que cierran o abren circuitos dependiendo de la presión del apriete, colocando varios juntos se detecta el intervalo de la presión existente, debido a las características diferentes y escalonadas del resorte de cada uno de ellos.

## **Sensores de visión**

Las cámaras de visión se utilizan en algunos campos de la robótica para facilitar el:

- Reconocimiento de objetos
- Selección de objetos
- Toma de objetos
- Ensamblado
- Trayectorias opcionales salvando objetos

No hay grandes diferencias con la cámara tradicional de TV, por lo que la zona del campo de acción debe ser iluminada con niveles de intensidad elevados. De esta forma evitamos en cierto modo las interacciones debidas a las variaciones de la intensidad de luminosidad ambiental.



## El control del robot

La posición y orientación de la pinza de un robot, como la velocidad, aceleración, etc., pueden ser controladas de muchas formas y por diferentes equipos, los métodos actuales de la teoría moderna control se centran en el empleo de:

### Los servosistemas

El diagrama de bloques de la figura 7 muestra el Sistema de Control de una articulación, se aprecia una variable de entrada como referencia y la misma variable en la salida, que es la que se pretende controlar.

Cada uno de los bloques está identificado por su correspondiente ecuación de transferencia, expresión matemática de cada una de las partes de que se compone el robot.

Se pretende que la variable de salida siga a la de entrada de una forma precisa, rápida y estable dentro de un margen amplio de posibilidades dinámicas y cinemáticas.

### El control de los servosistemas

En las tareas que realice el robot y se aprecien errores de precisión o de cualquier otro tipo, nos vemos obligados a corregirlos por medio de sistemas de control o servosistemas; nos vamos a limitar al estudio de los servosistemas con motores de C.C. y C. A.

#### Con motores eléctricos de C. C.

Se suele utilizar casi exclusivamente el control por inducido, debido a la linealidad que existe y al amplio margen de regulación que es posible conseguir.

#### Con motores eléctricos de C. A.

Hace unos años era impensable controlar por software la frecuencia de la C.A. de alimentación, hoy los equipos variadores de frecuencia permiten un amplio margen de velocidades de sincronismo facilitando notablemente su regulación.

## El control cinemático

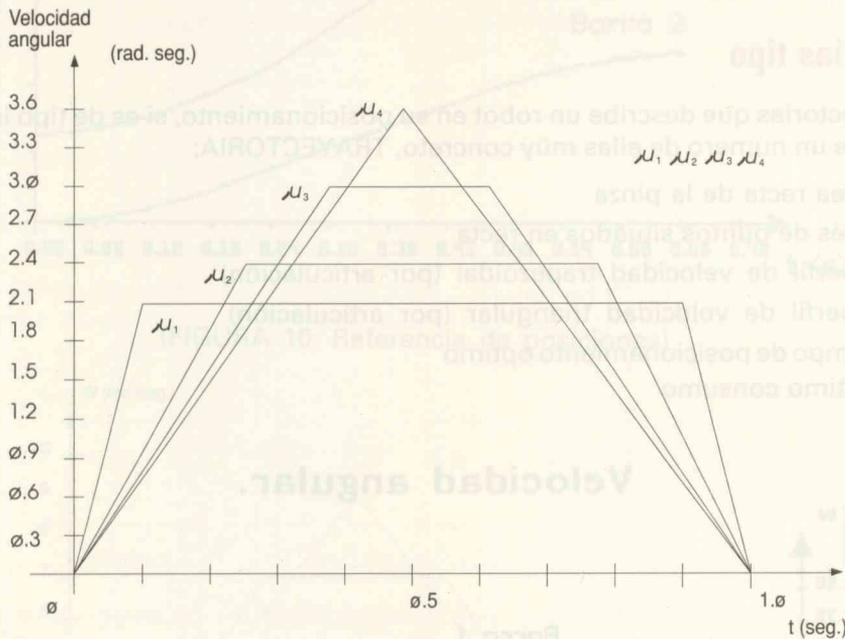
Este tipo de control se caracteriza porque no contempla los esfuerzos ni las pequeñas deformaciones que se producen, se trata de servosistemas de posición, velocidad y aceleración.

## El control dinámico

Además de controlar las magnitudes cinemáticas, considera el control de los pares motores en los ejes, así como presiones de apriete, etc., manteniendo siempre la estabilidad del sistema.

## Trayectorias de posicionamiento

El control del camino por donde pasan las diferentes partes de la estructura debe ser muy preciso, para evitar colisiones en el espacio con otros objetos, para optimizar el tiempo de posicionamiento sin sobrepasar las especificaciones impuestas por el fabricante, para mover la pinza en trayectoria recta, etc.



(FIGURA 8: Perfiles teóricos de velocidad de una articulación en relación con el perfil de velocidad triangular)

## Modo de aprendizaje del robot

Es un método muy utilizado en tareas repetitivas, consiste en controlar al robot desde el puesto de trabajo mediante una unidad programadora portátil, de esta forma las diferentes articulaciones pasan por determinados puntos que son grabados en soportes con sus características cinemáticas, para posteriormente y de una forma automática repetir la trayectoria cuantas veces se considere oportuno.

## Método de aprendizaje manual

En este método el operario conduce manualmente al robot por determinados puntos de la trayectoria, por los que posteriormente, de forma automática, pasará el robot una vez que aquellos han sido grabados.

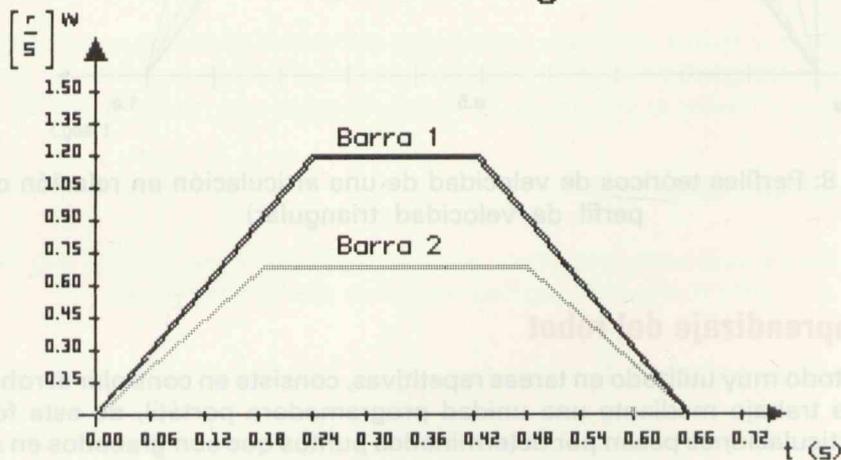
En estos puntos de paso se debe tener en cuenta el criterio de continuidad, tanto en posición como en velocidad y si fuera posible también en aceleración, pues de lo contrario aparecerán vibraciones o esfuerzos bruscos en las diferentes partes del robot.

## Trayectorias tipo

Las trayectorias que describe un robot en su posicionamiento, si es de tipo industrial, se reducen a un número de ellas muy concreto, TRAYECTORIA:

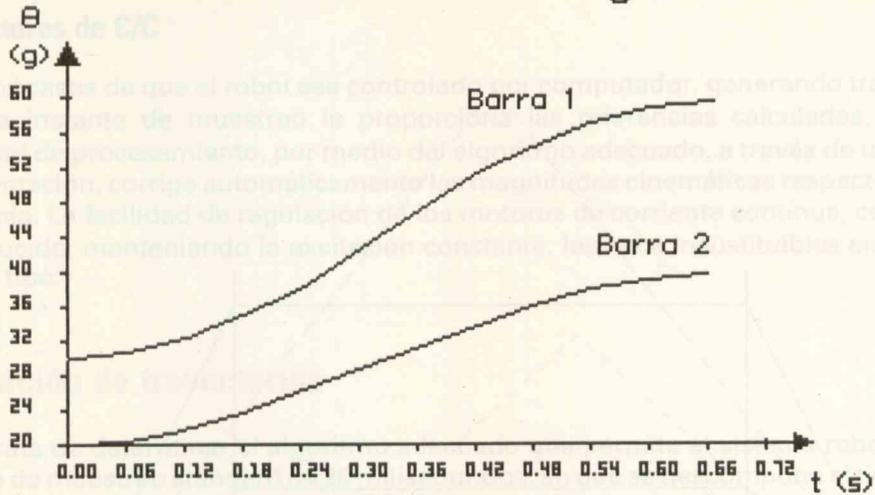
- en línea recta de la pinza
- a través de puntos situados en recta
- con perfil de velocidad trapezoidal (por articulación)
- con perfil de velocidad triangular (por articulación)
- de tiempo de posicionamiento óptimo
- de óptimo consumo

### Velocidad angular.

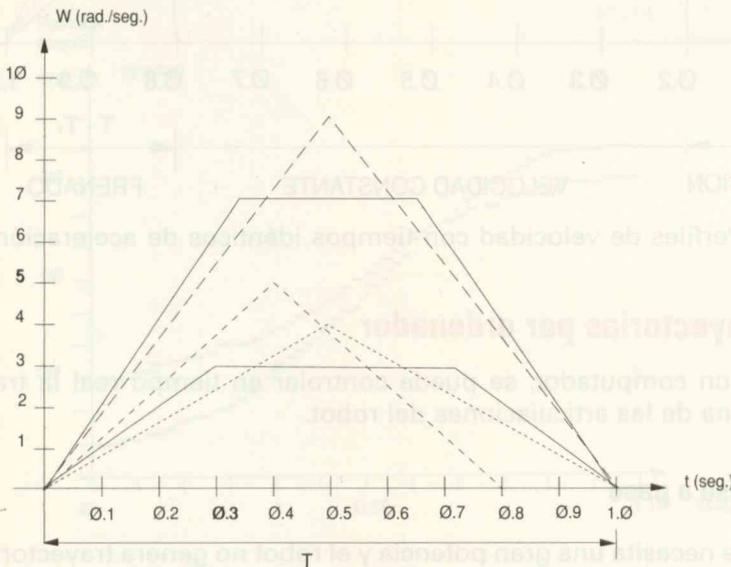


(FIGURA 9: Referencia de velocidades angulares)

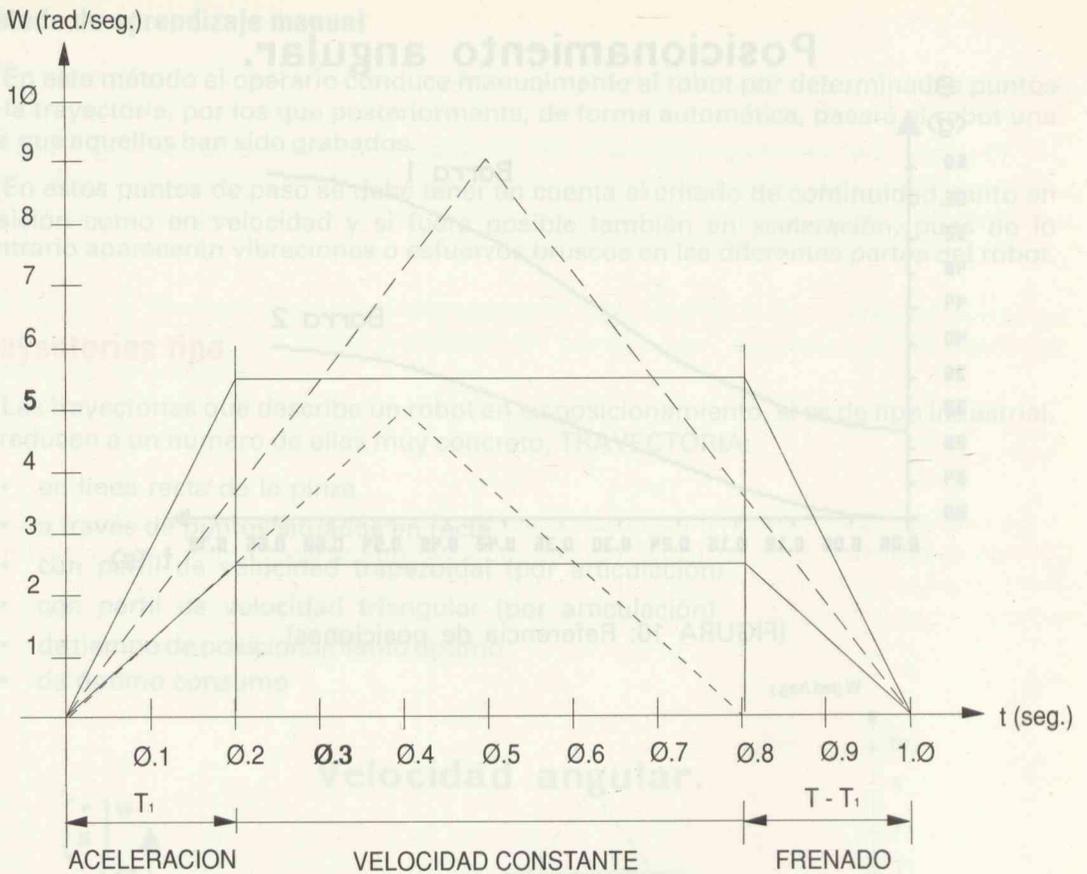
## Posicionamiento angular.



(FIGURA 10: Referencia de posiciones)



(FIGURA 11: Perfiles de velocidad con aceleración SUBOPTIMA de una articulación)



(FIGURA 12: Perfis de velocidade com tempos idénticos de aceleração e frenado)

## Control de trayectorias por ordenador

A través de un computador, se puede controlar en tiempo real la trayectoria que describe cada una de las articulaciones del robot.

### Con motores paso a paso

Cuando no se necesita una gran potencia y el robot no genera trayectorias en tiempo real, el servosistema utilizado es de lazo abierto y el computador se limita a suministrar un tren de impulsos de número muy concreto, por medio del control de un contador.

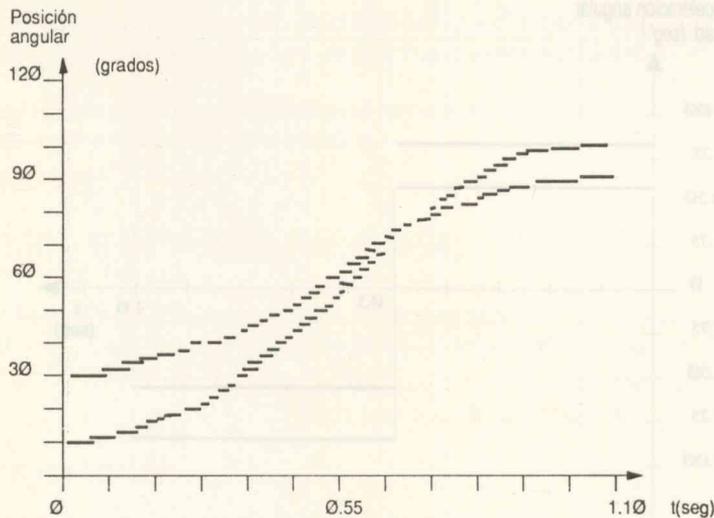
Los servosistemas con este tipo de accionadores carecen de bucle de realimentación.

## Con motores de C/C

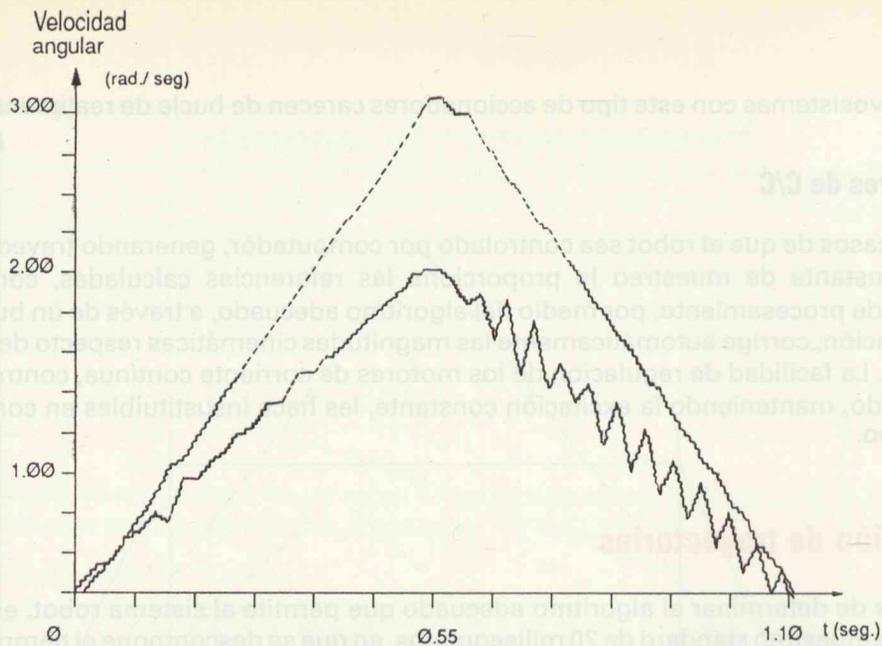
En los casos de que el robot sea controlado por computador, generando trayectorias, en cada instante de muestreo le proporciona las referencias calculadas, con gran velocidad de procesamiento, por medio del algoritmo adecuado, a través de un bucle de realimentación, corrige automáticamente las magnitudes cinemáticas respecto de las de referencia. La facilidad de regulación de los motores de corriente continua, controlados por inducido, manteniendo la excitación constante, les hace insustituibles en controles de este tipo.

## Generación de trayectorias

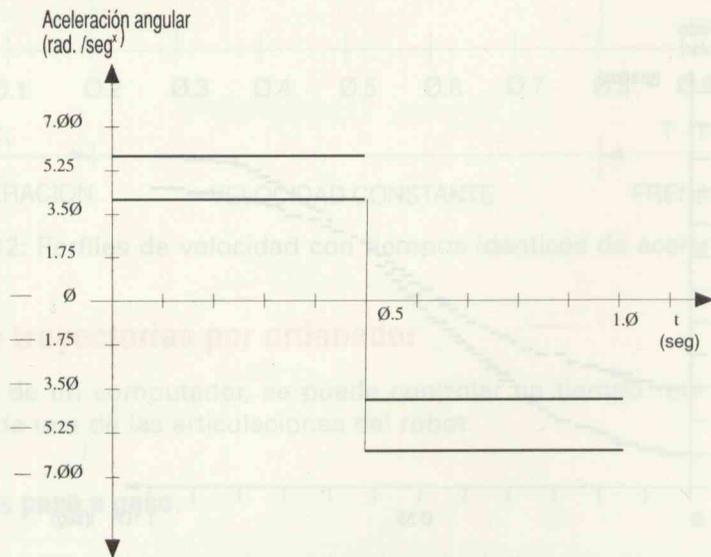
Se trata de determinar el algoritmo adecuado que permite al sistema robot, en cada periodo de muestreo standard de 20 milisegundos, en que se descompone el tiempo total de posicionamiento, pasarle las magnitudes cinemáticas de referencia y de esta forma guiar al robot desde que inicia el movimiento hasta que se posiciona.



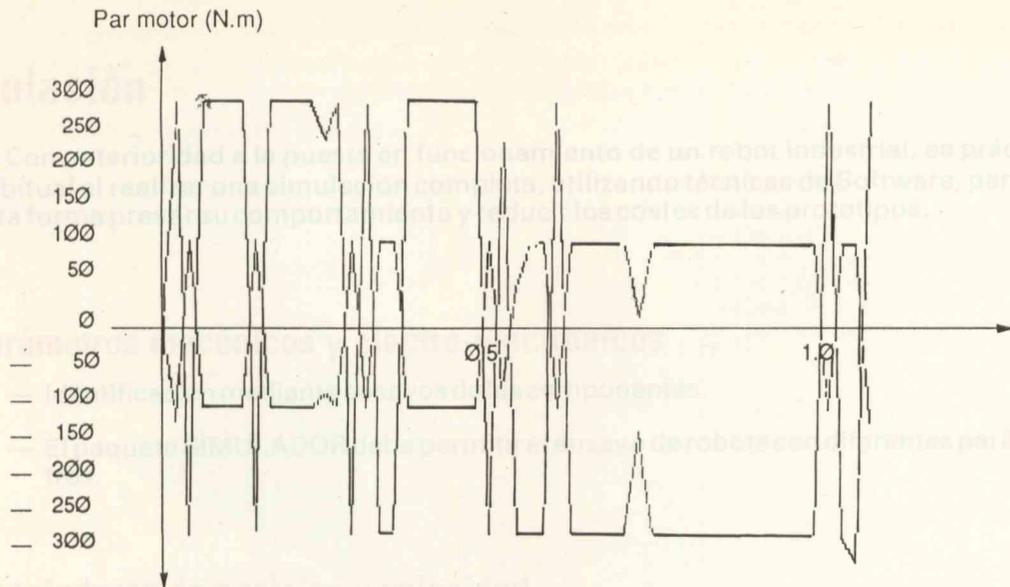
(FIGURA 13: Evolución del posicionamiento angular de las dos articulaciones, por trayectoria de velocidad del tipo TRIANGULAR)



(FIGURA 14: Perfíles de velocidad de las dos articulaciones, por trayectoria TRIANGULAR)



(FIGURA 15: Perfíles de aceleración de las dos articulaciones, por trayectoria de velocidad de tipo TRIANGULAR)



(FIGURA 16: Evolución del PAR MOTOR de las dos articulaciones, con limitadores)

- Proporcionar la robustez de control suficiente para que en régimen dinámico permita el seguimiento de las variables de referencia sin errores excesivos.
- Facilitar al usuario la posibilidad de modificar los parámetros de los reguladores y observar la respuesta del sistema.

### Puntos inicial y final del desplazamiento

El usuario los administra en coordenadas angulares o cartesianas, dentro del campo de acción del robot, que se suele utilizar para evitar movimientos imposibles.

### Tipos de trayectorias

El usuario opta por una de las trayectorias tipo, con el fin de analizar el comportamiento del robot, observando los errores dinámicos en el posicionamiento, así como la evolución de la intensidad en cada uno de los motores, el par motor, el consumo energético y cuantas otras variables se consideren de interés.



## Simulación

Con anterioridad a la puesta en funcionamiento de un robot industrial, es práctica habitual el realizar una simulación completa, utilizando técnicas de Software, para de esta forma prever su comportamiento y reducir los costes de los prototipos.

### Parámetros mecánicos y electro-electrónicos

- Identificación mediante ensayos de los componentes.
- El paquete SIMULADOR debe permitir el ensayo de robots con diferentes parámetros.

### Reguladores de posición y velocidad

- Diseñar los reguladores que se ajusten a las especificaciones de funcionamiento.
- Proporcionar la robustez de control suficiente para que en régimen dinámico permita el seguimiento de las variables de referencia sin errores excesivos.
- Facilitar al usuario la posibilidad de modificar los parámetros de los reguladores y observar la respuesta del sistema.

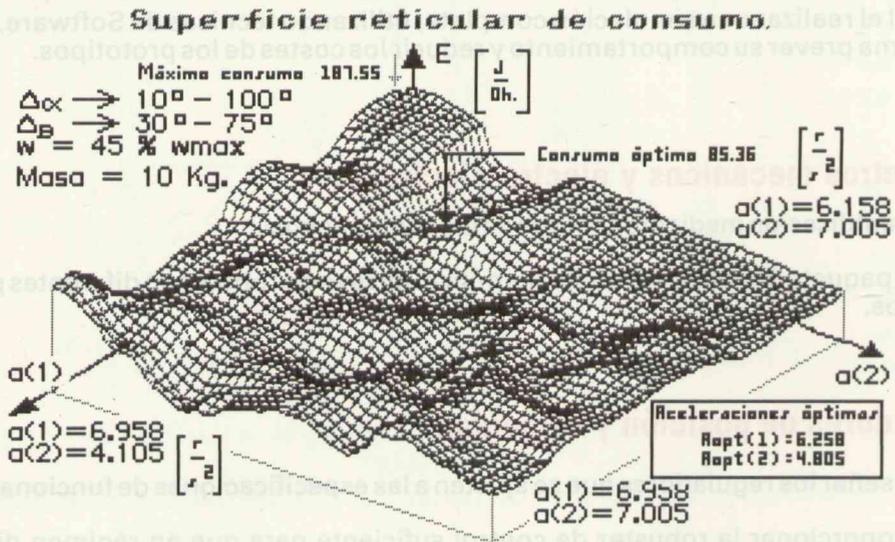
### Puntos inicial y final del desplazamiento

El usuario los suministra en coordenadas angulares o cartesianas, dentro del campo de acción del robot, que se suele acotar para evitar movimientos imposibles.

### Tipos de trayectorias

El usuario opta por una de las trayectorias tipo, con el fin de analizar el comportamiento del robot, observando los errores cinemáticos en el posicionamiento, así como la evolución de la intensidad en cada uno de los motores, el par motor, el consumo energético y cuantas otras variables se consideren de interés.

Si se trata de observar como varía el consumo en un robot de ensamblado que trabaja en el plano horizontal, por trayectorias diferentes para el mismo tiempo de posicionamiento, se obtiene la gráfica representada en la figura 17.



(FIGURA 17: Superficie reticular para optimización de consumo de un Robot con dos articulaciones)

## Tiempo de simulación

A veces no es necesario observar la evolución del robot durante todo el tiempo de posicionamiento, sino que es suficiente con el periodo inicial, de esta forma el usuario evitará tiempos muertos de espera al ejecutar la simulación.

## Pruebas de simulación

Son muchas y variadas las pruebas de simulación, a continuación se muestran varias tablas en las que se recoge el tiempo a velocidad constante en un perfil de velocidad trapecoidal con aceleración máxima especificada por el fabricante.

**TABLA N.º 1**

Resultados:

<b>Tiempo de posicionamiento</b>	<b>Tiempo a velocidad constante</b>
Segundos	Segundos
0,960	0,111
0,970	0,178
0,980	0,226
0,990	0,266
1,000	0,301
1,010	0,333

(FIGURA 18: Tabla que relaciona los Tiempos de posicionamiento de una articulación con los de Velocidad constante)

**TABLA N.º 2**

Resultados:

<b>Tiempo de posicionamiento</b>	<b>Aceleración óptima</b>
Segundos	rad seg.**2
0,06	327,2500
0,07	240,4287
0,08	184,0784
0,10	117,8104
0,20	29,4531
0,50	4,7130
1,00	1,1788
10,00	0,0125
100,00	0,0001

(FIGURA 19: Tabla que relaciona los Tiempos de posicionamiento de una articulación, con su aceleración óptima correspondiente)

**TABLA N.º 3**

Resultados:	
Relación B/J	Aceleración óptima
1 rad. seg.	rad seg.**2
0,001	4,71242
0,010	4,71268
0,100	4,71530
1,000	4,74201
10,000	5,04737
12,000	5,12185
13,000	5,67683
13,500	5,54347
13,700	5,49401
13,730	5,48698
13,735	5,48582
13,736	-1,31231
13,740	-1,31258
14,000	-1,33008
16,000	-1,46077
17,990	-1,58467
17,999	-1,58527
18,001	4,87270
20,000	4,72724
21,460	2,14295
21,470	-0,51275
23,000	-0,53825
25,000	-0,57061
30,000	1,86690
50,000	1,71076
100,000	4,20556
1000,000	4,18894
9999,000	4,18879

(FIGURA 20: Tabla que relaciona dos parámetros de la estructura mecánica, con la aceleración óptima correspondiente)

Asímismo se recoge la variación de la aceleración óptima de consumo en función del tiempo de posicionamiento y de la relación B/J (Rozamiento viscoso/Momento de inercia) de una articulación.

## Un robot didáctico



El robot CS-113 es un robot tipo para aplicaciones didácticas en el aula, permite su control desde cualquier ordenador personal ya sea por medio de instrucciones sencillas en lenguajes de alto nivel como en lenguaje ensamblador.

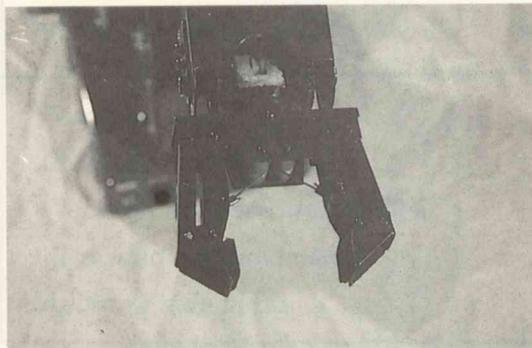
Se puede controlar por medio de instrucciones en "GW BASIC" a través de la salida paralelo CENTRONICS habilitada generalmente para la utilización de la impresora, si

bien admite el control por medio del protocolo RS-232 C.

### Descripción mecánica

Se trata de un robot del tipo de articulación coordinada, denominado así por la semejanza con los movimientos de las articulaciones del cuerpo humano.

Puede moverse articuladamente con seis ejes; el movimiento del primer eje se corresponde con el de la cintura, el segundo



es homólogo al movimiento vertical del hombro. El tercer eje se corresponde al giro del codo, también en sentido vertical. La combinación de los movimientos del cuarto y quinto eje producen los movimientos de giro e inclinación de la muñeca y el sexto eje efectúa los movimientos de abrir y cerrar la pinza.

## Especificaciones eléctricas

Los elementos principales del circuito de control son: el microprocesador Z-80 A como unidad principal de control, la memoria ROM contiene el programa del sistema y de las aplicaciones de test, la memoria RAM que contiene la zona de datos del sistema y por último los circuitos de accionamiento de los motores.

A continuación se recogen en forma de tablas las características fundamentales de alimentación y movimientos, así como de las características generales:

### CS 113 ALIMENTACION ELECTRICA Y CONTROL

<i>Motores</i>	6 Motores paso a paso
<i>Control de velocidad</i>	Control por onda cuadrada
<i>Interface</i>	Centronics
<i>Lenguajes de Programación</i>	Basic y Ensamblador
<i>Alimentación</i>	AC 110/220 v. 50/60 Hz.
<i>Accesorios</i>	Cable de conexión Fuente de alimentación

## CS 113 ESPECIFICACIONES DEL MOVIMIENTO DE LOS MOTORES

<i>Motor</i>	<i>Eje</i>	<i>Grados/Paso</i>	<i>Desplazamiento máximo desde la posición cero</i>
M1	Cintura	0,12°	120°
M2	Hombro	0,12°	70°
M3	Codo	0,08°	50°
M4	Muñeca	0,05°	90°
M5			
M6	Mano	0,1°	360°

**NOTA:** Si la entrada correspondiente a M4 y M5 es p. ej.: (90,-90) la muñeca se moverá hacia abajo 90°, si la entrada fuera (-90,-90) giraría 90° en el sentido positivo (según las agujas del reloj). Si M4>M5 o M4<M5, se producen simultáneamente los movimientos de inclinación y giro.

## CARACTERISTICAS GENERALES DEL CS 113

<i>Construcción</i>	Estructura de metal soldado
<i>Capacidad de carga</i>	500 g. (max.)
<i>Peso del robot</i>	8 kg. (aprox.)
<i>Grados de libertad</i>	5
<i>Precisión del posicionamiento</i>	0,9 mm.
<i>Velocidad del movimiento</i>	300 mm./seg. (max.)
<i>Apertura de la pinza</i>	45 mm.







**INTRODUCCION AL DISEÑO ELECTRONICO**





---

**Ministerio de Educación y Ciencia**

---

Secretaría de Estado de Educación

---

*Programa de Nuevas Tecnologías de la Información y Comunicación*

---

