

ESCUELA POLITECNICA NACIONAL

FACULTAD DE INGENIERIA ELECTRICA

ESPECIALIZACION DE ELECTRONICA

Y

TELECOMUNICACIONES

DISEÑO DE CIRCUITOS INTEGRADOS DE APLICACION  
ESPECIFICA (ASICs) DIGITALES CON TECNOLOGIA CMOS

VOLUMEN III

CODIFICADOR/DECODIFICADOR DE LINEA PROGRAMABLE  
HDB<sub>n</sub> COMO CASO DE ESTUDIO

IVAN BERNAL CARRILLO  
FREDY LEMUS CRIOLLO

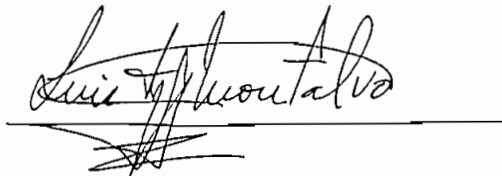
TESIS PREVIA A LA OBTENCION DEL TITULO DE  
INGENIERO EN LA ESPECIALIZACION DE INGENIERIA  
ELECTRONICA DE LA ESCUELA POLITECNICA NACIONAL

SEPTIEMBRE 1992

*Certifico que bajo mi dirección  
la presente tesis fue realizada  
en su totalidad por los  
señores:*

*Iván Bernal Carrillo*

*Fredy Lemus Criollo*

A handwritten signature in black ink, reading "Luis Montalvo Ramirez", written over a horizontal line. The signature is stylized and cursive.

*Ing. Luis Montalvo Ramirez*

*DIRECTOR*

## CONTENIDO GENERAL

### VOLUMEN I. FUNDAMENTOS TEORICOS DE DISEÑO VLSI (VERY LARGE SCALE INTEGRATION)

#### CAPITULO 1. GENERALIDADES DE LA METODOLOGIA DE DISEÑO VLSI.

- 1.1 Introducción.
- 1.2 Niveles de abstracción.
- 1.3 Estrategias para la reducción de la complejidad del diseño de un CI.
- 1.4 Alternativas de diseño de sistemas electrónicos.
- 1.5 Tecnologías de fabricación.
- 1.6 Aspectos económicos del desarrollo y producción de CIs.

#### CAPITULO 2. FUNDAMENTOS DE CIRCUITOS CMOS.

- 2.1 Generalidades.
- 2.2 El Transistor MOS como switch.
- 2.3 Funcionamiento y física del Transistor MOS.
- 2.4 El Inversor CMOS - Características DC.
- 2.5 La Compuerta de Transmisión - Características DC.
- 2.6 El problema del efecto parásito de LATCH-UP.
- 2.7 Comparación con otras tecnologías.

### CAPITULO 3. PROCESOS DE FABRICACION CMOS.

- 3.1 Técnicas básicas de fabricación.
- 3.2 Tecnologías de fabricación CMOS.
- 3.3 Interfaz Diseño-Fabricación.
- 3.4 Estructuras de Entrada/Salida: PADs.
- 3.5 Tipos de empaquetamiento.
- 3.6 Chips Multi-proyecto.

### CAPITULO 4. CARACTERIZACION Y ESTIMACION DEL FUNCIONAMIENTO DE CIRCUITOS INTEGRADOS.

- 4.1 Parámetros eléctricos: estimación de Resistencia y Capacitancia.
- 4.2 Características de conmutación y retardos.
- 4.3 Estimación de consumo de potencia.
- 4.4 Escalamiento de dimensiones del transistor MOS.
- 4.5 Estrategias de Temporización de los sistemas.

## VOLUMEN II. HERRAMIENTAS PARA DISEÑO DE ASICs

### CAPITULO 5. HERRAMIENTAS DE DISEÑO VLSI.

- 5.1 Tipos de herramientas de diseño.
- 5.2 Secuencias de diseño para la concepción de ASICs.
- 5.3 Principios de simulación eléctrica y lógica.
- 5.4 Formato CIF para descripción geométrica de layouts.

CAPITULO 6. DISEÑO DE ASICs BASADO EN CELDAS ESTANDAR.

- METODOLOGIA CONVENCIONAL.

6.1 Diseño basado en bandas y celdas estándar.

6.2 Herramientas del sistema "TENTOS" para diseño convencional.

CAPITULO 7. CASO DE ESTUDIO SENCILLO UTILIZANDO TENTOS:

DISEÑO DE UN "MEDIO SUMADOR".

7.1 Diseño a nivel funcional.

7.2 Diseño a nivel estructural.

7.3 Diseño a nivel físico.

7.4 Resultados del funcionamiento de los prototipos fabricados en la fundidora ES2(Francia).

CAPITULO 8. DISEÑO DE ASICs BASADO EN CELDAS ESTANDAR.

- METODOLOGIA ESTRUCTURADA PPL (*Physical Placement of Logic*).

8.1 Diseño convencional versus diseño PPL.

8.2 Metodología de diseño PPL.

8.3 Manipulación de las herramientas PPL. Caso de estudio sencillo: diseño de una compuerta "EXOR".

VOLUMEN III. CODIFICADOR/DECODIFICADOR  
DE LINEA PROGRAMABLE  
HDB<sub>n</sub> COMO CASO DE ESTUDIO.

CAPITULO 9. CONCEPCION DEL CIRCUITO CODIFICADOR/DECODIFI-  
CADOR.

- 9.1 Consideraciones teóricas del Código HDB<sub>n</sub>.
- 9.2 Diseño a nivel funcional del Codificador en base a Máquinas de Estados Finitos (MEF).
- 9.3 Diseño a nivel funcional del Decodificador en base a Máquinas de Estados Finitos (MEF).

CAPITULO 10. DISEÑO DEL CODIFICADOR/DECODIFICADOR HDB<sub>n</sub>  
BASADO EN CELDAS ESTANDAR - METODO CONVEN-  
CIONAL.

- 10.1 Diseño del Codificador Programable HDB<sub>n</sub>.
- 10.2 Diseño del Decodificador Programable HDB<sub>n</sub>.
- 10.3 Diseño de anillo de PADs e inserción del diseño global.

CAPITULO 11. DISEÑO DEL CODIFICADOR/DECODIFICADOR HDB<sub>n</sub>  
BASADO EN CELDAS ESTANDAR - METODO ESTRUCTURA-  
DO PPL.

- 11.1 Diseño del Codificador Programable HDB<sub>n</sub>.
- 11.2 Diseño del Decodificador Programable HDB<sub>n</sub>.
- 11.3 Inserción del diseño global en el PAD-FRAME.

## CAPITULO 12. CONCLUSIONES Y RECOMENDACIONES.

12.1 Conclusiones generales.

12.1 Conclusiones respecto al paquete TENTOS.

12.2 Conclusiones respecto al paquete PPL.

## BIBLIOGRAFIA

### ANEXOS

ANEXO A. DISPOSITIVOS Y COMANDOS SPICE EMPLEADOS EN SIMULACION DIGITAL.

ANEXO B. DISPOSITIVOS Y COMANDOS EMPLEADOS EN SIMULACION LOGICA NDL.

ANEXO C. DIRECTIVAS DE DESCRIPCION CIF.

ANEXO D. DETALLES DEL SISTEMA TENTOS.

ANEXO E. ARCHIVOS AUXILIARES DEL TENTOS.

ANEXO F. CELDAS ESTANDAR EMPLEADAS - METODO CONVENCIONAL.

ANEXO G. REGLAS DE DISEÑO .

ANEXO H. PLANOS DEL CIRCUITO CODIFICADOR/DECODIFICADOR HDBn.

## INDICE DEL VOLUMEN III

CAPITULO 9. CONCEPCION DEL CIRCUITO CODIFICADOR DECODIFICADOR	
9.1	CONSIDERACIONES TEORICAS DEL CODIGO HDBn . . . . . 1
9.1.1	Codificación de línea. . . . . 1
9.1.2	Transmisión de señales de datos binarias (encendido-apagado) . . . . . 2
9.1.3	Códigos de línea prácticos. . . . . 4
9.2	DISEÑO A NIVEL FUNCIONAL DEL CODIFICADOR EN BASE A MAQUINAS DE ESTADOS FINITOS (MEF) . . . . . 8
9.2.1	Establecimiento de bloques funcionales . . . . . 8
9.2.2	Diseño de la MEF(1) . . . . . 14
9.2.3	Diseño de la MEF(2) . . . . . 22
9.2.4	Etapa de salida . . . . . 32
9.2.5	Estructura del Codificador HDBn . . . . . 33
9.3	DISEÑO A NIVEL FUNCIONAL DEL DECODIFICADOR EN BASE A MAQUINAS DE ESTADOS FINITOS (MEF) . . . . . 36
9.3.1	Establecimiento de bloques funcionales . . . . . 36
9.3.2	Diseño de la MEF(1) . . . . . 42
9.3.3	Diseño de la MEF(2) . . . . . 51
9.3.4	Etapa de salida . . . . . 67
9.3.5	Estructura del Decodificador HDBn . . . . . 68



CAPITULO 10.	DISEÑO DEL CODIFICADOR/DECODIFICADOR HDBn	
	BASADO EN CELDAS ESTANDAR - METODO	
	CONVENCIONAL . . . . .	70
10.1	DISEÑO DEL CODIFICADOR PROGRAMABLE HDBn . . . . .	71
10.1.1	Diseño a nivel funcional . . . . .	71
10.1.2	Diseño a nivel lógico . . . . .	72
10.1.3	Diseño a nivel fisico . . . . .	85
10.2	DISEÑO DEL DECODIFICADOR PROGRAMABLE HDBn . . . . .	115
10.2.1	Diseño a nivel funcional . . . . .	115
10.2.2	Diseño a nivel lógico . . . . .	116
10.2.3	Diseño a nivel físico . . . . .	129
10.3	DISEÑO DEL ANILLO DE PADS E INSERCIÓN DEL DISEÑO	
	GLOBAL. . . . .	155
10.3.1	Plano de base del núcleo . . . . .	155
10.3.2	Selección e interconexión de PADS . . . . .	156
10.3.3	Plano de base global del diseño . . . . .	157
10.3.4	Layout global del diseño . . . . .	159
CAPITULO 11.	DISEÑO DEL CODIFICADOR/DECODIFICADOR HDBn	
	BASADO EN CELDAS ESTANDAR - METODO ESTRUCTURA-	
	DO PPL. . . . .	161
11.1	DISEÑO DEL CODIFICADOR PROGRAMABLE HDBn . . . . .	162
11.1.1	MEF(1) . . . . .	162
11.1.2	MEF(2) . . . . .	181
11.1.3	Multiplexores . . . . .	191
11.1.4	Integración de los módulos del	
	codificador . . . . .	198

11.2	DISEÑO DEL DECODIFICADOR PROGRAMABLE HDBn . . . . .	207
11.2.1	MEF(1) . . . . .	207
11.2.2	MEF(2) . . . . .	217
11.2.3	Integración de los módulos del decodificador . . . . .	225
11.3	INSERCIÓN DEL DISEÑO GLOBAL EN EL <i>PAD-FRAME</i> . . . . .	235
 <b>CAPITULO 12. CONCLUSIONES Y RECOMENDACIONES . . . . .</b>		<b>240</b>
12.1	CONCLUSIONES GENERALES . . . . .	240
12.2	CONCLUSIONES RESPECTO AL PAQUETE TENTOS . . . . .	243
12.3	CONCLUSIONES RESPECTO AL PAQUETE PPL . . . . .	248
12.4	RECOMENDACIONES . . . . .	253
 <b>BIBLIOGRAFIA . . . . .</b>		<b>256</b>
 <b>ANEXOS</b>		
ANEXO A	DISPOSITIVOS Y COMANDOS SPICE EMPLEADOS EN SIMULACION DIGITAL . . . . .	260
ANEXO B	DISPOSITIVOS Y COMANDOS EMPLEADOS EN SIMULACION LOGICA NDL . . . . .	280
ANEXO C	DIRECTIVAS DE DESCRIPCION CIF . . . . .	295
ANEXO D	DETALLES DEL SISTEMA TENTOS . . . . .	306
ANEXO E	ARCHIVOS AUXILIARES DEL TENTOS . . . . .	330
ANEXO F	CELDA ESTANDARD EMPLEADA - METODO CONVENCIONAL . . . . .	345
ANEXO G	REGLAS DE DISEÑO . . . . .	363
ANEXO H	P L A N O S        D E L        C I R C U I T O CODIFICADOR/DECODIFICADOR HDBn . . . . .	384

## CAPITULO 9

### CONCEPCION DEL CIRCUITO CODIFICADOR/DECODIFICADOR

#### 9.1 CONSIDERACIONES TEORICAS DEL CODIGO HDBn

##### 9.1.1 Codificación de línea.

Para la transmisión de señales binarias en banda base, es necesario primeramente adaptar la información al canal de transmisión, para lo cual se utilizan los denominados "Códigos de línea". Existen diferentes códigos de línea, cada uno con sus ventajas e inconvenientes. Entre otras propiedades deseables, un código de línea debe tener las siguientes:

- a) Debe permitir la extracción de la señal de sincronización o reloj, a partir de la señal recibida.
  
- b) Debe tener la mínima probabilidad de error; es decir, la máxima inmunidad al ruido de canal y a la interferencia intersímbolo, para un ancho de banda y una potencia de transmisión dados.

- c) Debe ser posible detectar, y de preferencia corregir errores.
- d) Debe proveer una forma adecuada del espectro de frecuencia, para el canal de transmisión. Es también deseable tener  $f=0$  (componente continua), ya que debe realizar el acoplamiento a AC en los repetidores o receptores, principalmente utilizando transformadores.
- e) Debe posibilitar la codificación y decodificación de manera única para reproducir la señal binaria original.

#### 9.1.2 Transmisión de señales de datos binarias (encendido-~~do~~-apagado)

Consiste en la transmisión directa de la información binaria, utilizando dos niveles de voltaje, un voltaje positivo ("1") y otro de cero voltios ("0"). En la Fig.9.1 se presenta la forma del espectro de potencia para el caso RZ (retorno a cero antes de concluir el tiempo de duración del voltaje positivo) de 50%. Puede notarse la presencia de una componente a la frecuencia de reloj y la concentración de la potencia a bajas frecuencias. Se debe mencionar también a para el caso NRZ (no retorno a cero), la componente a la frecuencia del reloj se anula. De la Fig. 9.1 se puede observar que el ancho de banda mínimo de la señal es  $2f_0$ .

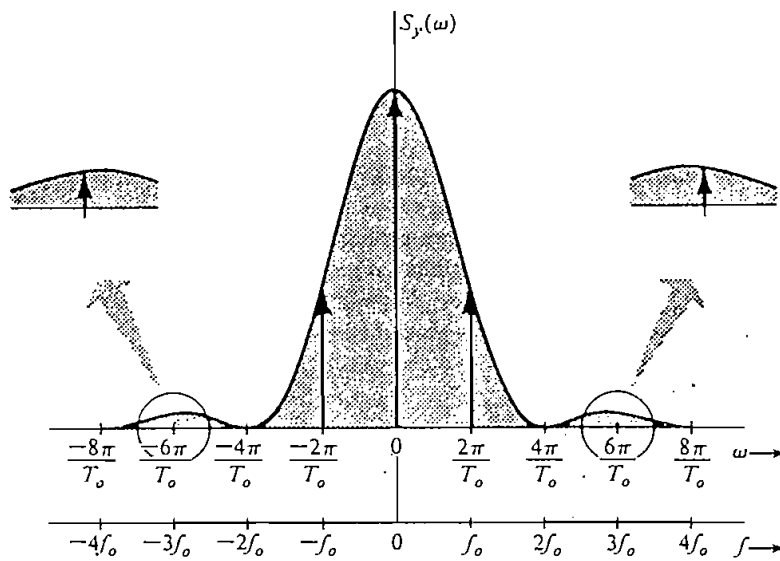


FIGURA 9.1 Densidad espectral de potencia para transmisión encendido-apagado.

El tipo de transmisión encendido-apagado es atractivo desde el punto de vista de simplicidad del circuito terminal, pero presenta algunas desventajas. En primer lugar, la densidad de potencia es diferente de cero en  $\omega=0$ , lo que produce problemas debido al acoplamiento AC. El acoplamiento AC permite a los transformadores y capacitores de bloqueo ayudar a la igualación de impedancias y eliminar la polarización. En segundo lugar, los requisitos de ancho de banda de transmisión son excesivos. En tercer lugar, no tiene la capacidad ni de detección ni corrección de errores. Por último, una sucesión larga de "0s" puede crear errores en la extracción de la temporización.

### 9.1.3 Códigos de línea prácticos.

#### a) Código polar de dos niveles

En este tipo de codificación, un "1" se transmite como un pulso positivo y un "0" se transmite mediante un pulso de igual amplitud pero negativo. El espectro obtenido en este caso también posee componente continua, como en la señal de encendido-apagado, debido a largas secuencias de "0s" o "1s" (Fig. 9.2). Este esquema tiene una probabilidad de error menor que la de encendido-apagado, por la diferencia entre las amplitudes utilizadas para la transmisión del "0" y el "1". Sin embargo, todavía presenta las demás desventajas del esquema de transmisión de encendido-apagado.

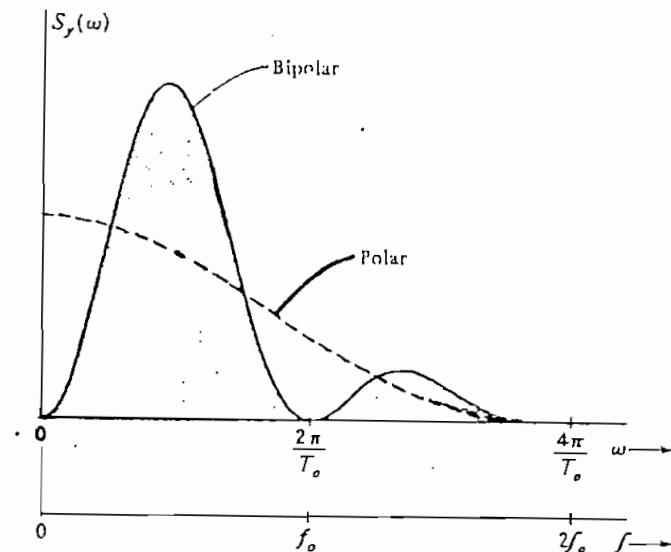


FIGURA 9.2 Densidad espectral de potencia para los códigos polar y bipolar.

## b) Código bipolar o AMI

El código que satisface la mayoría de los requerimientos mencionados anteriormente es el código bipolar o AMI (Alternate Mark Inversion). Este esquema transmite un "0" como la ausencia de pulso (0), y un 1 mediante un pulso positivo (+) o negativo (-), dependiendo si el "1" anterior se transmitió como pulso positivo o negativo. Alternando pulsos consecutivos, se puede conseguir una componente continua casi nula en la densidad espectral de potencia. Puesto que utiliza (0, +, y -), el esquema es de un código ternario. En la Fig. 9.2 se presenta el esquema de densidad espectral de potencia para este tipo de señal con RZ del 50%.

Las características de este código son:

- a) El espectro de potencia no tiene componente continua.
- b) El ancho de banda esencial requerido es  $f_b$ , es decir no es excesivo.
- c) La fácil detección de errores, observando la alternabilidad de los pulsos transmitidos.
- d) Largas secuencias de "0s" y "1s" no proporcionan la suficiente información para la recuperación de la señal de reloj.

c) Códigos bipolares de alta densidad (hdbn)

La codificación bipolar de alta densidad (High Density Bipolar) es una modificación de la codificación bipolar. El problema de recuperación de la señal de reloj queda eliminado al agregar pulsos cuando el número de "0s" consecutivos excede de "n", donde "n" puede tomar cualquier valor 1,2,3,... La idea general de estos códigos es que cuando ocurre una sucesión de más de "n" ceros, los "n+1" ceros se reemplazan por una sucesión de dígitos binarios especiales. La sucesión se elige para que incluyan algunos "1s" con la finalidad de aumentar el contenido de cruces por cero de la señal. Los "1s" que se incluyen deliberadamente violan la regla bipolar para la fácil identificación de la sucesión sustituida. A pesar de las violaciones bipolares deliberadas, los códigos HDBn retienen la capacidad de detección de errores.

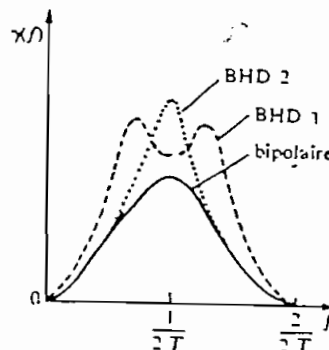


FIGURA 9.3 Densidad espectral de potencia para los códigos bipolar, HDB<sub>1</sub> y HDB<sub>2</sub>.



Los códigos bipolares de alta densidad han sido ampliamente adoptados para la transmisión de señales multiplexadas de primer y segundo orden, especialmente en Estados Unidos y Europa. De estos códigos, el más utilizado es el HDB<sub>3</sub>.

En la Fig. 9.3 se presenta la forma de los espectros para los códigos HDB<sub>1</sub>, HDB<sub>2</sub>, conjuntamente con el del código bipolar (AMI). Puede notarse que los códigos de alta densidad presentan mayores niveles que el bipolar, conforme se incrementa "n". En la Fig. 9.4 se presentan los espectros del código HDB<sub>3</sub> y el bipolar. El ancho de banda requerido en todos los casos es  $f_0$ .

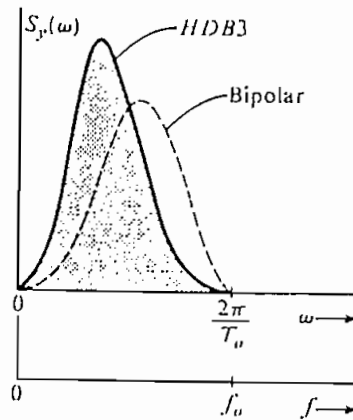


FIGURA 9.4 Densidad espectral de potencia para los códigos bipolar y HDB<sub>3</sub>

9.2 DISEÑO A NIVEL FUNCIONAL DEL CODIFICADOR EN BASE A MAQUINAS DE ESTADOS FINITOS (MEF)

9.2.1 Establecimiento de bloques funcionales

Dada una secuencia cualquiera:

$$\dots\dots\dots; \underbrace{1/0 \ 1/0 \ 1/0 \ \dots\dots 1/0 \ 1 \ \underline{0 \ \dots \ 0}}_{\substack{(n+1) \text{ Os} \\ \text{CONSECUTIVOS}}} ; \underbrace{1/0 \ 1/0 \ 1/0 \ \dots\dots 1/0 \ 1 \ \underline{0 \ \dots \ 0}}_{\substack{(n+1) \text{ Os} \\ \text{CONSECUTIVOS}}} ; \dots\dots\dots$$

Cada secuencia encerrada entre ;.....; contiene un grupo de bits terminados en (n+1)Os consecutivos a los que se aplica el principio de codificación HDBn luego de lo que una nueva secuencia se inicia. Un grupo representativo de esta secuencia tiene la estructura:

$$\underbrace{\phantom{; 1/0 \ 1/0 \ 1/0 \ \dots\dots\dots 1/0 \ 1 \ 0_1 0_2 \dots 0_n 0_{n+1} ;}}_{\substack{(I) \\ U = \# \text{ de 1s}}} \quad \underbrace{\phantom{; 1/0 \ 1/0 \ 1/0 \ \dots\dots\dots 1/0 \ 1 \ 0_1 0_2 \dots 0_n 0_{n+1} ;}}_{(II)}$$

Donde:

- (I) Es una secuencia de 1s y/o 0s terminada en "1", tiene una cantidad total "U" de 1s y un máximo de (n)Os consecutivos.
- (II) Es una secuencia de (n+1)Os consecutivos.

En adelante se empleará la siguiente nomenclatura:

B: Es un código "+" ó "-" que guarda alternabilidad con su signo "-" ó "+" próximo anterior, independientemente del número de Os que medien entre ellos.

V: Es un código "+" ó "-" que viola la alternabilidad con su signo "+" ó "-" próximo anterior, independientemente del número de Os que medien entre ellos.

Por ejemplo:

B B : significa: + - ó - +  
 B 0 0 B : significa: + 0 0 - ó - 0 0 +  
 B 0 V : significa: + 0 + ó - 0 -

El principio de codificación HDBn dice:

La secuencia:  $1/0 \ 1/0 \ 1/0 \ \dots \ 1/0 \ 1 \ 0_1 0_2 \dots 0_n 0_{n+1}$   
 (I) (II)

Se codifica como:  $B/0 \ B/0 \ B/0 \ \dots \ B/0 \ B \ B_1 0_2 \dots 0_n V_{n+1}$   
 (I) (II)  
 Si U = #par

O como:  $B/0 \ B/0 \ B/0 \ \dots \ B/0 \ B \ 0_1 0_2 \dots 0_n V_{n+1}$   
 (I) (II)  
 Si U = #impar

En otras palabras:

- a) Si U=#par de 1s en (I), se introduce un bit B en la secuencia (II) y al final de ésta un bit V.
- b) Si U=#impar de 1s en (I), únicamente se introduce un bit V en (II).

En consecuencia "el número de bits B en la secuencia completa es siempre IMPAR".

El módulo funcional que realice la codificación de la secuencia anterior, debería en una primera aproximación (Fig.9.5) codificar la secuencia (I):

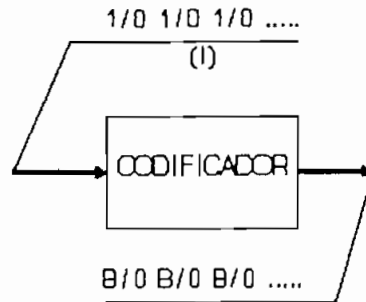


FIGURA 9.5 Módulo funcional del codificador HDBn (primera aproximación).

Sin embargo, al llegar la secuencia (II) deben considerarse dos opciones de codificación que se representan en las Figuras 9.6 (a) y (b).

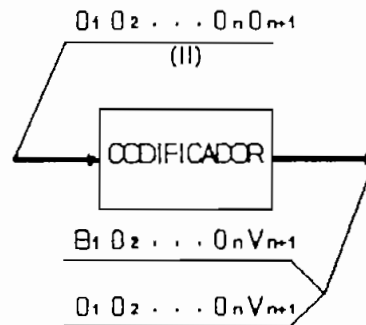


FIGURA 9.6 Módulo funcional del codificador (a), módulo simplificado (b).

Por tanto, la codificación de  $0_1$  como  $B_1$  ó  $0_1$  no puede decidirse "a priori" ya que es consecuencia de la llegada de  $(n)0$ s consecutivos "posteriores", ello obliga al módulo a:

- a) Retener una secuencia de  $(n+1)$  bits codificados a fin de tener capacidad de modificar la codificación del bit  $0_1$

de ser necesario cuando la secuencia de  $(n+1)0$ s consecutivos se presenta.

- b) Contar con una salida adicional que realice esta modificación.

Para ello al módulo de codificación de la Fig.9.6 se le ha dado la estructura esquemática de la Fig.9.7:

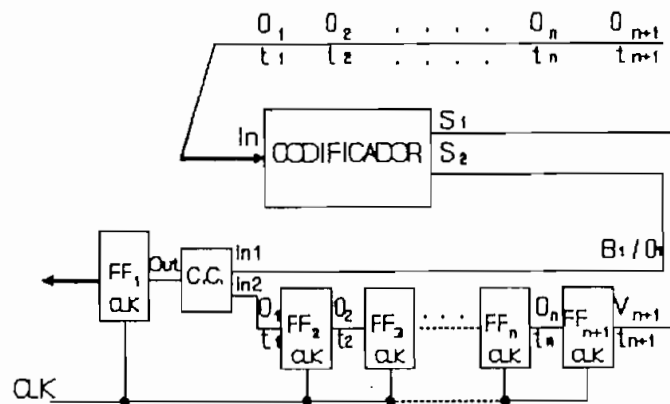


FIGURA 9.7 Esquema del codificador HDBn.

En este esquema se tiene:

- a) Los flip-flops (FFs) del codificador HDBn retienen los  $(n+1)$  últimos bits codificados y sincronizan su avance según un reloj externo.
- b) El Codificador cuenta con dos salidas:
- $S_1$ : empleada para generar los códigos resultantes de la codificación "a priori" de los bits entrantes.
  - $S_2$ : empleada para modificar, de ser necesario, al código que ingresa a FF1 cuando se ha identificado la secuencia de  $(n+1) 0$ s consecutivos, de manera que:

$S_2 = B_1$  Si es necesario cambiar a "B" la codificación del bit  $O_1$  codificado originalmente como "0" al inicio de la secuencia:  $O_1 O_2 \dots O_n O_{n+1}$ .

$S_2 = 0$  En cualquier otro caso.

c) El Circuito Combinacional (CC) trabaja de manera que:

Si  $S_2=B=In1$  Out=In1 Si  $S_2=0=In1$  Out=In2

como se indica en la Tabla 9.1

In2	In1	Out
0	0	0
B	0	B
0	B	B
B	B	B

TABLA 9.1 Funcionamiento del C.C. de la Fig.9.7.

Que corresponde a una compuerta OR.

En consecuencia, el esquema de la Fig.9.7 se modifica como indica la Fig.9.8.

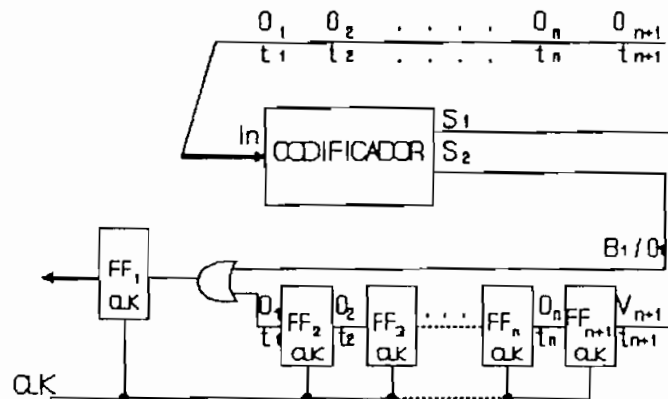


FIGURA 9.8 Esquema del codificador HDBn de la Fig.9.7 modificado.

NOTA: El número de líneas de cada bus de los esquemas planteados se deducirá posteriormente.

Por tanto, el codificador HDBn debe realizar tres tareas específicas:

- a) Monitorear las secuencias entrantes a fin de identificar secuencias de  $(n+1)$  Os consecutivos.
- b) Codificar las secuencias:  $1/0 \ 1/0 \ \dots \ 1/0 \ 1$   
como:  $B/0 \ B/0 \ \dots \ B/0 \ B$
- c) De detectarse la secuencia:  $B_1 \ 0_2 \ \dots \ 0_n \ V_{n+1}$   
codificarla como:  $0_1 \ 0_2 \ \dots \ 0_n \ V_{n+1}$   
ó como:  $0_1 \ 0_2 \ \dots \ 0_n \ V_{n+1}$   
según cada caso.

Este planteamiento permite descomponer el módulo del codificador en dos Máquinas de Estados Finitos (circuitos secuenciales) con funciones plenamente diferenciadas y mutuamente complementarias (Fig.9.9):

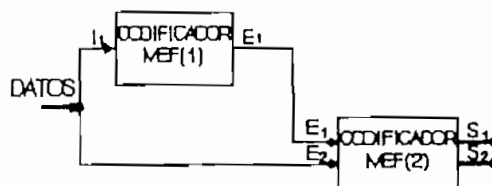


FIGURA 9.9 Codificador HDBn descompuesto en dos máquinas de estados finitos.

Donde:

- a) La Máquina de Estados Finitos MEF(1) monitorea el número de Os consecutivos en las secuencias entrantes.
- b) La Máquina de Estados Finitos MEF(2) codifica los bits  $1/0$  como  $B/0$ .
- c) La MEF(1) reporta a la MEF(2) a través de  $E_1$  las fluctuaciones en las secuencias de Os consecutivos que va monitoreando.

Concebido de esta manera el diseño, las dos máquinas funcionan en forma paralela y reducen el número de estados lógicos necesarios al mínimo, evitándose estados lógicos redundantes con miras a un ahorro de área en términos de diseño VLSI.

### 9.2.2 Diseño de la MEF(1)

#### a) Diagramas y tablas de estados

La MEF(1) es un identificador de secuencias con capacidad de identificar hasta (n)0s consecutivos (Fig.9.10).

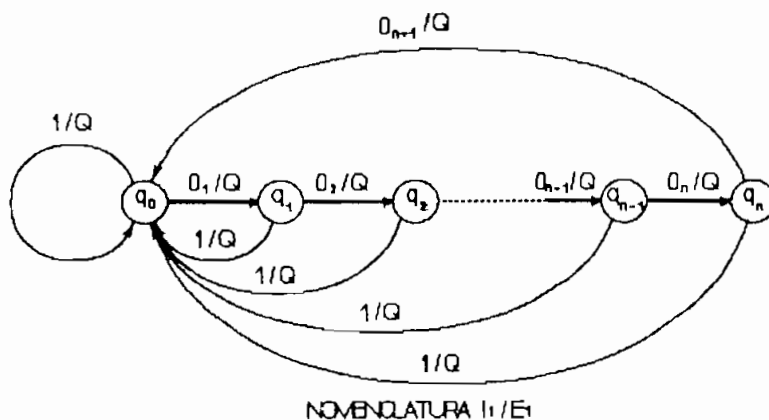


FIGURA 9.10 Identificación de secuencias de 0s de la MEF(1) del Codificador.

Donde:

$I_1$  : Códigos de entrada que son monitoreados.

$E_1$  : Salida empleada por la MEF(1) para informar a la MEF(2) la "información relevante" del estado "q" al que pasa,



esta "información relevante" ha sido nominada como su estado "Q" según la siguiente definición:

ESTADO (q) de la MEF(1)	ESTADO (Q) QUE LA MEF(2) LEE DE LA MEF(1)
$q_0$	$Q_0$
$q_1$	[Aún no se han identificado (n) 0s consecutivos]
'	
'	
$q_{n-1}$	
$q_n$	$Q_1$
	[Se han identificado (n) 0s consecutivos]

TABLA 9.2 Definición de los estados Q de la MEF(1) del Codificador.

La forma como emplea esta información la MEF(2) será estudiada en detalle cuando ésta se diseñe.

Considerando que se va a construir un codificador HDBn con "n" variable, se requiere una entrada adicional "n" (Fig.9.11(a)) que indique a la MEF(1) el número de 0s consecutivos que debe detectar. Para un valor de "n" cualquiera, el diagrama secuencial la MEF(1) se indica en la Fig.9.11(b).

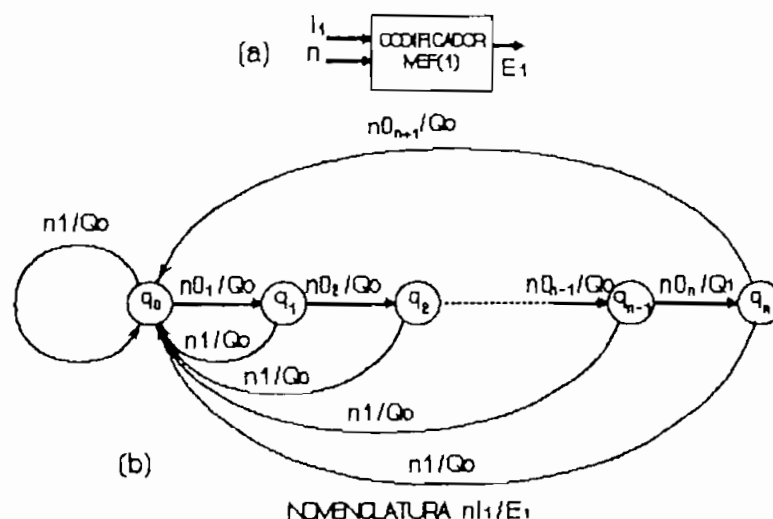


FIGURA 9.11 (a) Terminales de la MEF(1), (b) Diagrama secuencial para "n" cualquiera.

Para el caso del presente diseño, el valor de "n" se ha limitado a  $n = 3, 2, 1$  y 3. Así:

i)

CASO  $n=3$

n	ESTADO $q_i$	ENTRADA $I_i$	ESTADO $q_{i+1}$	ESTADO Q Salida $E_i$
3	$q_0$	0	$q_1$	$Q_0$
3	$q_0$	1	$q_0$	$Q_0$
3	$q_1$	0	$q_2$	$Q_0$
3	$q_1$	1	$q_0$	$Q_0$
3	$q_2$	0	$q_3$	$Q_1$
3	$q_2$	1	$q_0$	$Q_0$
3	$q_3$	0	$q_0$	$Q_0$
3	$q_3$	1	$q_0$	$Q_0$

TABLA 9.3 Transición de estados de la MEF(1) del Codificador, caso  $n=3$ .

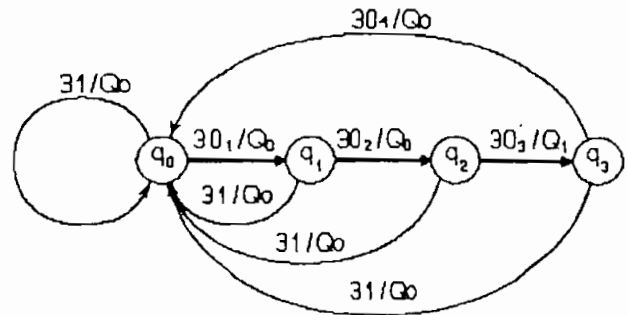


FIGURA 9.12 Diagrama secuencial de la MEF(1) del Codificador, caso  $n=3$ .

ii)

CASO  $n=2$

n	ESTADO $q_i$	ENTRADA $I_i$	ESTADO $q_{i+1}$	ESTADO Q Salida $E_i$
2	$q_0$	0	$q_1$	$Q_0$
2	$q_0$	1	$q_0$	$Q_0$
2	$q_1$	0	$q_2$	$Q_1$
2	$q_1$	1	$q_0$	$Q_0$
2	$q_2$	0	$q_0$	$Q_0$
2	$q_2$	1	$q_0$	$Q_0$

TABLA 9.4 Transición de estados de la MEF(1) del Codificador, caso  $n=2$ .

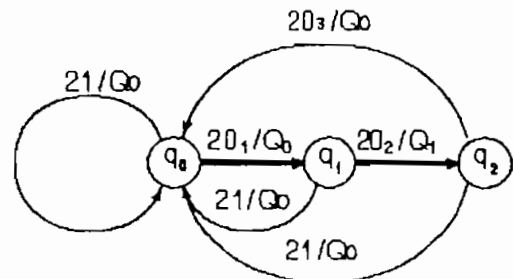


FIGURA 9.13 Diagrama secuencial de la MEF(1) del Codificador, caso  $n=2$ .

iii)

CASO  $n=1$

n	ESTADO $q_1$	ENTRADA $I_1$	ESTADO $q_{1+1}$	ESTADO Q Salida $E_1$
1	$q_0$	0	$q_1$	$Q_1$
1	$q_0$	1	$q_0$	$Q_0$
1	$q_1$	0	$q_0$	$Q_0$
1	$q_1$	1	$q_0$	$Q_0$

TABLA 9.5 Transición de estados de la MEF(1) del Codificador caso,  $n=1$ .

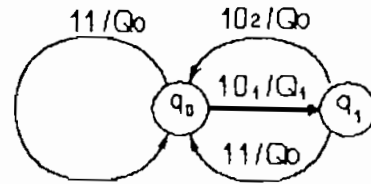


FIGURA 9.14 Diagrama secuencial de la MEF(1) del Codificador, caso  $n=1$ .

iv)

CASO  $n=0$

n	ESTADO $q_1$	ENTRADA $I_1$	ESTADO $q_{1+1}$	ESTADO Q Salida $E_1$
0	$q_0$	0	$q_0$	$Q_0$
0	$q_0$	1	$q_0$	$Q_0$

TABLA 9.6 Transición de estados de la MEF(1) del Codificador, caso  $n=0$ .

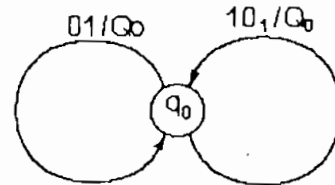


FIGURA 9.15 Diagrama secuencial de la MEF(1) del Codificador, caso  $n=0$ .

En el caso  $n=0$  los estados  $Q_0$  y  $Q_1$  se confunden en un solo estado  $Q$  ya que el estado  $q$  que reportan a la MEF(2) es único, por lo que resulta indiferente nominarlo como  $Q_0$  ó  $Q_1$ .

#### v) Diagrama y tabla de estados generales

Resumiendo los diagramas de las Figuras 9.12 a 9.15, el diagrama de estados de la MEF(1) del codificador HDBn es como se indica en la Fig.9.16.

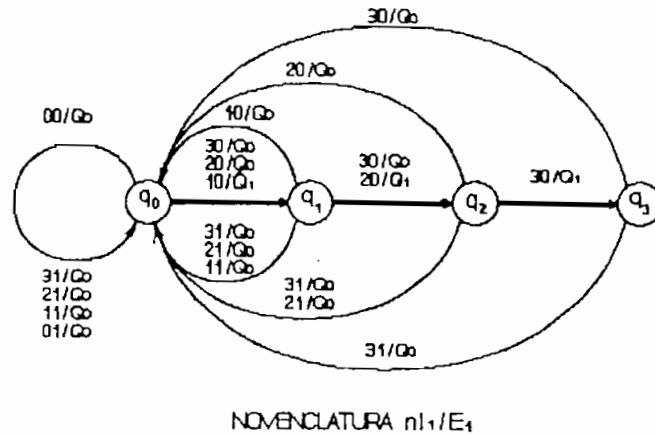


FIGURA 9.16 Diagrama de estados de la MEF(1) del Codificador HDBn.

Para codificar en forma binaria los elementos de la MEF(1) se ha adoptado la nomenclatura que se indica en la Tabla 9.7.

ELEMENTOS DE MEF(1)	VALORES POSIBLES	# DE BITS DE NOTACION	NOTACION
ESTADOS Q	$Q_0$ $Q_1$	1	0 1
ESTADOS q	$q_0$ $q_1$ $q_2$ $q_3$	2	00 01 10 11
ENTRADA n	$n=0$ $n=1$ $n=2$ $n=3$	2	00 01 10 11

TABLA 9.7 Codificación binaria de los estados de la MEF(1) del Codificador.

En consecuencia, la tabla de estados de la MEF(1) y su codificación binaria correspondiente son como se indica en las Tablas 9.8 y 9.9 respectivamente.

n	ESTADO $q_i$	ENTRADA $I_i$	ESTADO $q_{i+1}$	ESTADO Q Salida $E_i$
0	$q_0$	0	$q_0$	$Q_0$
0	$q_0$	1	$q_0$	$Q_0$
1	$q_0$	0	$q_1$	$Q_1$
1	$q_0$	1	$q_0$	$Q_0$
1	$q_1$	0	$q_0$	$Q_0$
1	$q_1$	1	$q_0$	$Q_0$
2	$q_0$	0	$q_1$	$Q_0$
2	$q_0$	1	$q_0$	$Q_0$
2	$q_1$	0	$q_2$	$Q_1$
2	$q_1$	1	$q_0$	$Q_0$
2	$q_2$	0	$q_0$	$Q_0$
2	$q_2$	1	$q_0$	$Q_0$
3	$q_0$	0	$q_1$	$Q_0$
3	$q_0$	1	$q_0$	$Q_0$
3	$q_1$	0	$q_2$	$Q_0$
3	$q_1$	1	$q_0$	$Q_0$
3	$q_2$	0	$q_3$	$Q_1$
3	$q_2$	1	$q_0$	$Q_0$
3	$q_3$	0	$q_0$	$Q_0$
3	$q_3$	1	$q_0$	$Q_0$

TABLA 9.8 Transición de estados de la MEF(1) del Codificador HDBn.

n	ESTADO $q_i$	ENTRADA $I_i$	ESTADO $q_{i+1}$	ESTADO Q Salida $E_i$
	$A_i A_{i-1}$	$A_i$	$D_i D_{i-1}$	$D_i$
0 0	0 0	0	0 0	0
0 0	0 0	1	0 0	0
0 1	0 0	0	0 1	1
0 1	0 0	1	0 0	0
0 1	0 1	0	0 0	0
0 1	0 1	1	0 0	0
1 0	0 0	0	0 1	0
1 0	0 0	1	0 0	0
1 0	0 1	0	1 0	1
1 0	0 1	1	0 0	0
1 0	1 0	0	0 0	0
1 0	1 0	1	0 0	0
1 1	0 0	0	0 1	0
1 1	0 0	1	0 0	0
1 1	0 1	0	1 0	0
1 1	0 1	1	0 0	0
1 1	1 0	0	1 1	1
1 1	1 0	1	0 0	0
1 1	1 1	0	0 0	0
1 1	1 1	1	0 0	0

TABLA 9.9 Transición de estados de la MEF(1) del Codificador HDBn (codificación binaria).

b) Estructura esquemática

De acuerdo a la tabla de estados 9.9 deducida para la MEF(1) del Codificador HDBn tiene la estructura indicada en la Fig.9.17.

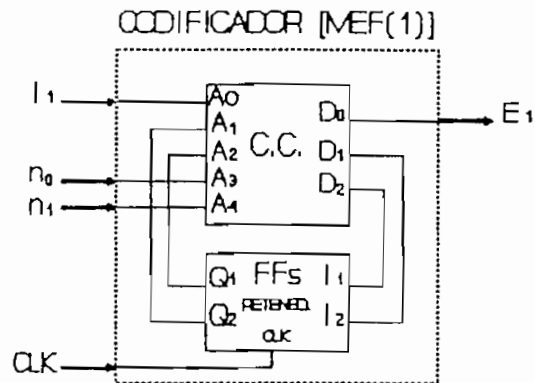


FIGURA 9.17 Estructura de la MEF(1) del Codificador HDBn.

c) Mapas de Karnaugh

Los valores de las salidas  $D_2D_1D_0$  del circuito combinacional CC de la Fig.9.17 deben ser correspondientes con las combinaciones lógicas de sus entradas  $A_4 \dots A_0$  de acuerdo con la Tabla 9.9, las ecuaciones que las relacionan han sido deducidas a partir de los Mapas de Karnaugh correspondientes.

Para facilidad de ilustración se ha construídos la Tabla 9.10 considerando las combinaciones lógicas de las variables de entrada ausentes en la Tabla 9.9, a cuyas salidas se ha asignado la combinación XXX (condición no importa).

ENTRADAS					SALIDAS		
A <sub>4</sub>	A <sub>3</sub>	A <sub>2</sub>	A <sub>1</sub>	A <sub>0</sub>	D <sub>2</sub>	D <sub>1</sub>	D <sub>0</sub>
0	0	0	0	0	0	0	0
0	0	0	0	1	0	0	0
0	0	0	1	0	X	X	X
0	0	0	1	1	X	X	X
0	0	1	0	0	X	X	X
0	0	1	0	1	X	X	X
0	0	1	1	0	X	X	X
0	0	1	1	1	X	X	X
0	1	0	0	0	0	1	1
0	1	0	0	1	0	0	0
0	1	0	1	0	0	0	0
0	1	0	1	1	0	0	0
0	1	1	0	0	X	X	X
0	1	1	0	1	X	X	X
0	1	1	1	0	X	X	X
0	1	1	1	1	X	X	X
1	0	0	0	0	0	1	0
1	0	0	0	1	0	0	0
1	0	0	1	0	1	0	1
1	0	0	1	1	0	0	0
1	0	1	0	0	0	0	0
1	0	1	0	1	0	0	0
1	0	1	1	0	X	X	X
1	0	1	1	1	X	X	X
1	1	0	0	0	0	1	0
1	1	0	0	1	0	0	0
1	1	0	1	0	1	0	0
1	1	0	1	1	0	0	0
1	1	1	0	0	1	1	1
1	1	1	0	1	0	0	0
1	1	1	1	0	0	0	0
1	1	1	1	1	0	0	0

TABLA 9.10 Transición de estados de la MEF(1) del Codificador (se han considerado condiciones no importa)

i) Salida: D<sub>2</sub>

		A <sub>3</sub> A <sub>2</sub>						A <sub>3</sub> A <sub>2</sub>					
		00		01		11		10				A <sub>4</sub> =1	
A <sub>1</sub> A <sub>0</sub>		00	01	11	10	A <sub>1</sub> A <sub>0</sub>		00	01	11	10		
00		0	X	X	0	00		0	0	1	0		
01		0	X	X	0	01		0	0	0	0		
11		X	X	X	0	11		0	X	0	0		
10		X	X	X	0	10		1	X	0	1		

$$D_2 = A_4 \bar{A}_2 \bar{A}_1 \bar{A}_0 + A_3 A_2 \bar{A}_1 \bar{A}_0$$

ii) Salida: D<sub>1</sub>

		A <sub>3</sub> A <sub>2</sub>						A <sub>3</sub> A <sub>2</sub>					
		00		01		11		10				A <sub>4</sub> =1	
A <sub>1</sub> A <sub>0</sub>		00	01	11	10	A <sub>1</sub> A <sub>0</sub>		00	01	11	10		
00		0	X	X	1	00		1	0	1	1		
01		0	X	X	0	01		0	0	0	0		
11		X	X	X	0	11		0	X	0	0		
10		X	X	X	0	10		0	X	0	0		

$$D_1 = A_3 \bar{A}_1 \bar{A}_0 + A_4 \bar{A}_2 \bar{A}_1 \bar{A}_0$$

iii) Salida: D<sub>0</sub>

		A <sub>3</sub> A <sub>2</sub>						A <sub>3</sub> A <sub>2</sub>					
		00		01		11		10				A <sub>4</sub> =1	
A <sub>1</sub> A <sub>0</sub>		00	01	11	10	A <sub>1</sub> A <sub>0</sub>		00	01	11	10		
00		0	X	X	1	00		0	0	1	0		
01		0	X	X	0	01		0	0	0	0		
11		X	X	X	0	11		0	X	0	0		
10		X	X	X	0	10		1	X	0	0		

$$D_0 = \bar{A}_3 \bar{A}_1 \bar{A}_0 + A_3 \bar{A}_2 \bar{A}_1 \bar{A}_0 + \bar{A}_4 \bar{A}_3 \bar{A}_1 \bar{A}_0$$

### 9.2.3 Diseño de la MEF(2)

#### a) Diagramas y tablas de estados

La máquina de estados finitos MEF(2) básicamente codifica los bits de entrada "1" como "B" ó "V", y "0" como "O", además de introducir modificaciones en los símbolos codificados anteriormente de ser necesario. Estos códigos "B" ó "V" pueden catalogarse en base a dos características:

- i) Signo, que puede ser "positivo (+)" ó "negativo (-)"
- ii) Paridad, puesto que cada "1" que va siendo codificado puede ser "par (p)" ó "impar (i)".

Estas dos características son mutuamente independientes, por lo que se pueden combinar en cuatro estados posibles que son los estados de la MEF(2). A saber:

		PARIDAD	
		PAR	IMPAR
SIGNO	+	p+	i+
	-	p-	i-

TABLA 9.11 Estados de la MEF(2) del Codificador HDBn.

De acuerdo a la Fig.9.9, el módulo funcional de la MEF(2) tiene la estructura de la Fig.9.18.

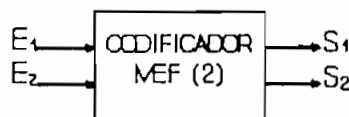


FIGURA 9.18 Módulo funcional de la MEF(2) del Codificador HDBn.



Donde:

$E_1$ : Entrada proveniente de la MEF(1).

$E_2$ : Entada de bits a codificarse.

$S_1$ : Salida de codificación inmediata "a priori"

$S_2$ : Salida de modificación de símbolos codificados anteriormente

La MEF(2) opera en forma complementaria con la MEF(1) en función del estado  $Q$  que esta reporta, se tienen por tanto los casos:

i) Estado de la MEF(1):  $Q_0$

ii) Estado de la MEF(1):  $Q_1$

i) Estado de la MEF(1):  $Q_0$

Cuando la MEF(1) indica a través de  $E_1$  que aún no se han detectado  $(n)0$ s consecutivos en la secuencia de entrada, entonces  $E_1=Q_0$ , el diagrama de estados corresponde al de la Fig.9.19.

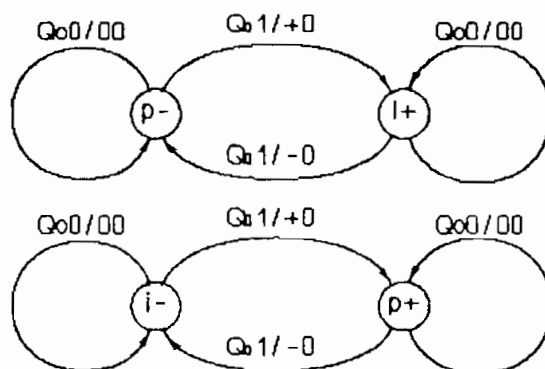


FIGURA 9.19 Diagrama de estados la MEF(2), caso  $E_1=Q_0$ .

Donde, cuando se recibe en la entrada  $E_1$  un "1", el estado de signo y el código de salida en  $S_1$  varían en forma alterna entre "+" y "-" al tiempo que también alterna su estado de paridad. En caso de tener  $E_2=0$  la salida  $S_1=0$  y el estado de paridad se conserva.

La salida  $S_2$  se mantiene en "0" puesto que no se requieren modificaciones a los códigos generados con anterioridad (ver Tabla 9.1).

ii) Estado de la MEF(1): $Q_1$

La indicación de un estado  $Q_1$  en la MEF(1) a través de la entrada  $E_1$  de la MEF(2) advierte que se han detectado  $(n)0$ s consecutivos. Dependiendo del código que llegue luego del  $(n)0$ , se tienen los siguientes casos:

ii.1) Caso  $E_2 = 1$

SECUENCIA DE ENTRADA	CODIFICACION	OBSERVACION
1 0 <sub>1</sub> 0 <sub>2</sub> ..... 0 <sub>n-1</sub> 0 <sub>n</sub> 1	B 0 <sub>1</sub> 0 <sub>2</sub> ..... 0 <sub>n-1</sub> 0 <sub>n</sub> B  (+ 0 0            0 0 -) (- 0 0            0 0 +)	NO EXISTE VIOLACION DE ALTERNABILIDAD DE SIGNO

TABLA 9.12 Codificación de una secuencia de  $(n)0$ s, caso  $E_1=Q_1$ ,  $E_2=1$ .

En este caso a  $E_2$  ha llegado un "1" luego de  $(n)0$ s consecutivos, para las salidas se tiene:

$S_1=B$             que puede ser "+" ó "-" manteniendo alternabilidad con el código "B" inmediatamente anterior.

$S_2=0$             puesto que no se requiere cambiar la codificación de  $0_1$ .

Los estados mantienen la alternabilidad de signo y de paridad como se indica en la Fig.9.20.

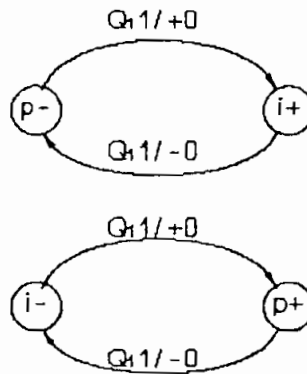


FIGURA 9.20 Diagrama de estados de la MEF(2), caso  $E_1=Q_1$ ,  $E_2=1$ .

ii.2) Caso  $E_2 = 0$ ,  $l_0$ (impar)

SECUENCIA DE ENTRADA	CODIFICACION	OBSERVACION
$l_0 \ 0_1 \ 0_2 \ \dots \ 0_{n-1} \ 0_n \ 0_{n+1}$	$B_0 \ 0_1 \ 0_2 \ \dots \ 0_{n-1} \ 0_n \ V_{n+1}$ $(+ \ 0 \ 0 \ \dots \ 0 \ 0 \ +)$ $(- \ 0 \ 0 \ \dots \ 0 \ 0 \ -)$	. EXISTE VIOLACION DE ALTERNABILIDAD DE SIGNO . $l_0$ : IMPAR

TABLA 9.13 Codificación de una secuencia de  $(n)0$ s, caso  $E_1=Q_1$ ,  $E_2=0$ ,  $l_0$  (impar).

En este caso luego del  $(n)0$  ha llegado por  $E_2$  un  $(n+1)0$  y el último "1" en llegar fue  $l_0$  (impar). Por tanto:

$S_1=V_{n+1}$  se viola la alternabilidad de signo con el último "B" codificado:  $B_0$ .

$S_2=0$  puesto que no se requiere cambiar la codificación de  $0_1$ .

En el diagrama de estados (Fig.9.21) se mantiene el estado de signo y cambia el de paridad a "par" ya que

indefectiblemente al terminar la secuencia de codificación el primer "1" de la siguiente secuencia pasará siempre a un estado "impar".

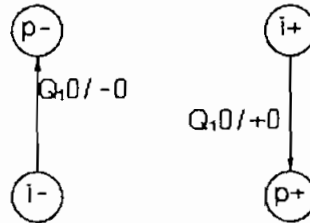


FIGURA 9.21 Diagrama de estados de la MEF(2), caso  $E_1=\theta_1$ ,  $E_2=0$ ,  $l_0$ (impar).

ii.3) Caso  $E_2 = 0$ ,  $l_0$ (par)

SECUENCIA DE ENTRADA	CODIFICACION	OBSERVACION
$l_0 \ 0_1 \ 0_2 \ \dots \ 0_{n-1} \ 0_n \ 0_{n+1}$	$B_0 \ B_1 \ 0_2 \ \dots \ 0_{n-1} \ 0_n \ V_{n+1}$ (+ - 0 ..... 0 0 -) (- + 0 ..... 0 0 +)	. EXISTE VIOLACION DE ALTERNABILIDAD DE SIGNO . $l_0$ : PAR

TABLA 9.14 Codificación de una secuencia de (n)0s, caso  $E_1=\theta_1$ ,  $E_2=0$ ,  $l_0$  (par).

En este caso  $E_2=0$ , en contraste con el caso anterior el último "1" en llegar fue  $l_0$ (par). En las salidas se tiene:

$S_1=V_{n+1}$  que es un código "+" ó "-" con alternabilidad de signo violada con respecto a  $B_1$ .

$S_2=B$  puesto que antes de identificarse la secuencia de (n)0s consecutivos  $0_1$  fue codificado como "0" según el diagrama de la Fig.9.19, esta codificación debe ser cambiada a  $B_1$  (ver Tablas 9.1 y 9.14).

En el diagrama de estados de la Fig.9.22 la secuencia debe terminar en un estado "par" por lo indicado para el caso anterior, el estado de signo también cambia puesto que  $V_{n+1}$  es un signo alternado con respecto al de  $B_0$  (ver Tabla 9.14).

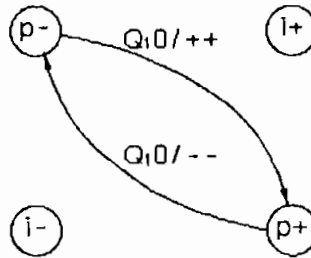


FIGURA 9.22 Diagrama de estados de la MEF(2), caso  $E_1=0_1$ ,  $E_2=0$ ,  $l_0(\text{par})$ .

Agrupando los diagramas de las Figuras 9.19 a 9.22, la secuencia de estados de la MEF(2) del Codificador HDBn es como se indica en la Fig.9.23.

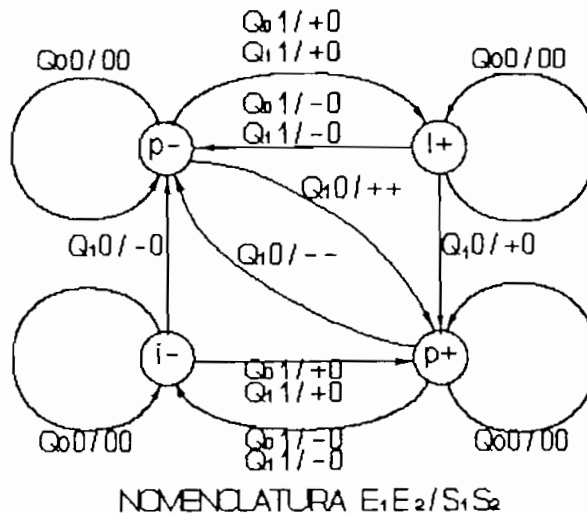


FIGURA 9.23 Diagrama de estados de la MEF(2) del Codificador HDBn.

Nótese que esta máquina es independiente del valor de "n" por lo que puede usarse sin variación para cualquier tipo de codificación HDBn.

El diagrama de estados de la Fig.9.23 puede caracterizarse mediante la Tabla de Estados 9.15.

ESTADO $q_1$	ESTADO Q DE LA MEF(1) $E_1$	ENTRADA $E_2$	ESTADO $q_{1+1}$	SALIDAS	
				$S_1$	$S_2$
$p^-$	$Q_0$	0	$p^-$	0	0
$p^-$	$Q_0$	1	$i^+$	+	0
$p^-$	$Q_1$	0	$p^+$	+	+
$p^-$	$Q_1$	1	$i^+$	+	0
$p^+$	$Q_0$	0	$p^+$	0	0
$p^+$	$Q_0$	1	$i^-$	-	0
$p^+$	$Q_1$	0	$p^-$	-	-
$p^+$	$Q_1$	1	$i^-$	-	0
$i^-$	$Q_0$	0	$i^-$	0	0
$i^-$	$Q_0$	1	$p^+$	+	0
$i^-$	$Q_1$	0	$p^-$	-	0
$i^-$	$Q_1$	1	$p^+$	+	0
$i^+$	$Q_0$	0	$i^+$	0	0
$i^+$	$Q_0$	1	$p^-$	-	0
$i^+$	$Q_1$	0	$p^+$	+	0
$i^+$	$Q_1$	1	$p^-$	-	0

TABLA 9.15 Transición de estados de la MEF(2) del Decodificador HDBn.

A fin de codificar en forma binaria los elementos de la de la Tabla 9.15 se ha adoptado la nomenclatura indicada en la Tabla 9.16.

ELEMENTOS DE MEF(2)	VALORES POSIBLES	# DE BITS DE NOTACION	NOTACION
ESTADOS DE SIGNO Y PARIDAD	p- p+ i- i+	2	00 01 10 11
ESTADOS Q	Q <sub>0</sub> Q <sub>1</sub>	1	0 0
SALIDAS S <sub>1</sub> y S <sub>0</sub>	0 + -	2	00 01 10

TABLA 9.16 Codificación binaria de los estados de la MEF(2) del Decodificador HDBn.

Considerando esta nomenclatura, la Tabla 9.15 se replantea en base a códigos binarios como se indica en la Tabla 9.17.

ESTADO q <sub>1</sub>	ESTADO Q DE LA MEF(1) E <sub>1</sub>	ENTRADA E <sub>2</sub>	ESTADO q <sub>1+1</sub>	SALIDAS	
				S <sub>1</sub>	S <sub>2</sub>
A <sub>3</sub> A <sub>2</sub>	A <sub>1</sub>	A <sub>0</sub>	D <sub>3</sub> D <sub>4</sub>	D <sub>3</sub> D <sub>2</sub>	D <sub>1</sub> D <sub>0</sub>
0 0	0	0	0 0	0 0	0 0
0 0	0	1	1 1	0 1	0 0
0 0	1	0	0 1	0 1	0 1
0 0	1	1	1 1	0 1	0 0
0 1	0	0	0 1	0 0	0 0
0 1	0	1	1 0	1 0	0 0
0 1	1	0	0 0	1 0	1 0
0 1	1	1	1 0	1 0	0 0
1 0	0	0	1 0	0 0	0 0
1 0	0	1	0 1	0 1	0 0
1 0	1	0	0 0	1 0	0 0
1 0	1	1	0 1	0 1	0 0
1 1	0	0	1 1	0 0	0 0
1 1	0	1	0 0	1 0	0 0
1 1	1	0	0 1	0 1	0 0
1 1	1	1	0 0	1 0	0 0

TABLA 9.17 Transición de estados de la MEF(2) del Codificador HDBn (codificación binaria).

b) Estructura esquemática

De acuerdo al número de entradas y salidas necesarias, indicadas como  $A_3 \dots A_0$  y  $D_5 \dots D_0$ , en la Tabla 9.17, la MEF(2) del codificador tiene la estructura esquemática de la Fig.9.24.

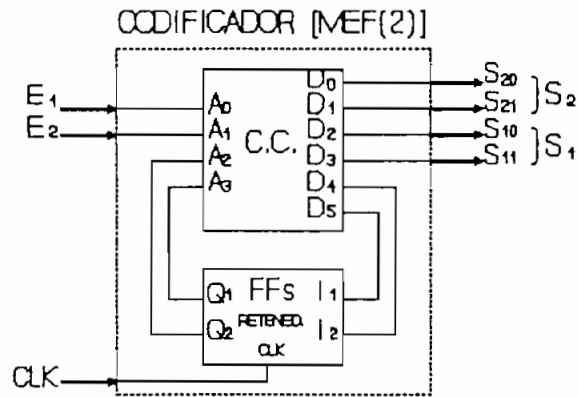


FIGURA 9.24 Estructura esquemática de la MEF(2) del Codificador HDBn.

c) Mapas de Karnaugh

Las salidas  $D_5 \dots D_0$  del circuito combinacional CC de la Fig.9.21 deben corresponder a las combinaciones lógicas de las variables de entrada  $A_3 \dots A_0$  indicadas en la Tabla 9.17 que para ilustración se ha reformulado en la Tabla 9.18.



ENTRADAS	SALIDAS
$A_3A_2A_1A_0$	$D_3D_4D_3D_2D_1D_0$
0 0 0 0	0 0 0 0 0 0
0 0 0 1	1 1 0 1 0 0
0 0 1 0	0 1 0 1 0 1
0 0 1 1	1 1 0 1 0 0
0 1 0 0	0 1 0 0 0 0
0 1 0 1	1 0 1 0 0 0
0 1 1 0	0 0 1 0 1 0
0 1 1 1	1 0 1 0 0 0
1 0 0 0	1 0 0 0 0 0
1 0 0 1	0 1 0 1 0 0
1 0 1 0	0 0 1 0 0 0
1 0 1 1	0 1 0 1 0 0
1 1 0 0	1 1 0 0 0 0
1 1 0 1	0 0 1 0 0 0
1 1 1 0	0 1 0 1 0 0
1 1 1 1	0 0 1 0 0 0

TABLA 9.18 Transición de estados de la MEF(2) del Decodificador (se han considerado condiciones no importa).

i) Salida:  $D_3$

$A_3A_2$	$A_1A_0$			
	00	01	11	10
00	0	0	1	1
01	1	1	0	0
11	1	1	0	0
10	0	0	0	0

$$D_3 = A_3\bar{A}_1\bar{A}_0 + \bar{A}_3A_0$$

ii) Salida:  $D_4$

$A_3A_2$	$A_1A_0$			
	00	01	11	10
00	0	1	1	0
01	1	0	0	1
11	1	0	0	1
10	1	0	1	0

$$D_4 = A_2\bar{A}_1\bar{A}_0 + \bar{A}_3\bar{A}_2A_1 + A_3A_2\bar{A}_0 + \bar{A}_2A_0$$

iii) Salida:  $D_3$

$A_3A_2$	$A_1A_0$			
	00	01	11	10
00	0	0	0	0
01	0	1	1	0
11	0	1	1	0
10	0	1	0	1

$$D_3 = A_2A_0 + \bar{A}_3A_2A_1 + A_3\bar{A}_2A_1\bar{A}_0$$

iv) Salida:  $D_2$

$A_3A_2$	$A_1A_0$			
	00	01	11	10
00	0	0	0	0
01	1	0	0	1
11	1	0	0	1
10	1	0	1	0

$$D_2 = \bar{A}_3\bar{A}_2A_1 + A_3A_2A_1\bar{A}_0 + \bar{A}_2A_0$$

Las ecuaciones para  $D_1$  y  $D_0$  pueden deducirse por simple inspección considerando el único "1" de cada columna. Así:

$$D_1 = \bar{A}_3 A_2 A_1 \bar{A}_0$$

$$D_0 = \bar{A}_3 \bar{A}_2 A_1 \bar{A}_0$$

#### 9.2.4 Etapa de salida

En el esquema del codificador HDB<sub>n</sub> de la Fig.9.8 los buses de salida llevan los códigos "+" y/o "-", la representación de tales códigos en forma binaria requiere de 2 bits de acuerdo a la Tabla 9.16, por tanto se requieren dos líneas de señal por cada bus como se indica en la Fig.9.25.

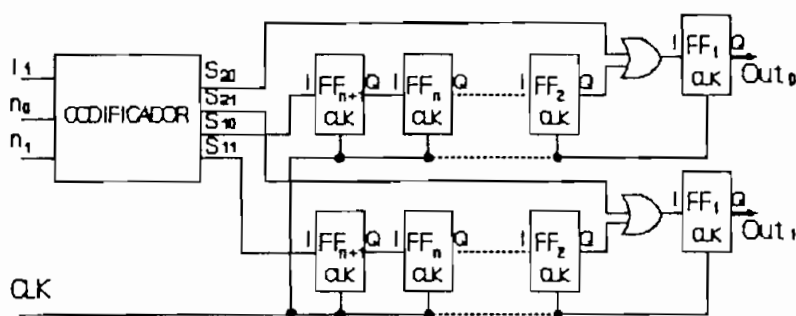


FIGURA 9.25 Salida del codificador para "n" cualquiera.

En el diagrama de la Fig.9.25 se han colocado  $(n+1)$  FFs en cascada en la salida del codificador HDB<sub>n</sub> con "n" cualquiera, en el presente diseño se tiene  $n=3, 2, 1$  y  $0$ , por lo que el número de FFs es variable (Fig.9.26(a)). Para ello se ha colocado un Multiplexor a la salida de los FFs que permita indicar el último FF de la cascada según el valor de "n"

seleccionado (Fig.9.26(b)).

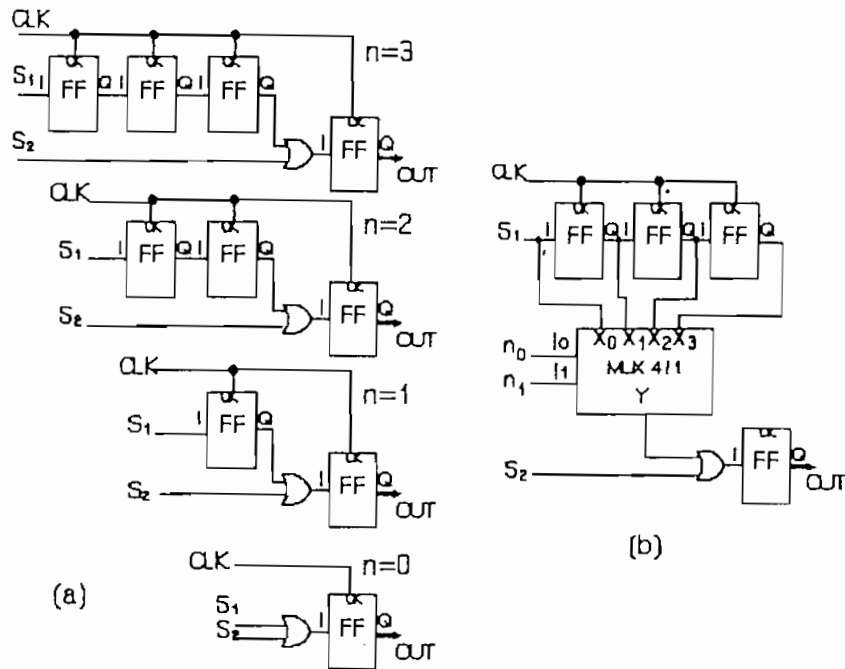


FIGURA 9.26 Salida del codificador para  $n=3, 2, 1, 0$  (a) y con "n" seleccionable (b).

### 9.2.5 Estructura del Codificador HDBn

Interconectando los módulos de los esquemas de las máquinas MEF(1) de la Fig.9.17, de la MEF(2) de la Fig.9.24 y de la etapa de salida de la Fig.9.26, se obtiene la estructura general del Codificador indicada en la Fig.9.27.

Para el funcionamiento correcto del circuito, se deben considerar los siguientes detalles adicionales:

- a) Los datos que entran a la MEF(2) que son: el resumen de una secuencia que reporta la MEF(1) y el dato que sigue a esta secuencia, pueden hallarse desplazados en el tiempo, es necesario que estos datos sean leídos simultáneamente por la MEF(2) para evitar desajustes en su funcionamiento. Para corregir este desajuste se han colocado FF(a) y FF(b) sincronizados por el reloj externo del circuito.
- b) En las Tablas de estados 9.9 para la MEF(1) y 9.17 para la MEF(2), se tiene que el estado inicial de cada máquina corresponde a aquel en que las entradas  $A_3 \dots A_0$  se hallan en 0, sin embargo cuando el codificador inicia su funcionamiento el estado lógico del que arranca cada máquina y su funcionamiento posterior es incierto, para evitar ello se ha aumentado la señal de START de manera tal que:
- i) Si  $START=0$  las máquinas se mantienen en su estado inicial.
  - ii) Si  $START=1$  las máquinas trabajan normalmente.
- De este modo se asegura que al arrancar la máquina siempre partirá de su estado inicial.

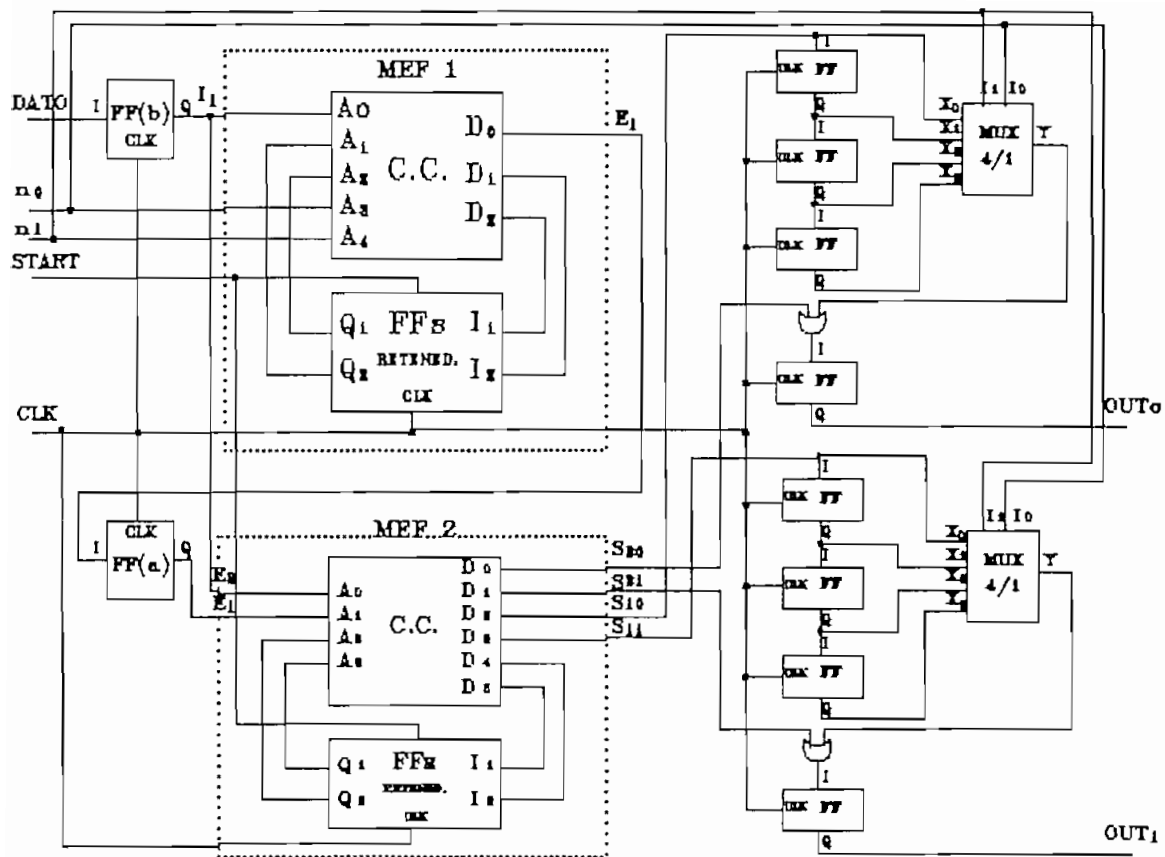


FIGURA 9.27 Estructura del Codificador HDBn.



secuencia:  $0_2 0_3 \dots 0_n$ , es decir  $(n-1)$  0s consecutivos.  
 En el caso (2), el código "V" está antecedido por una  
 secuencia:  $0_1 0_2 \dots 0_n$ , es decir  $(n)$  0s consecutivos.

De acuerdo a lo visto en el Codificador, para los casos  
 (1) y (2) la secuencia antes de la codificación es:

$$\frac{1/0 \ 1/0 \ 1/0 \ \dots \ 1/0 \ 1 \ 0_1 0_2 \dots 0_n 0_{n+1}}{\text{(I) \hspace{15em} (II)}}$$

Para diseñar el decodificador se plantea como primera  
 aproximación el módulo funcional que decodifica la secuencia  
 (I) (Fig.9.28)

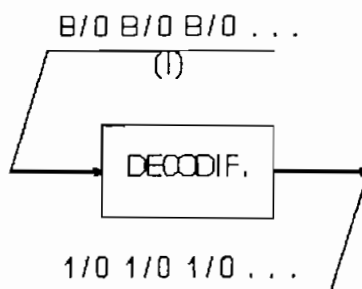


FIGURA 9.28 Módulo funcional del decodificador HD8n (primera aproximación).

Al llegar la secuencia (II) se pueden presentar los  
 casos (1) ó (2), como se grafica en la Fig.9.29.

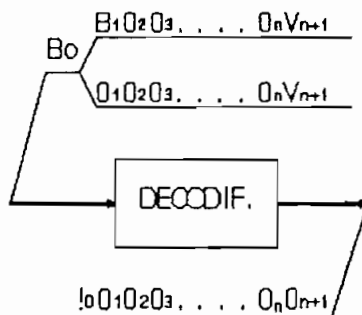


FIGURA 9.29 Módulo funcional del decodificador para secuencias críticas.

La formulación de la Fig.9.29 podría invertirse como se indica en la Fig.9.30.

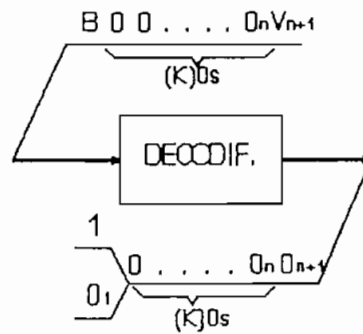


FIGURA 9.30 Reformulación del módulo funcional del Decodificador.

Donde:

K: Número de 0s consecutivos detectados antes de un código de violación "V".

De modo que:

- a) Si  $K = n-1$ . Entonces el código "B" en la entrada corresponde a "B<sub>1</sub>" de la secuencia (II) del caso (1) y se decodifica como "0<sub>1</sub>".
- b) Si  $K = n$ . Entonces el código "B" en la entrada se halla fuera de la secuencia (II) del caso (2) y se decodifica como "1".

Se concluye que la codificación del último código "B" anterior a la secuencia de (K)0s, no puede decidirse "a priori" ya que depende del número "K" de 0s consecutivos "posteriores". Por tanto es necesario:



- a) Retener los últimos  $(n+1)$  bits decodificados a fin de poder modificar la decodificación del bit  $B$  previo a la secuencia de 0s, de presentarse el caso (1) anterior.
- b) Contar con una salida adicional que realice esta modificación.

De acuerdo a estos criterios, el decodificador tiene la estructura esquemática de la Fig.9.31.

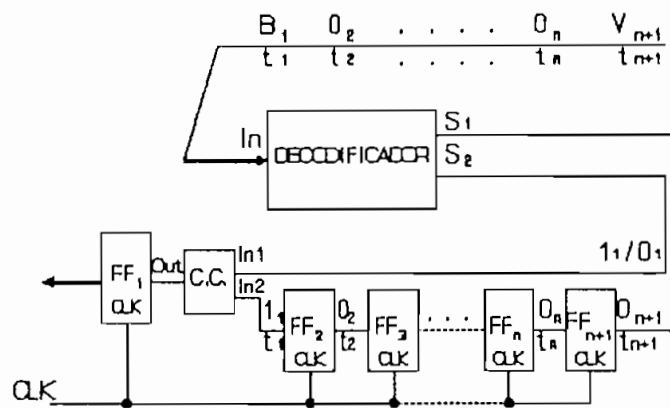


FIGURA 9.31 Esquema del decodificador HDBn.

Donde:

- a) Los flip-flops (FFs) retienen los  $(n+1)$  últimos bits decodificados y sincronizan su avance según una secuencia de reloj externa.
- b) El Decodificador cuenta con dos salidas:
  - $S_1$ : Empleada para generar los códigos resultantes de la decodificación "a priori" de los códigos entrantes.
  - $S_2$ : Empleada para modificar al bit que ingresa a FF1 cuando se presenta la secuencia (1), de manera que:

$S_2 = 0_1$  Si se requiere modificar a  $0_1$  la decodificación del código  $B_1$  interpretado como "1" al inicio de la secuencia:  $B_1 0_2 \dots 0_n V_{n+1}$

$S_2 = 1$  En cualquier otro caso.

c) El Circuito Combinacional (CC) trabaja de manera que:

Si  $S_2=0=In1$  Out = In1 Si  $S_2=1=In1$  Out=In2

como se indica en la Tabla 9.19.

In2	In1	Out
0	0	0
1	0	0
0	1	0
1	1	1

TABLA 9.19 Funcionalidad del C.C. de la Fig.9.31.

Que corresponde a una compuerta AND.

Por lo que el esquema de la Fig.9.31 se modifica como se indica en la Fig.9.32.

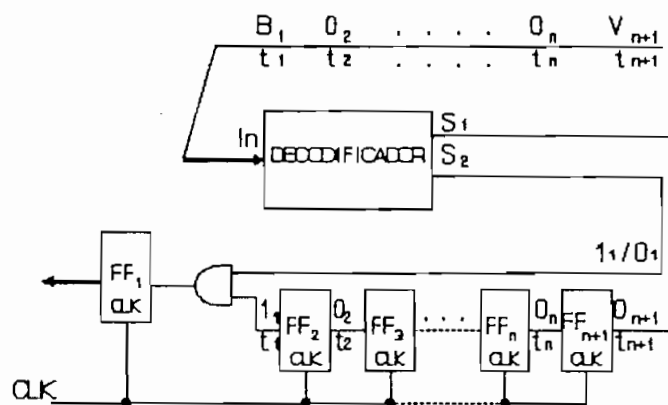


FIGURA 9.32 Esquema del decodificador HDBn de la Fig.9.31 reformulado.

En consecuencia, el circuito del decodificador debe realizar las siguientes tareas específicas:

- a) Monitorear los códigos entrantes a fin de detectar secuencias con  $(K=n)$  ó  $(K=n-1)$  Os consecutivos.
- b) Decodificar las secuencias:  $B/O \ B/O \ \dots \ B/O \ B$   
 como:  $1/O \ 1/O \ \dots \ 1/O \ 1$
- c) De detectarse la secuencias:  $B_1 \ O_2 \ \dots \ O_n \ V_{n+1} \ (1)$   
 ó:  $O_1 \ O_2 \ \dots \ O_n \ V_{n+1} \ (2)$   
 decodificarlas como:  $O_1 \ O_2 \ \dots \ O_n \ O_{n+1}$

Para realizar estas tareas el decodificador ha sido concebido con una filosofía de funcionamiento similar a la del codificador, es decir, se ha formulado en base a dos máquinas de estados finitos con funciones bien definidas y complementarias como se indica en la Fig.9.33.

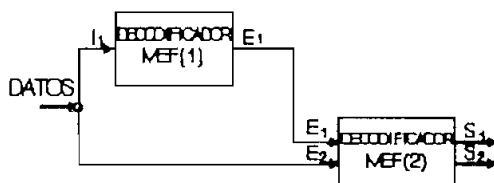


FIGURA 9.33 Decodificador HDBn descompuesto en dos máquinas de estados finitos.

Donde:

- a) La Máquina de Estados Finitos MEF(1) monitorea el número de Os consecutivos en las secuencias entrantes.
- b) La Máquina de Estados Finitos MEF(2) decodifica los códigos B/O como 1/O.
- c) La MEF(1) emplea el terminal E<sub>1</sub> para reportar a la MEF(2) la forma como van llegando las secuencias de Os consecutivos en los códigos de entrada.

### 9.3.2 Diseño de la MEF(1)

#### a) Diagramas y tablas de estados

Esta máquina monitorea los códigos entrantes con el fin de detectar secuencias de (n) ó (n-1) Os consecutivos. Por tanto en esta máquina únicamente interesa saber si el código entrante es o no "0", siendo irrelevante saber si el código distinto de "0" es "+" ó "-". El módulo funcional que representa a esta máquina se indica en la Fig.9.34.

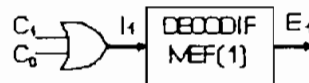


FIGURA 9.34 Módulo funcional de la MEF(1) del decodificador.

La compuerta OR a la entrada de  $I_1$  permite simplificar los códigos de entrada a la MEF(1) de modo que ya sean "+" ó "-" se los identifique siempre como "1", como se indica en la Tabla 9.18.

C	I1
0	0
+	1
-	1

(a)

C1	C0	I1
0	0	0
0	1	1
1	0	1
1	1	X

(b)

} COMPUERTA "OR"

TABLA 9.18 Códigos de entrada a la MEF(1) del Decodificador.

NOTA: Las combinaciones "+ = 01", "- = 10" y "0 = 00" son aquellas que se asignaron a estos códigos en el diseño del codificador (ver Tabla 9.16)

El diagrama de estados de la MEF(1) corresponde a un

identificador de secuencias de hasta (n)0s consecutivos como se indica en la Fig.9.35.

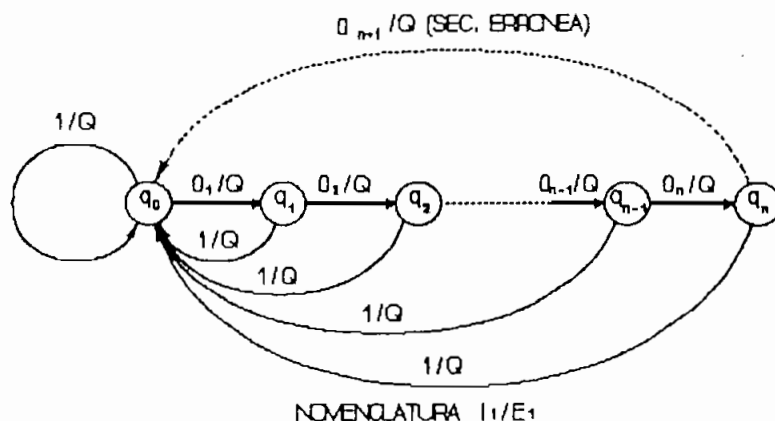


FIGURA 9.35 Identificador de secuencias de 0s de la MEF(1) del decodificador.

Donde:

- I<sub>1</sub>: Valor de entrada al codificador, que puede ser "0" o "1".
- E<sub>1</sub>: Salida empleada por la MEF(1) para indicar a la MEF(2) la "información relevante" del estado q al que pasa y que se ha denominado como el estado Q de la MEF(1) de acuerdo con la siguiente definición.

ESTADO (q) de la MEF(1)	ESTADO (Q) QUE LA MEF(2) LEE DE LA MEF(1)
q <sub>0</sub>	Q <sub>0</sub> [# de 0s consecutivos detectados < (n-1)]
q <sub>1</sub>	
'	
'	
q <sub>n-2</sub>	
q <sub>n-1</sub>	Q <sub>1</sub> [Se han identificado (n-1) 0s consecutivos]
q <sub>n</sub>	Q <sub>2</sub> [Se han identificado (n) 0s consecutivos]

TABLA 9.21 Definición de los estados Q de la MEF(1) del Decodificador HD0n.

La manera como se interpreta esta información y su influencia en el funcionamiento de la MEF(2) del decodificador se analizarán posteriormente.

En el diagrama de la Fig.9.35 también se tiene que al llegar un  $(n+1)0$  luego de los  $(n)0$ s consecutivos, este proviene de una secuencia "errónea" ya que, según la definición de los códigos HDBn, cualquier secuencia codificada "sin error" debe tener hasta  $(n)0$ s consecutivos. En el caso de la Fig.9.35 el  $(n+1)0$  se lo ha empleado para reinicializar la secuencia de conteo (en el diseño de la MEF(2) se realizará un análisis más detallado de las secuencias erróneas).

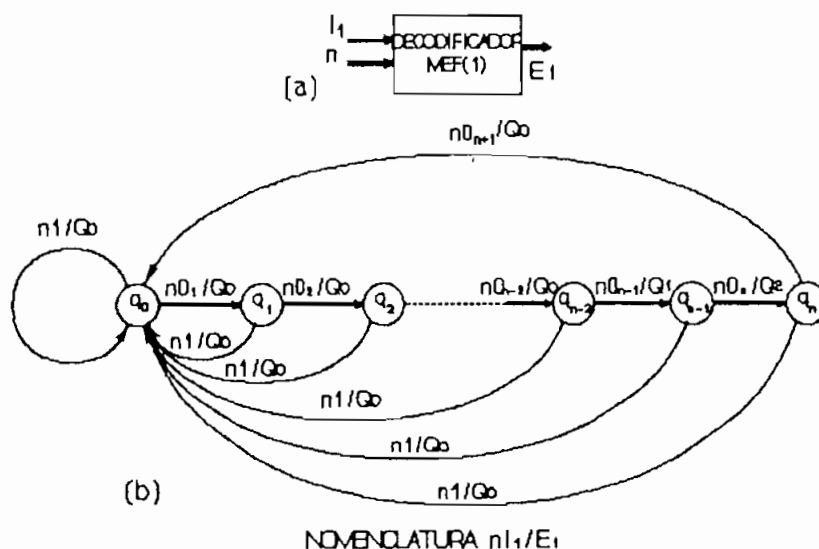


FIGURA 9.36 (a) Terminales de la MEF(1), (b) Diagrama secuencial para un "n" cualquiera.

Considerando que se va a diseñar un decodificador con "n" variable se requiere un bus de entrada adicional "n" (Fig.9.36(a)) que permita seleccionar el número máximo de 0s

consecutivos a identificarse. El diagrama secuencial de la MEF(1) para un "n" cualquiera se indica en la Fig.9.36(b).

El presente diseño ha sido propuesto para los casos particulares en que  $n=0, 1, 2$  y  $3$ , así:

i)

CASO  $n=3$

n	ESTADO $q_i$	ENTRADA $I_1$	ESTADO $q_{i+1}$	ESTADO Q Salida $E_1$
3	$q_0$	0	$q_1$	$0_0$
3	$q_0$	1	$q_0$	$0_0$
3	$q_1$	0	$q_2$	$0_1$
3	$q_1$	1	$q_0$	$0_0$
3	$q_2$	0	$q_3$	$0_2$
3	$q_2$	1	$q_0$	$0_0$
3	$q_3$	0	$q_0$	$0_0$
3	$q_3$	1	$q_0$	$0_0$

TABLA 9.22 Transición de estados de la MEF(1) del Decodificador, caso  $n=3$ .

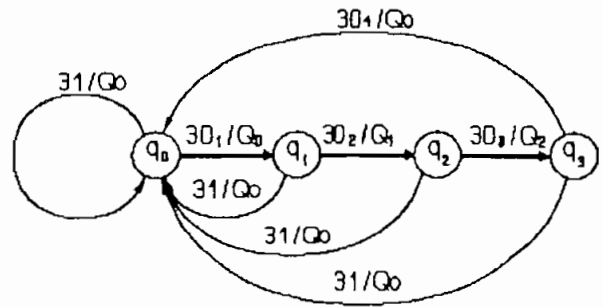


FIGURA 9.37 Diagrama secuencial de la MEF(1) del Decodificador, caso  $n=3$ .

ii)

CASO  $n=2$

n	ESTADO $q_i$	ENTRADA $I_1$	ESTADO $q_{i+1}$	ESTADO Q Salida $E_1$
2	$q_0$	0	$q_1$	$0_1$
2	$q_0$	1	$q_0$	$0_0$
2	$q_1$	0	$q_2$	$0_2$
2	$q_1$	1	$q_0$	$0_0$
2	$q_2$	0	$q_0$	$0_0$
2	$q_2$	1	$q_0$	$0_0$

TABLA 9.23 Transición de estados de la MEF(1) del Decodificador, caso  $n=2$ .

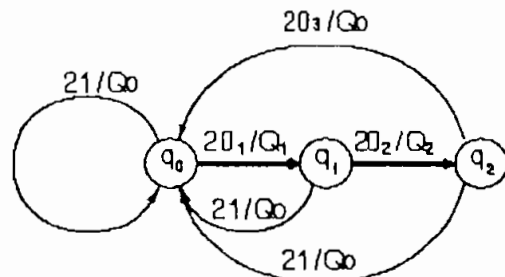


FIGURA 9.38 Diagrama secuencial de la MEF(1) del Decodificador, caso  $n=2$ .

iii)

CASO  $n=1$

n	ESTADO $q_i$	ENTRADA $I_i$	ESTADO $q_{i+1}$	ESTADO Q Salida $E_i$
1	$q_0$	0	$q_1$	$Q_2$
1	$q_0$	1	$q_0$	$Q_1$
1	$q_1$	0	$q_0$	$Q_1$
1	$q_1$	1	$q_0$	$Q_1$

TABLA 9.24 Transición de estados de la MEF(1) del Decodificador, caso  $n=1$ .

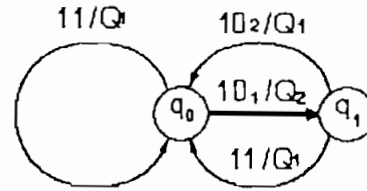


FIGURA 9.39 Diagrama secuencial de la MEF(1) del Decodificador, caso  $n=1$ .

iv)

CASO  $n=0$

n	ESTADO $q_i$	ENTRADA $I_i$	ESTADO $q_{i+1}$	ESTADO Q Salida $E_i$
0	$q_0$	0	$q_0$	$Q_0$
0	$q_0$	1	$q_0$	$Q_0$

TABLA 9.25 Transición de estados de la MEF(1) del Decodificador, caso  $n=0$ .

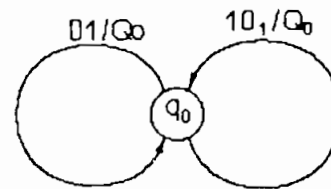


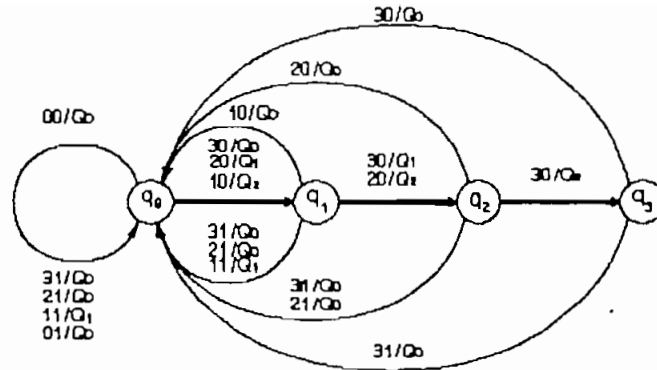
FIGURA 9.40 Diagrama secuencial de la MEF(1) del Decodificador, caso  $n=0$ .

Para este último caso, los estados  $Q_0$ ,  $Q_1$  y  $Q_2$  que resumen a los estados  $q$  de 1 MEF(1) se confunden en un estado  $Q$  único que ha sido nominado como  $Q_0$ .

#### v) Diagrama y tablas de estados generales

El diagrama completo de estados de la MEF(1) del decodificador HDBn se construye agrupando los diagramas secuenciales de las Figuras 9.37 a 9.30, como se indica en la Fig. 9.41.





NOVENCLATURA  $n_i/E_i$

FIGURA 9.41 Diagrama de estados de la MEF(1) del Decodificador.

A fin de interpretar en forma binaria los elementos empleados en el decodificador se ha adoptado la nomenclatura binaria indicada en la Tabla 9.26

ELEMENTOS DE MEF(1)	VALORES POSIBLES	# DE BITS NECESARIOS	NOTACION
ESTADOS Q	Q <sub>0</sub> Q <sub>1</sub> Q <sub>3</sub>	2	00 01 10
ESTADOS q	q <sub>0</sub> q <sub>1</sub> q <sub>2</sub> q <sub>3</sub>	2	00 01 10 11
ENTRADA n	n=0 n=1 n=2 n=3	2	00 01 10 11

TABLA 9.26 Codificación binaria de los estados de la MEF(1) del Decodificador HDBn.

La Tabla de Estados General del Decodificador correspondiente con el diagrama de la Fig.9.41 se indica en la Tabla

9.27 que ha sido construida agrupando las Tablas 9.22 a 9.25. La Tabla 9.28 contiene la misma información en forma binaria según la nomenclatura de la Tabla 9.26.

n	ESTADO $q_i$	ENTRADA $I_1$	ESTADO $q_{i+1}$	ESTADO Q Salida $E_1$
0	$q_0$	0	$q_0$	$Q_0$
0	$q_0$	1	$q_0$	$Q_0$
1	$q_0$	0	$q_1$	$Q_2$
1	$q_0$	1	$q_0$	$Q_1$
1	$q_1$	0	$q_0$	$Q_1$
1	$q_1$	1	$q_0$	$Q_1$
2	$q_0$	0	$q_1$	$Q_1$
2	$q_0$	1	$q_0$	$Q_0$
2	$q_1$	0	$q_2$	$Q_2$
2	$q_1$	1	$q_0$	$Q_0$
2	$q_2$	0	$q_0$	$Q_0$
2	$q_2$	1	$q_0$	$Q_0$
3	$q_0$	0	$q_1$	$Q_0$
3	$q_0$	1	$q_0$	$Q_0$
3	$q_1$	0	$q_2$	$Q_1$
3	$q_1$	1	$q_0$	$Q_0$
3	$q_2$	0	$q_3$	$Q_2$
3	$q_2$	1	$q_0$	$Q_0$
3	$q_3$	0	$q_0$	$Q_0$
3	$q_3$	1	$q_0$	$Q_0$

TABLA 9.27 Transición de estados de la MEF(1) del Decodificador HDBn.

n	ESTADO $q_i$	ENTRADA $I_1$	ESTADO $q_{i+1}$	ESTADO Q Salida $E_1$
$A_4A_3$	$A_2A_1$	$A_0$	$D_3D_2$	$D_1D_0$
0 0	0 0	0	0 0	0 0
0 0	0 0	1	0 0	0 0
0 1	0 0	0	0 1	1 0
0 1	0 0	1	0 0	0 1
0 1	0 1	0	0 0	0 1
0 1	0 1	1	0 0	0 1
1 0	0 0	0	0 1	0 1
1 0	0 0	1	0 0	0 0
1 0	0 1	0	1 0	1 0
1 0	0 1	1	0 0	0 0
1 0	1 0	0	0 0	0 0
1 0	1 0	1	0 0	0 0
1 1	0 0	0	0 1	0 0
1 1	0 0	1	0 0	0 0
1 1	0 1	0	1 0	0 1
1 1	0 1	1	0 0	0 0
1 1	1 0	0	1 1	1 0
1 1	1 0	1	0 0	0 0
1 1	1 1	0	0 0	0 0
1 1	1 1	1	0 0	0 0

TABLA 9.28 Transición de estados de la MEF(1) del Decodificador HDBn (Codificación Binaria).

b) Estructura esquemática

Considerando la Tabla 9.28 se ha deducido el esquema general de la máquina de estados finitos MEF(1) del Decodificador que se indica en la Fig.9.42.

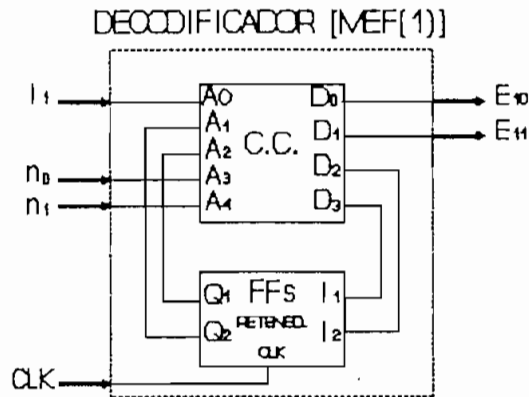


FIGURA 9.42 Esquema de la MEF(1) del Decodificador HDBn.

c) Mapas de Karnaugh

Las salidas  $D_3 \dots D_0$  del circuito combinacional CC de la Fig.9.42 deben adoptar valores lógicos frente a las combinaciones de los valores de entrada  $A_4 \dots A_0$  de acuerdo con la Tabla 9.28. A fin construir los Mapas de Karnaugh que permitan deducir las ecuaciones lógicas que rijan el comportamiento del C.C., la Tabla 9.28 se ha replanteado en la Tabla 9.29 considerando todas las combinaciones lógicas de las variables de entrada posibles. Así:

ENTRADAS	SALIDAS
$A_4A_3A_2A_1A_0$	$D_3D_2D_1D_0$
0 0 0 0 0	0 0 0 0
0 0 0 0 1	0 0 0 0
0 0 0 1 0	X X X X
0 0 0 1 1	X X X X
0 0 1 0 0	X X X X
0 0 1 0 1	X X X X
0 0 1 1 0	X X X X
0 0 1 1 1	X X X X
0 1 0 0 0	0 1 1 0
0 1 0 0 1	0 0 0 1
0 1 0 1 0	0 0 0 1
0 1 0 1 1	0 0 0 1
0 1 1 0 0	X X X X
0 1 1 0 1	X X X X
0 1 1 1 0	X X X X
0 1 1 1 1	X X X X
1 0 0 0 0	0 1 0 1
1 0 0 0 1	0 0 0 0
1 0 0 1 0	1 0 1 0
1 0 0 1 1	0 0 0 0
1 0 1 0 0	0 0 0 0
1 0 1 0 1	0 0 0 0
1 0 1 1 0	X X X X
1 0 1 1 1	X X X X
1 1 0 0 0	0 1 0 0
1 1 0 0 1	0 0 0 0
1 1 0 1 0	1 0 0 1
1 1 0 1 1	0 0 0 0
1 1 1 0 0	1 1 1 0
1 1 1 0 1	0 0 0 0
1 1 1 1 0	0 0 0 0
1 1 1 1 1	0 0 0 0

TABLA 9.29 Transición de estados de la MEF(1) del Decodificador (incluidas condiciones no importa).

i) Salida:  $D_3$

$A_3A_2$		$A_1A_0$				$A_3A_2$		$A_1A_0$			
		00	01	11	10			00	01	11	10
$A_3A_2$	00	0	X	X	0	$A_3A_2$	00	0	0	1	0
	01	0	X	X	0		01	0	0	0	0
	11	X	X	X	0		11	0	X	0	0
	10	X	X	X	0		10	1	X	0	1

$A_4=0$                        $A_4=1$

$$D_3 = \bar{A}_3\bar{A}_1\bar{A}_0 + A_4A_3\bar{A}_2\bar{A}_1\bar{A}_0 + A_3A_2\bar{A}_1\bar{A}_0$$

i) Salida:  $D_2$

$A_3A_2$		$A_1A_0$				$A_3A_2$		$A_1A_0$			
		00	01	11	10			00	01	11	10
$A_3A_2$	00	0	X	X	1	$A_3A_2$	00	1	0	1	1
	01	0	X	X	0		01	0	0	0	0
	11	X	X	X	0		11	0	X	0	0
	10	X	X	X	0		10	0	X	0	0

$A_4=0$                        $A_4=1$

$$D_2 = A_3\bar{A}_1\bar{A}_0 + A_4\bar{A}_3\bar{A}_2\bar{A}_1\bar{A}_0$$

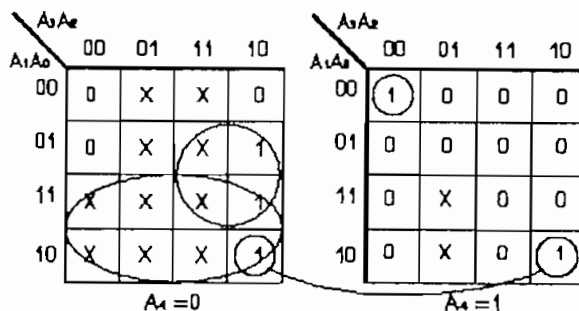
iii) Salida:  $D_1$

$A_3A_2$		$A_1A_0$				$A_3A_2$		$A_1A_0$			
		00	01	11	10			00	01	11	10
$A_3A_2$	00	0	X	X	1	$A_3A_2$	00	0	0	1	0
	01	0	X	X	0		01	0	0	0	0
	11	X	X	X	0		11	0	X	0	0
	10	X	X	X	0		10	1	X	0	0

$A_4=0$                        $A_4=1$

$$D_1 = \bar{A}_4\bar{A}_3\bar{A}_1\bar{A}_0 + \bar{A}_3\bar{A}_1\bar{A}_0 + A_3A_2\bar{A}_1\bar{A}_0$$

iv) Salida:  $D_0$



$$D_0 = \bar{A}_4 A_1 + \bar{A}_4 A_3 A_0 + A_3 \bar{A}_2 A_1 \bar{A}_0 + A_4 \bar{A}_3 \bar{A}_2 \bar{A}_1 \bar{A}_0$$

Los espacios comunes han sido seleccionados con el criterio de tener la mayor cantidad grupos de variables coincidentes entre los mapas de las salidas  $D_2 \dots D_0$  lo que permitirá agrupar compuertas lógicas en el planteamiento lógico del circuito combinacional.

### 9.3.3 Diseño de la MEF(2)

#### a) Diagramas y tablas de estados

La MEF(2) decodifica los códigos "+" y/o "-" en su entrada que pueden ser "B"(alternados) ó "V"(no alternados) como "1" en la salida y los códigos "0" como "0", introduciendo modificaciones en este esquema de decodificación de ser necesario.

La MEF(2) puede pasar por dos estados, según el signo del código que recibe, sea este "8" ó "V".

Signo "positivo": Estado "+"

Signo "negativo": Estado "-"

La estructura deducida para la MEF(2) en la Fig.9.33 reproduce en la Fig.9.43:



FIGURA 9.43 Módulo funcional de la MEF(2) del Decodificador HDBn.

Donde:

$E_1$ : Entrada por la que la MEF(1) reporta su estado  $Q$ .

$E_2$ : Entrada de los símbolos a decodificarse.

$S_1$ : Salida de decodificación inmediata "a priori"

$S_2$ : Salida de modificación de los bits decodificados anteriormente.

La MEF(2) trabaja en función de la MEF(1) dependiendo del estado  $Q$  que esta reporta a través de  $E_1$ , pudiendo presentarse los siguientes casos:

i) Estado de la MEF(1):  $Q_0$

ii) Estado de la MEF(1):  $Q_1$

iii) Estado de la MEF(1):  $Q_2$

iv) Secuencias erróneas.

Dentro del literal (iv) han sido consideradas un conjun-

to de secuencias que bajo condiciones ideales (codificación sin error) no deberían presentarse a la entrada del decodificador.

i) Estado de la MEF(1):  $Q_0$

En la Tabla 9.21 se ha indicado que la MEF(1) se halla en el estado  $Q_0$  cuando:

# 0s consecutivos detectados  $< (n-1)$

bajo estas condiciones cualquier código "+" ó "-" recibido en  $E_2$  se decodifica en  $S_1$  como "1".  $S_2$  se mantiene en "1" (ver Tabla 9.19) puesto que no se requieren modificaciones a los bits decodificados anteriormente.

La MEF(2) pasa al estado "+" cuando se recibe un código "+" y al estado "-" cuando se recibe un código "-", y no cambia de estado cuando se recibe un código "0" (Fig.9.44).

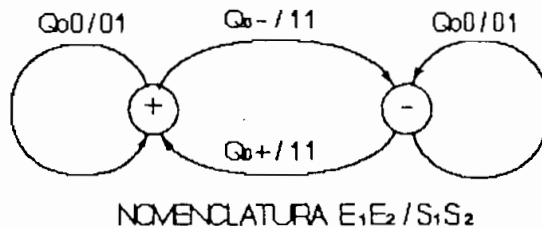


FIGURA 9.44 Diagrama de estados de la MEF(2) para  $E_1 = Q_0$ .

ii) Estado de la MEF(1):  $Q_1$

De acuerdo a la Tabla 9.21, la MEF(1) indica en  $E_1$  que ha llegado al estado  $Q_1$  cuando:

# 0s consecutivos detectados =  $(n-1)$

dependiendo del símbolo (n) que llegue a la MEF(2) pueden producirse los siguientes casos:

ii.1) Caso  $E_2 = 0$

SECUENCIA DE ENTRADA	DECODIFICACION	OBSERVACION
$B_0 \ 0_1 \ 0_2 \ \dots \ 0_{n-1} \ 0_n$	$1_0 \ 0_1 \ 0_2 \ \dots \ 0_{n-1} \ 0_n$	CODIGOS $B_1$ Y $V_{n+1}$ INTRODUCIDOS DURANTE LA CODIFICACION
(+ 0 0 ..... 0 0)		
(- 0 0 ..... 0 0)		

TABLA 9.30 Decodificación de una secuencia de (n-1)0s, caso  $E_1=0_1$ ,  $E_2=0$ .

En este caso en  $E_2$  el decodificador recibe el (n)0 de la secuencia, de modo que existen en total (n)0s consecutivos. En las salidas se tiene:

$S_1=0$  que corresponde a  $0_n$  de la secuencia decodificada.  
 $S_2=1$  puesto que  $B_0$  ya fue decodificado antes como  $1_0$  y no hace falta modificar este valor.

Los estados "+" y "-" se mantienen invariables (Fig.9.45).



FIGURA 9.45 Diagrama de la MEF(2), caso  $E_1=0_1$ ,  $E_2=0$ .



ii.2) Caso  $E_2 = B$

SECUENCIA DE ENTRADA	DECODIFICACION	OBSERVACION
$B_0 \ 0_1 \ 0_2 \ \dots \ 0_{n-1} B_n$ $(+ \ 0 \ 0 \ \dots \ 0 \ -)$ $(- \ 0 \ 0 \ \dots \ 0 \ +)$	$1_0 \ 0_1 \ 0_2 \ \dots \ 0_{n-1} 1_n$	NO EXISTE VIOLACION DE SIGNO EN EL ULTIMO CODIGO "B" RECIBIDO

TABLA 9.31 Decodificación de una secuencia de  $(n-1)0$ s, caso  $E_1=0_1 \ E_2=B$ .

En este caso luego de la secuencia de  $(n-1)0$ s consecutivos se recibe en  $E_2$  un código  $B_n$  cuyo signo es alternado con el  $B_0$  previo a la secuencia de  $0$ s. Sus salidas son:

$S_1=1$  puesto que el código  $B_n$  se decodifica como  $1_n$ .

$S_2=1$  puesto que el código  $B_0$  ya fue decodificado anteriormente como  $1_0$  y no es necesario modificarlo.

En el diagrama de estados los códigos "B" que llegan mantienen la alternabilidad de signos y por tanto de estados en la MEF(2) (Fig.9.46).

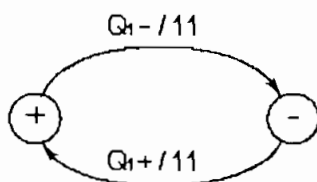


FIGURA 9.46 Diagrama de estados de la MEF(2), caso  $E_1=0_1, E_2=B$ .

ii.3) Caso  $E_2=V$

SECUENCIA DE ENTRADA	DECODIFICACION	OBSERVACION
$B_1 \ 0_2 \ 0_3 \ \dots \ 0_n \ V_{n+1}$ $(+ \ 0 \ 0 \ \dots \ 0 \ +)$ $(- \ 0 \ 0 \ \dots \ 0 \ -)$	$0_1 \ 0_2 \ 0_3 \ \dots \ 0_n 0_{n+1}$	CODIGOS $B_1$ (DE PARIDAD) Y $V_{n+1}$ (DE VIOLACION) TÍPICOS DE CODIFICACION HDBn

TABLA 9.32 Decodificación de una secuencia de  $(n-1)0$ s, caso  $E_1=0_1 \ E_2=V$ .

En este caso luego de los  $(n-1)0$ s consecutivos, la MEF(2) recibe un código "V", se detecta entonces que los códigos  $B_1$  y  $V_{n+1}$  fueron introducidos durante la codificación de una secuencia de  $(n+1)0$ s, por tanto las salidas son:

$S_1=0$  puesto que  $V_{n+1}$  en la entrada corresponde al bit  $0_{n+1}$  de la secuencia original.

$S_2=0$  puesto que la decodificación de  $B_1$  fue originalmente realizada como "1" (caso (i)) y según la Tabla 9.32 debe modificarse a  $0_1$ .

De este modo se recupera la secuencia original de  $(n+1)0$ s consecutivos. El último código  $V_{n+1}$  llega con signo no alternado por lo que el estado de la MEF(2) se conserva (Fig.9.47).



FIGURA 9.47 Diagrama de estados de la MEF(2), caso  $E_1=0, E_2=V$ .

iii) Estado de la MEF(1):  $Q_2$

De acuerdo a la Tabla 9.21, la MEF(1) indica mediante  $E_1$  que ha llegado al estado  $Q_2$  cuando:

$$\# \text{ Os consecutivos detectados} = (n)$$

según el código que llega a  $E_2$  se tienen los casos:

iii.1) Caso  $E_2 = B$

SECUENCIA DE ENTRADA	DECODIFICACION	OBSERVACION
$B_0 \ 0_1 \ 0_2 \ \dots \ 0_n \ B_{n+1}$ $(+ \ 0 \ 0 \ \dots \ 0 \ -)$ $(- \ 0 \ 0 \ \dots \ 0 \ +)$	$1_0 \ 0_1 \ 0_2 \ 0_3 \ \dots \ 0_n \ 1_{n+1}$	NO EXISTE VIOLACION DE SIGNO EN ULTIMO "B" RECIBIDO

TABLA 9.33 Decodificación de una secuencia de (n)0s, caso  $E_1=0_2 \ E_2=B$ .

Es un caso similar al caso (ii.2) con la diferencia de que entre los dos códigos "B" existen (n)0s consecutivos, las salidas son las mismas, y se mantiene la alternabilidad de estados del diagrama de la Fig.9.46 como se indica en la Fig.9.48.

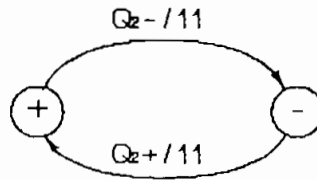


FIGURA 9.48 Diagrama de estados de la MEF(2), caso  $E_1=0_2 \ E_2=B$ .

iii.2) Caso  $E_2=V$

SECUENCIA DE ENTRADA	DECODIFICACION	OBSERVACION
$B_0 \ 0_1 \ 0_2 \ \dots \ 0_n \ V_{n+1}$ $(+ \ 0 \ 0 \ \dots \ 0 \ +)$ $(- \ 0 \ 0 \ \dots \ 0 \ -)$	$1_0 \ 0_1 \ 0_2 \ 0_3 \ \dots \ 0_n \ 0_{n+1}$	NO EXISTE VIOLACION DE SIGNO EN ULTIMO "B" RECIBIDO

TABLA 9.33 Decodificación de una secuencia de (n)0s, caso  $E_1=0_2 \ E_2=V$ .

En este caso, luego de los (n)0s consecutivos llega el signo de violación  $V_{n+1}$  introducido durante la codificación, este código debe ser interpretado como "0" con lo que se completan los (n+1) 0s consecutivos originales. En las salidas se tiene:

$S_1=0$  ya que  $V_{n+1}$  es decodificado como "0".

$S_2=1$  puesto que  $0_1$  ya fue decodificado como "0" anteriormente y no es necesario modificarlo.

El código "V" no implica una alternabilidad de signos por lo que, al igual que en el diagrama de la Fig.9.47, los estados se conservan (Fig.9.49).



FIGURA 9.49 Diagrama de estados de la MEF(2), caso  $E_1=Q_2$   $E_2=V$ .

#### iv) Secuencias erróneas

Las secuencias estudiadas en los casos (i) a (iii) corresponden a secuencias legales (sin error), no obstante en el diagrama de estados de la MEF(2) existen secuencias que bajo condiciones normales no deberían producirse y que se estudian a continuación.

##### iv.1) Caso 1

Los códigos "V" únicamente se admiten en el caso de que se tenga las secuencias:

$$B \ 0_1 \ 0_2 \ \dots \ 0_k \ V$$

donde  $K=n-1$  ó  $K=n$  como se vió en lo casos (ii.3) e (iii.2) respectivamente.

En el caso:  $K < (n-1)$  no existe decodificación posible puesto que el código "V" se halla en una posición ilegal.

Por ejemplo: En HDB3 ( $n=3$ )

Las secuencias: +000+ ( $K=3$ ) y +000+ ( $K=3$ ) son válidas.

Las secuencias: +0+ ( $K=1$ ) y ++ ( $K=0$ ) son erróneas.

El problema de este caso se ha abordado considerando que el último código "V" llegó alterado, para el ejemplo anterior las posibilidades factibles se indican en la Tabla 9.35.

SECUENCIA ERRONEA	SECUENCIA FACTIBLE	SECUENCIA DECODIFICADA
+0+	(1) +00	100
	(2) +0-	101
-0-	(1) -00	100
	(2) -0+	101

TABLA 9.35 Algunas posibles secuencias erróneas en HDB3.

Que son las formas más simples de interpretar este error, no obstante se pueden realizar análisis más complejos estudiando el origen del error en los canales de transmisión, o mediante algoritmos que analizan estadísticamente el comportamiento de una secuencia antes de indicar un valor de corrección, estos análisis trascienden los propósitos del presente trabajo, e incluso justificarían una estructura secuencial completa únicamente dedicada a la corrección del error.

Para el caso del diseño se ha asumido la posibilidad (2)

de la Tabla 9.35, no sin antes advertir que "de la forma como se aborda el error en la Tabla 9.35 las secuencias factibles (1) y (2) son mutuamente excluyentes y sin mas elementos de juicio asumir la posibilidad (1) resulta tan arbitrario como asumir la posibilidad (2)"

El caso resultante es:

SECUENCIA DE ENTRADA	SECUENCIA CORREGIDA	DECODIFICACION
$B_0 \ 0_1 \ 0_2 \ \dots \ 0_K \ V_{K+1}$ $K \leq (n-1)$	$B_0 \ 0_1 \ 0_2 \ 0_3 \ \dots \ 0_K \ B_{K+1}$	$1 \ 0_1 \ 0_2 \ 0_3 \ \dots \ 0_K \ 1_{K+1}$
$(+ \ 0 \ 0 \ \dots \ 0 \ +)$	$(+ \ 0 \ 0 \ 0 \ \dots \ 0 \ -)$	
$(- \ 0 \ 0 \ \dots \ 0 \ -)$	$(- \ 0 \ 0 \ 0 \ \dots \ 0 \ +)$	

TABLA 9.36 Decodificación de una secuencia errónea, caso 1.

En el diagrama de estados se tiene:

$E_1=Q_0$  puesto que  $K \leq n-1$

$E_2=V_{K+1}$  que puede ser "+" ó "-"

$S_1=1$  ya que  $V_{K+1}$  es decodificado como "1"

$S_2=1$  puesto que no es necesario modificar la decodificación de los símbolos anteriores.

Al asumirse la secuencia corregida de la Tabla 9.36, los estados "+" y "-" alternan (Fig.9.50) aún cuando se ha recibido un código "V" (con violación de signo).

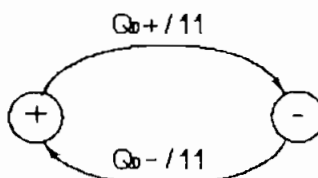


FIGURA 9.50 Diagrama de estados de la MEF(2) para la entrada errónea del caso 1.

iv.2) Caso 2

Cuando la MEF(1) indica mediante  $E_2=Q_2$  que:

$$\# \text{ Os consecutivos detectados} = (n)$$

existe la posibilidad de que llegue otro 0 que sería el  $0_{n+1}$  de la secuencia como se indica en la Tabla 9.37.

SECUENCIA DE ENTRADA	DECODIFICACION	OBSERVACION
$B_0 \ 0_1 \ 0_2 \ \dots \ 0_n \ 0_{n+1}$ $(+ \ 0 \ 0 \ \dots \ 0 \ 0)$ $(- \ 0 \ 0 \ \dots \ 0 \ 0)$	NO ES POSIBLE	EXISTEN (n+1) Os CONSECUTIVOS (** SECUENCIA ERRONEA **)

TABLA 9.37 Secuencia errónea, caso 2: (n+1) Os consecutivos.

Por la definición de HDBn se sabe que la secuencia decodificada puede tener hasta (n) Os consecutivos, por tanto ninguna secuencia correcta puede tener (n+1) Os consecutivos.

Al igual que en el caso anterior, el error ha sido considerado asumiendo que el último 0 llegó distorsionado, sus valores correctos posibles se indican en la Tabla 9.38.

SECUENCIA ERRONEA	SECUENCIAS FACTIBLES	SECUENCIA DECODIFICADA
$B_0 \ 0_1 \ 0_2 \ \dots \ 0_n \ 0_{n+1}$ $(+ \ 0 \ 0 \ \dots \ 0 \ 0)$ $(- \ 0 \ 0 \ \dots \ 0 \ 0)$	(1) $B_0 \ 0_1 \ 0_2 \ 0_3 \ \dots \ 0_n \ B_{n+1}$ $(+ \ 0 \ 0 \ 0 \ \dots \ 0 \ -)$ $(- \ 0 \ 0 \ 0 \ \dots \ 0 \ +)$	$1 \ 0_1 \ 0_2 \ 0_3 \ \dots \ 0_n \ 1_{n+1}$
	(2) $B_0 \ 0_1 \ 0_2 \ 0_3 \ \dots \ 0_n \ V_{n+1}$ $(+ \ 0 \ 0 \ 0 \ \dots \ 0 \ +)$ $(- \ 0 \ 0 \ 0 \ \dots \ 0 \ -)$	$1 \ 0_1 \ 0_2 \ 0_3 \ \dots \ 0_n \ 0_{n+1}$

TABLA 9.38 Decodificación de la secuencia errónea del caso 2.

Con las mismas advertencias del caso 1, se ha optado por la posibilidad (1), es decir que un cuando  $E_2=0$ , la máquina

MEF(2) se comporta como si  $E_2=B$ , con lo que se regresa al caso analizado en (iii.1) (Tabla 9.33), al igual que en este caso los estados de la MEF(2) mantienen alternabilidad Fig.9.51.

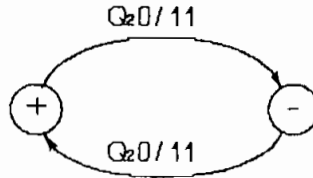


FIGURA 9.51 Diagrama de estados de la MEF(2) para el caso  $E_1=Q_2$   $E_2=0$ .

Resumiendo los diagramas parciales de las Figuras 9.44 a 9.51 se construye el diagrama de estados de la MEF(2) del Decodificador HDBn que se grafica en la Fig.9.52.

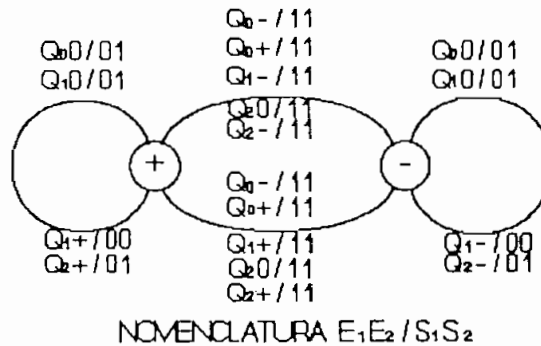


FIGURA 9.52 Diagrama de estados de la MEF(2) del Decodificador HDBn.

Nótese que este diagrama es independiente del valor de "n" por lo que puede usarse para la decodificación de una secuencia HDBn cualquiera.

Este diagrama de estados se lo ha caracterizado mediante la Tabla de Estados 9.39:



ESTADO $q_1$	ENTRADA $E_2$	ESTADO Q DE LA MEF(1) $E_1$	ESTADO $q_{1+1}$	SALIDAS	
				$S_1$	$S_2$
+	0	$Q_0$	+	0	1
+	0	$Q_1$	+	0	1
+	0	$Q_2$	-	1	1
+	+	$Q_0$	-	1	1
+	+	$Q_1$	+	0	0
+	+	$Q_2$	+	0	1
+	-	$Q_0$	-	1	1
+	-	$Q_1$	-	1	1
+	-	$Q_2$	-	1	1
-	0	$Q_0$	-	0	1
-	0	$Q_1$	-	0	1
-	0	$Q_2$	+	1	1
-	+	$Q_0$	+	1	1
-	+	$Q_1$	+	1	1
-	+	$Q_2$	+	1	1
-	-	$Q_0$	+	1	1
-	-	$Q_1$	-	0	0
-	-	$Q_2$	-	0	1

TABLA 9.39 Transición de estados en la MEF(2) del Decodificador HDBn.

A fin de codificar en forma binaria los elementos de la Tabla 9.39 se ha adoptado la nomenclatura de la Tabla 9.40.

ELEMENTOS DE MEF(2)	VALORES POSIBLES	# DE BITS DE NOTACION	NOTACION
ESTADO DE SIGNO	+	1	0
	-		1
ESTADOS Q	$Q_0$	2	00
	$Q_1$		01
	$Q_2$		10
ENTRADAS	0	2	00
	+		01
	-		10

TABLA 9.40 Codificación binaria de los estados de la MEF(2) del Decodificador HDBn.

Conforme a esta nomenclatura se ha replanteado la Tabla de Estados 9.39, obteniéndose la Tabla 9.41:

ESTADO $q_1$	ENTRADA $E_2$	ESTADO Q DE LA MEF(1) $E_1$	ESTADO $q_{1+1}$	SALIDAS	
				$S_1$	$S_2$
$A_4$	$A_3A_2$	$A_1A_0$	$D_2$	$D_1$	$D_0$
0	0 0	0 0	0	0	1
0	0 0	0 1	0	0	1
0	0 0	1 0	1	1	1
0	0 1	0 0	1	1	1
0	0 1	0 1	0	0	0
0	0 1	1 0	0	0	1
0	1 0	0 0	1	1	1
0	1 0	0 1	1	1	1
0	1 0	1 0	1	1	1
1	0 0	0 0	1	0	1
1	0 0	0 1	1	0	1
1	0 0	1 0	0	1	1
1	0 1	0 0	0	1	1
1	0 1	0 1	0	1	1
1	0 1	1 0	0	1	1
1	1 0	0 0	0	1	1
1	1 0	0 1	1	0	0
1	1 0	1 0	1	0	1

TABLA 9.41 Transición de estados de la MEF(2) del Decodificador HDBn (codificación binaria).

**b) Estructura esquemática**

En concordancia con el número de entradas y salidas necesarias, indicadas como  $A_4 \dots A_0$  y  $D_2 D_1 D_0$  en la Tabla 9.41 la MEF(2) del decodificador tiene la estructura esquemática de la Fig.9.53.

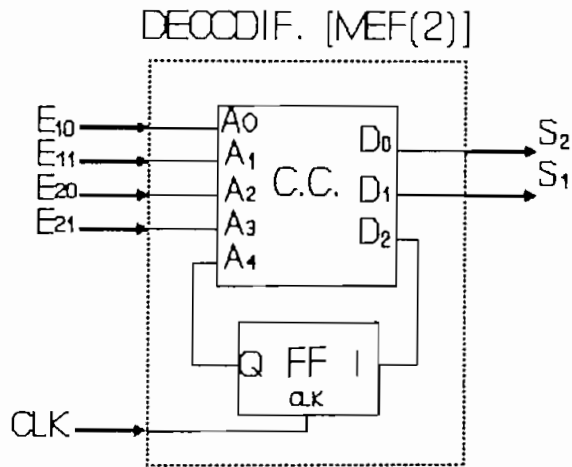


FIGURA 9.53 Estructura esquemática de la MEF(2) del decodificador.

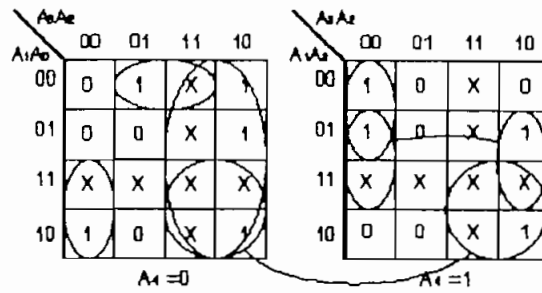
### c) Mapas de Karnaugh

Los valores de las salidas  $D_2D_1D_0$  del CC. de la Fig.9.53 deben ser correspondientes con las combinaciones lógicas de las variables de entrada  $A_4 \dots A_0$  de acuerdo a la Tabla 9.41. Esta Tabla ha sido reformulada en la Tabla 9.42 considerando las combinaciones que las variables de entrada no forman, asignándose a estas los valores de salida XXX (no importa).

ENTRADAS	SALIDAS
$A_4A_3A_2A_1A_0$	$D_2D_1D_0$
0 0 0 0 0	0 0 1
0 0 0 0 1	0 0 1
0 0 0 1 0	1 1 1
0 0 0 1 1	X X X
0 0 1 0 0	1 1 1
0 0 1 0 1	0 0 0
0 0 1 1 0	0 0 1
0 0 1 1 1	X X X
0 1 0 0 0	1 1 1
0 1 0 0 1	1 1 1
0 1 0 1 0	1 1 1
0 1 0 1 1	X X X
0 1 1 0 0	X X X
0 1 1 0 1	X X X
0 1 1 1 0	X X X
0 1 1 1 1	X X X
1 0 0 0 0	1 0 1
1 0 0 0 1	1 0 1
1 0 0 1 0	0 1 1
1 0 0 1 1	X X X
1 0 1 0 0	0 1 1
1 0 1 0 1	0 1 1
1 0 1 1 0	0 1 1
1 0 1 1 1	X X X
1 1 0 0 0	0 1 1
1 1 0 0 1	1 0 0
1 1 0 1 0	1 0 1
1 1 0 1 1	X X X
1 1 1 0 1	X X X
1 1 1 0 0	X X X
1 1 1 1 1	X X X
1 1 1 1 0	X X X

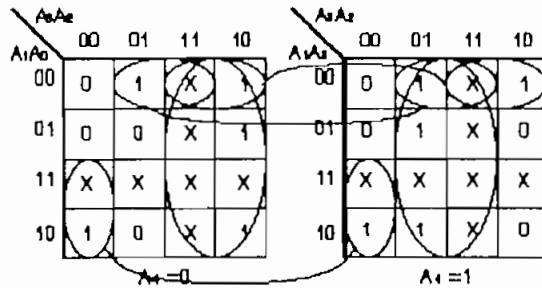
TABLA 9.42 Transición de estados de la MEF(2) del Decodificador (incluidas las condiciones no importa).

i) Salida:  $D_2$



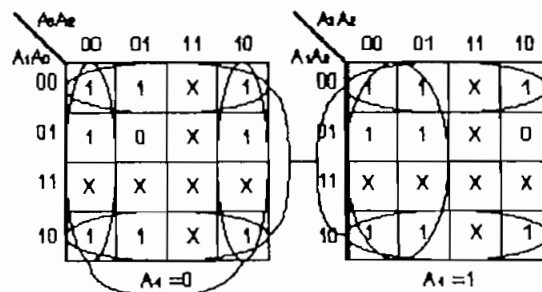
$$D_2 = \bar{A}_4A_3 + \bar{A}_4A_2\bar{A}_1\bar{A}_0 + \bar{A}_4A_3A_2A_1 + A_4\bar{A}_2A_0 + A_3A_1 + A_4\bar{A}_3\bar{A}_2\bar{A}_1$$

ii) Salida:  $D_1$



$$D_1 = \bar{A}_4A_3 + A_4A_2 + \bar{A}_3\bar{A}_2A_1 + A_2\bar{A}_1\bar{A}_0 + A_3\bar{A}_1\bar{A}_0$$

iii) Salida:  $D_0$



$$D_0 = \bar{A}_0 + \bar{A}_4\bar{A}_2 + A_4A_3$$

### 9.3.4 Etapa de salida

La etapa salida del decodificador tiene la estructura indicada en la Fig.9.32 en que los  $(n+1)$  FFs en cascada retienen los últimos  $(n+1)$  bits decodificados en caso de que se necesite modificarlos. Puesto que el diseño ha sido formulado para  $n=3, 2, 1$  ó  $0$ , el número de FFs es variable (Fig.9.54(a)), para ello se ha dispuesto un multiplexor que permita indicar el último FF de la cascada según el valor de "n" seleccionado (Fig.9.54(b)).

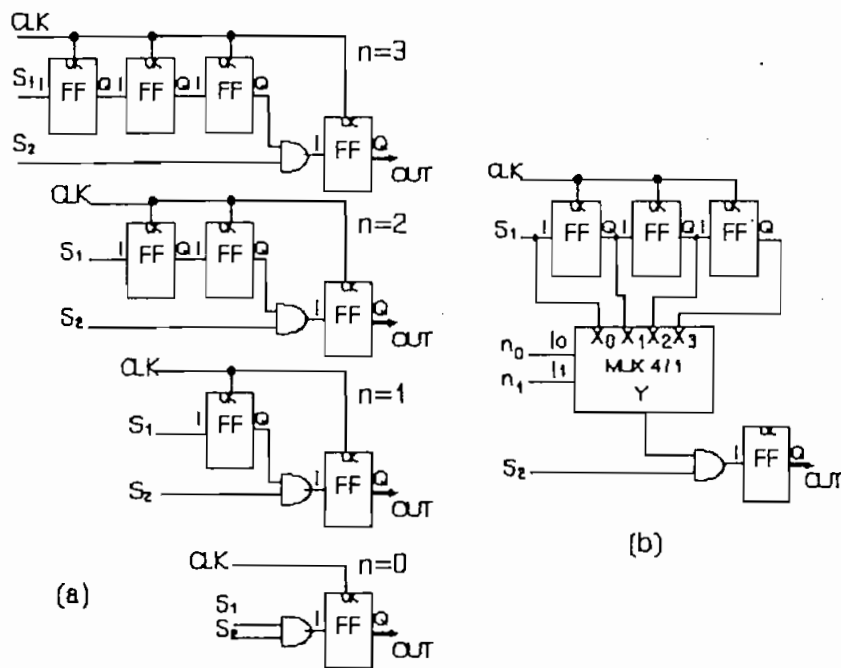


FIGURA 9.54 Salida del decodificador para  $n=3, 2, 1, 0$  (a) y con "n" seleccionable (b).

### 9.3.5 Estructura del Decodificador HDBn

Al interconectarse los módulos de las máquinas MEF(1) de la Fig.9.42 y MEF(2) de la Fig.9.43, y de la etapa de salida de la Fig.9.54, se obtiene la estructura general del decodificador indicada en la Fig.9.55.

De la misma forma que en el Codificador, se han considerado los siguientes detalles para el funcionamiento correcto del decodificador:

- a) Los datos que ingresan a la MEF(2) que son: el resumen de una secuencia reportado por la MEF(1) y el código que sigue a esta secuencia (ambos formados por dos bits en paralelo), deben ser leídos simultáneamente aun cuando lleguen con cierta desviación de tiempo, por ello se han colocado los FFs retenedores FF(a) y FF(b) entre las dos máquinas, y FF(c) y FF(d) en las líneas de los códigos de entrada, todos estos FFs se hallan sincronizados por el reloj externo del decodificador.
  
- b) Al iniciar el funcionamiento del decodificador, sus dos máquinas deben arrancar del estado inicial dado para  $A_4 \dots A_0 = 0$  en las Tablas 9.28 para la MEF(1) y 9.41 para la MEF(2), no obstante sin algún control adicional las máquinas arrancan de un estado incierto, para evitarlo se ha incluido la línea de START de modo que:

Si  $START=0$ , las máquinas se mantienen en su estado inicial.

Si  $START=1$ , las máquinas trabajan normalmente.

De esta manera se garantiza que al arrancar el decodificador (cuando  $START$  cambia a "1"), ambas máquinas comienzan en su estado inicial.

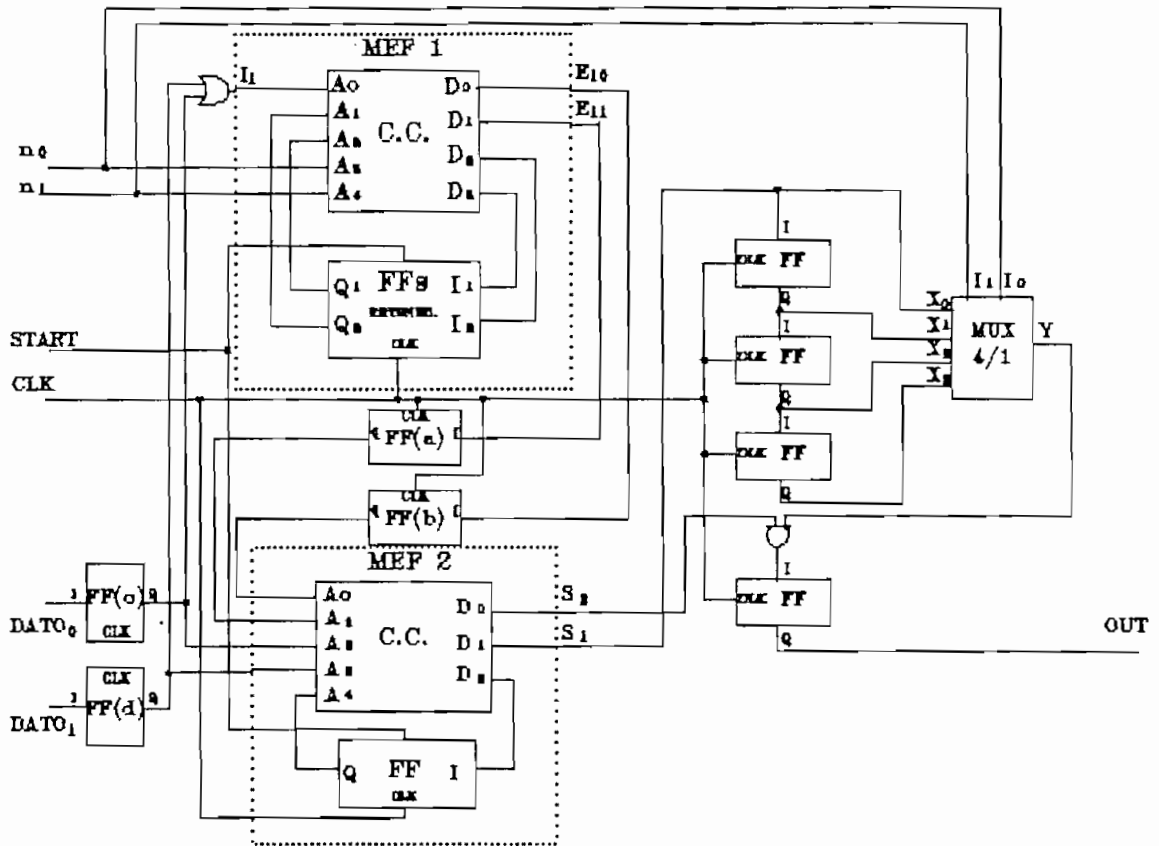


FIGURA 9.55 Esquema general del Decodificador HDBn.

## CAPITULO 10

### DISEÑO DEL CODIFICADOR/DECODIFICADOR HDBn BASADO EN CELDAS ESTANDAR - METODO CONVENCIONAL -

En este capítulo se diseña el circuito CODIFICADOR/DECODIFICADOR HDBn como una aplicación de los principios y metodologías de diseño VLSI introducidos en los capítulos anteriores. Se ha empleado la metodología convencional de diseño basada en celdas estándar para lo que se usa el sistema TENTOS estudiado en el Capítulo 6.

Se enfatiza en los resultados que van siendo obtenidos, a la vez que se omiten los detalles sobre el movimiento del diseñador dentro del entorno de diseño TENTOS que ya fueron analizados con detenimiento en la descripción del diseño del circuito "Medio Sumador" MED\_SUM en el Capítulo 7.



## 10.1 DISEÑO DEL CODIFICADOR PROGRAMABLE HDBn

### 10.1.1 Diseño a nivel funcional

En el Capítulo 9 fueron planteados los algoritmos y deducidas las ecuaciones que caracterizan funcionalmente al circuito CODIFICADOR HDBn y que llevaron a su formulación esquemática de la Fig.9.27, que se reproduce en la Fig.10.1 con ciertas modificaciones de nomenclatura a fin de evitar ambigüedades en la definición de los nodos en los niveles de diseño interiores.

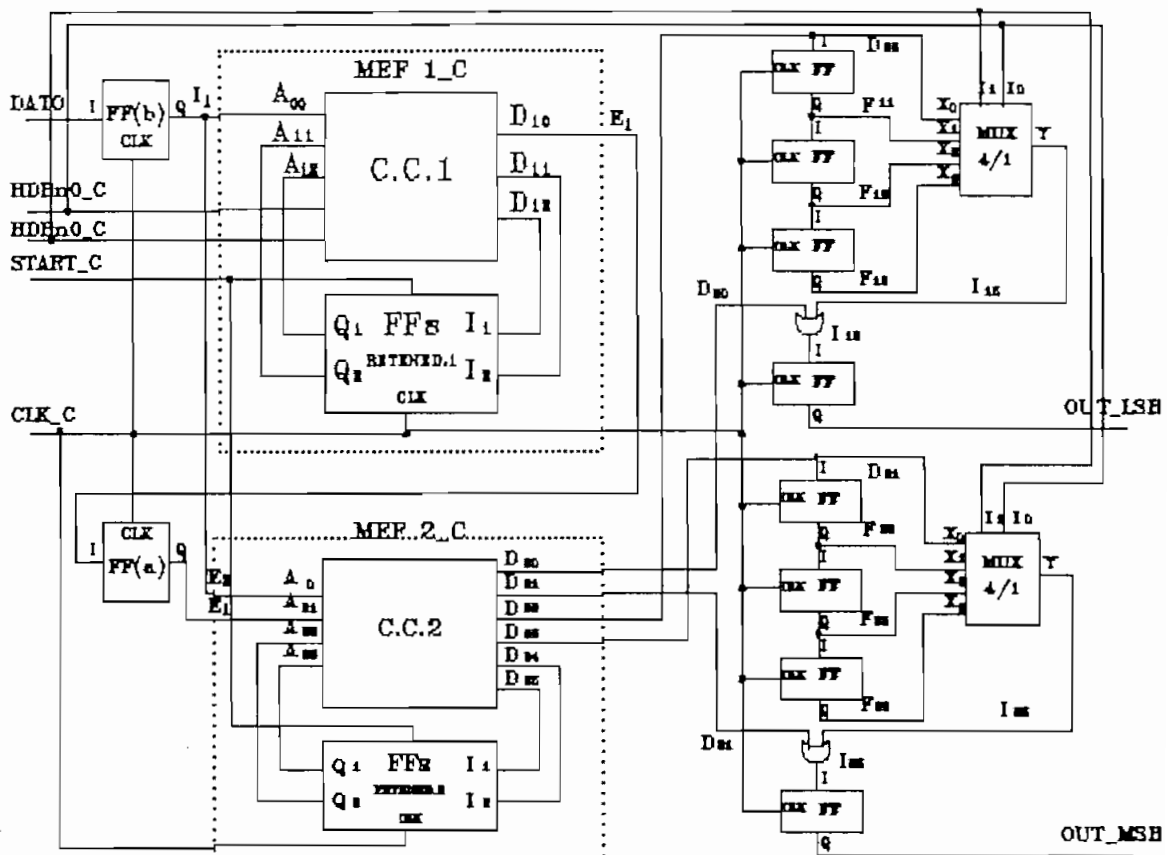


FIGURA 10.1 Esquema funcional del CODIFICADOR HDBn correspondiente el Plano 1.

### 10.1.2 Diseño a nivel lógico

Fuesto que se trata de un diseño en base a celdas estándar, para el diseño a nivel lógico es necesario reformular los módulos funcionales de la Fig.10.1 mediante el uso celdas para lo que se han empleado la biblioteca del sistema TENTOS. (Las celdas empleadas se indican en el Anexo F.1).

#### a) **Formulación del circuito lógico**

El circuito lógico equivalente al esquema de la Fig.10.1 se indica en el Plano 1 (Anexo H), su estructura básica es la siguiente:

##### i) **Circuito Combinacional 1 (CC1)**

En el CC1, los valores que adopten sus salidas (D) deben ser correspondientes con las combinaciones lógicas aplicadas a terminales de entrada (A) de acuerdo con la Tabla 10.1 (reformulada a partir de la Tabla 9.9). Para ello el CC1 ha sido construido en base a las ecuaciones siguientes que fueron deducidas en el numeral 9.2.2.

$$D_{12} = HDBn_1_c \bar{A}_{12} A_{11} \bar{A}_0 + HDBn_0_c A_{12} \bar{A}_{11} \bar{A}_0$$

$$D_{11} = HDBn_0_c \bar{A}_{11} \bar{A}_0 + HDBn_1_c \bar{A}_{12} \bar{A}_{11} \bar{A}_0$$

$$D_{10} = \overline{HDBn_0_c A_{11} \bar{A}_0} + HDBn_0_c A_{12} \bar{A}_{11} \bar{A}_0 + \overline{HDBn_1_c HDBn_0_c \bar{A}_{11} \bar{A}_0}$$

ENTRADAS					SALIDAS		
HDBn <sub>1_c</sub>	HDBn <sub>0_c</sub>	A <sub>12</sub>	A <sub>11</sub>	A <sub>0</sub>	D <sub>12</sub>	D <sub>11</sub>	D <sub>10</sub>
0	0	0	0	0	0	0	0
0	0	0	0	1	0	0	0
0	1	0	0	0	0	1	1
0	1	0	0	1	0	0	0
0	1	0	1	0	0	0	0
0	1	0	1	1	0	0	0
1	0	0	0	0	0	1	0
1	0	0	0	1	0	0	0
1	0	0	1	0	1	0	1
1	0	0	1	1	0	0	0
1	0	1	0	0	0	0	0
1	0	1	0	1	0	0	0
1	1	0	0	0	0	1	0
1	1	0	0	1	0	0	0
1	1	0	1	0	1	0	0
1	1	0	1	1	0	0	0
1	1	1	0	0	1	1	1
1	1	1	0	1	0	0	0
1	1	1	1	0	0	0	0
1	1	1	1	1	0	0	0

TABLA 10.1 Formulación funcional del Circuito Combinacional de la MEF1 del Codificador HDBn.

ii) FFs Retenedores (1)

Este módulo está formado por las celdas FFD\_MS (X15 y X16) que permiten sincronizar y enlazar las líneas de realimentación de estados lógicos al CC1 de la Máquina de Estados Finitos 1 (MEF1\_C). Las celdas AND2 (X5 y X6) a la entrada de los FFs permiten mantener la máquina MEF1\_C en estado inicial "0" mientras START\_C=0.

iii) Circuito Combinacional 2 (CC2)

La relación lógica entre los terminales de salida (D) y de entrada (A) del Circuito Combinacional 2 (CC2) está dada

por la Tabla 10.2 (reformulada a partir de la Tabla 9.17) para la que fueron deducidas las ecuaciones siguientes:

$$D_{25} = A_{23}\bar{A}_{21}\bar{A}_0 + \bar{A}_{23}A_0$$

$$D_{24} = A_{22}\bar{A}_{21}\bar{A}_0 + \bar{A}_{23}\bar{A}_{22}A_1 + A_{23}A_{22}\bar{A}_0 + \bar{A}_{22}A_0$$

$$D_{23} = A_{22}A_0 + \bar{A}_{23}A_{22}A_1 + A_{23}\bar{A}_{22}A_{21}\bar{A}_0$$

$$D_{22} = \bar{A}_{23}\bar{A}_{22}A_{21} + A_{23}A_{22}A_{21}\bar{A}_0 + \bar{A}_{22}A_0$$

$$D_{21} = \bar{A}_{23}A_{22}A_{21}\bar{A}_0$$

$$D_{20} = \bar{A}_{23}\bar{A}_{22}A_{21}\bar{A}_0$$

ENTRADAS				SALIDAS						
A <sub>23</sub> A <sub>22</sub> A <sub>21</sub> A <sub>0</sub>				D <sub>25</sub> D <sub>24</sub> D <sub>23</sub> D <sub>22</sub> D <sub>21</sub> D <sub>20</sub>						
0	0	0	0	0	0	0	0	0	0	0
0	0	0	1	1	1	0	1	0	0	0
0	0	1	0	0	1	0	1	0	1	0
0	0	1	1	1	1	0	1	0	0	0
0	1	0	0	0	1	0	0	0	0	0
0	1	0	1	1	0	1	0	0	0	0
0	1	1	0	0	0	1	0	1	0	0
0	1	1	1	1	1	0	1	0	0	0
1	0	0	0	1	0	0	0	0	0	0
1	0	0	1	0	1	0	1	0	0	0
1	0	1	0	0	0	1	0	0	0	0
1	0	1	1	0	1	0	1	0	0	0
1	1	0	0	1	1	1	0	0	0	0
1	1	0	1	0	0	1	0	0	0	0
1	1	1	0	0	1	0	1	0	0	0
1	1	1	1	0	0	1	0	0	0	0

TABLA 10.2 Formulación funcional del Circuito Combinacional de la MEF2 del Codificador HDBn.

En los terminales de salida de este módulo se han colocado las celdas AND2 (X20, X21 y X28 a X31) a fin de mantener las señales de salida en "0" mientras START\_C=0.

iv) FFS Retenedores 2

Esta formado por las celdas FFD\_MS (X40 y X42) que permiten realimentar los estados lógicos al Circuito Combinacional 2 (CC2) de la Máquina de Estados Finitos 2 (MEF2\_C), no se ha introducido la señal START\_C puesto que esta ya fue considerada en los terminales del módulo CC2.

v) Multiplexores de salida

Debido a la necesidad de variar el número de FFs en cascada en las líneas de salida fueron introducidos los módulos Multiplexores 4/1 en cada línea de salida (Figs 9.25 y 9.26), sin embargo este módulo no existe en la biblioteca del sistema TENTOS por lo que se lo ha implantado en las etapas de salida del Plano 1 en base a celdas elementales de acuerdo con la siguiente tabla funcional:

ENTRADA				SELECCION		SALIDA
X <sub>1</sub>	X <sub>2</sub>	X <sub>3</sub>	X <sub>4</sub>	HDBn <sub>1_c</sub>	HDBn <sub>o_c</sub>	Y
X <sub>1</sub>	.	.	.	0	0	X <sub>1</sub>
.	X <sub>2</sub>	.	.	0	1	X <sub>2</sub>
.	.	X <sub>3</sub>	.	1	0	X <sub>3</sub>
.	.	.	X <sub>4</sub>	1	1	X <sub>4</sub>

TABLA 10.3 Formulación funcional del Multiplexor 4/1.

A la que corresponden las ecuaciones:

$$Y = X_1 \overline{HDBn_{1_c}} \overline{HDBn_{o_c}} + X_2 HDBn_{1_c} \overline{HDBn_{o_c}} + X_3 \overline{HDBn_{1_c}} HDBn_{o_c} + X_4 HDBn_{1_c} HDBn_{o_c}$$

$$Y = (X_1 \overline{HDBn_{1_c}} \overline{HDBn_{o_c}}) \cdot (X_2 HDBn_{1_c} \overline{HDBn_{o_c}}) \cdot (X_3 \overline{HDBn_{1_c}} HDBn_{o_c}) \cdot (X_4 HDBn_{1_c} HDBn_{o_c})$$

Que para el bloque de Salida 1 del Plano 1 se reformula como:

$$Y = (D_{22}HDBn_1_cHDBn_0_c) \cdot (F_{11}HDBn_1_cHDBn_0_c) \cdot (F_{12}HDBn_1_cHDBn_0_c) \cdot (F_{13}HDBn_1_cHDBn_0_c)$$

y para el bloque de Salida 2 se reformula como:

$$Y = (D_{23}HDBn_1_cHDBn_0_c) \cdot (F_{21}HDBn_1_cHDBn_0_c) \cdot (F_{22}HDBn_1_cHDBn_0_c) \cdot (F_{23}HDBn_1_cHDBn_0_c)$$

#### b) Simulación lógica

Para verificar la validez del circuito lógico del Plano 1 se ha extraído la NETLIST NDL de sus módulos principales y confrontado los resultados de su simulación con las tablas funcionales correspondientes. Sería deseable integrar los módulos funcionales y simularlos globalmente, sin embargo ello no ha sido posible debido a las limitaciones en la capacidad de procesamiento del programa NDL.

#### Observaciones:

Respecto a las NETLISTs NDL subsiguientes se deben considerar los siguientes aspectos:

- i) La estructura de nodos es la misma que la indicada en el Plano 1, a excepción del carácter " ~ " que ha sido cambiado por "N" puesto de acuerdo a la sintaxis del lenguaje NDL (Numeral 5.3), el primer carácter de los nodos debe ser siempre una letra.

ii) En las Netlists únicamente se indican las definiciones e invocaciones a subcircuitos, la estructura detallada de cada subcircuito se indica en el Anexo F.1

i) Simulación del Circuito Combinacional 1 (CC1)

De acuerdo al Plano 1, la NETLIST NDL del Circuito Combinacional 1 (CC1) es:

```

-----
%   SIMULACION LOGICA           CCI_C.nd1 ;
%   CODIFICADOR HDBn           ;
%   F.LEMUS /I. BERNAL        ;
%   CIRCUITO COMBINACIONAL 1   ;
-----
FAMILY CMOS;

-----
%   DEFINICION DE SUBCIRCUITOS ;
-----

DEFINE AND2 I1 I2 Out;
,
END;

DEFINE AND3 In1 In2 In3 Out;
,
END;

DEFINE OR2 I1 I2 Out;
,
END;

DEFINE OR3 I1 I2 I3 Out;
,
END;

-----
%   LLAMADA A SUBCIRCUITOS     ;
-----

INPUT  A0 NAO A11 NA11 A12 NA12 HDBn0_c NHDBn0_c HDBn1_c NHDBn1_c;

AND2  A11     P1     P4           ;X(X1)
AND2  P3      A12    P6           ;X(X2)
AND2  NHDBn1_c P3     P5           ;X(X3)
AND2  P1      NA11   P7           ;X(X4)
OR3   P6      P2     P5     D10    ;X(X10)

```

```

AND3  A11    NHDBn0_c  NA0    P2          ;Z(X11)
AND3  NA12   HDBn1_c   NA0    P1          ;Z(X12)
AND3  NA11   HDBn0_c  NA0    P3          ;Z(X13)
OR2   P3     P7       D11          ;Z(X17)
OR2   P4     P6       D12          ;Z(X18)

```

END

Para la simulación de este circuito se han excitado sus terminales de entrada con las combinaciones lógicas indicadas en la Tabla 10.1. El comportamiento de los terminales de salida frente a estas excitaciones se grafica en la Fig.10.2.

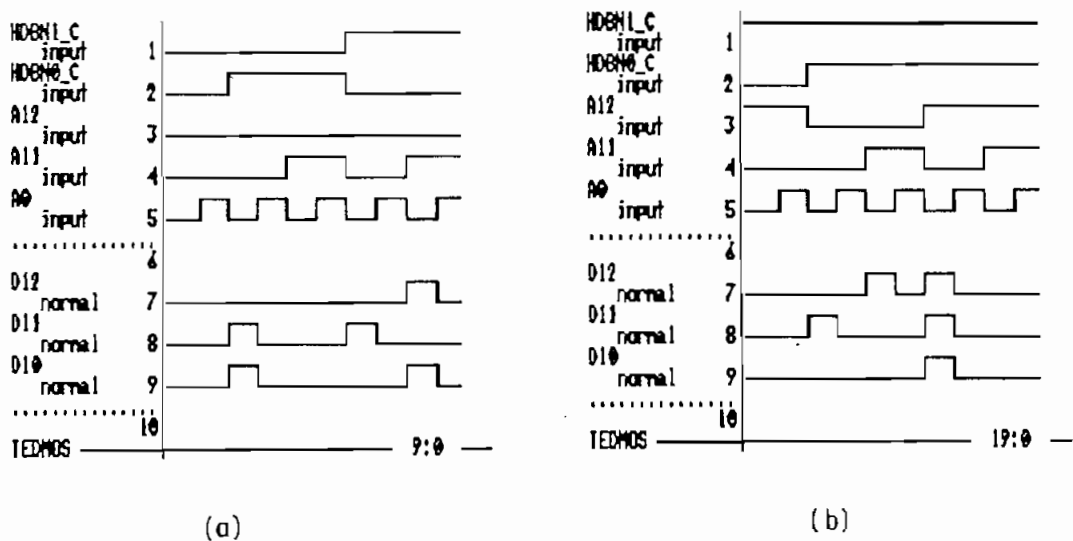


FIGURA 10.2 Simulación lógica del Circuito Combinacional 1 de la MEF1 del codificador HDBn.

Se observa que los resultados de la simulación lógica de la Fig.10.2 concuerdan con aquellos de la formulación funcional de la Tabla 10.1, en consecuencia el circuito lógico propuesto para el módulo CC1 de la MEF1 del Codificador HDBn es correcto.



ii) Simulación del Circuito Combinacional 2 (CC2)

La NETLIST NDL del circuito lógico correspondiente al módulo CC2 del Plano 1 es:

```

Z-----
Z   SIMULACION LOGICA           CC2_C.nd1 ;
Z   CODIFICADOR HD8n           ;
Z   F. LEMUS/ I. BERNAL       ;
Z   CIRCUITO COMBINACIONAL 2   ;
Z-----
FAMILY CMOS;

Z-----
Z  DEFINICION DE SUBCIRCUITOS   ;
Z-----
DEFINE AND2 I1 I2 Out;
,
END;

DEFINE AND3 In1 In2 In3 Out;
,
END;

DEFINE AND4 In1 In2 In3 In4 Out;
,
END;

DEFINE OR2 I1 I2 Out;
,
END;

DEFINE OR3 I1 I2 I3 Out;
,
END;

Z-----
Z   LLAMADA A SUBCIRCUITOS     ;
Z-----

INPUT      A0 NA0 A21 NA21 A22 NA22 A23 NA23 START_c;

AND2  START_c  Q16      D25                ;Z(X20)
AND2  START_c  Q17      D24                ;Z(X21)
AND2  Q4       A21      Q9                 ;Z(X22)
AND2  Q3       NA0     Q10                ;Z(X23)
AND2  NA22    A0       Q8                 ;Z(X24)
AND2  NA0     Q5       Q11                ;Z(X25)
AND2  NA23    A0       Q7                 ;Z(X26)
AND2  A22     A0       Q12                ;Z(X27)
AND2  START_c  Q10     D20                ;Z(X28)
AND2  START_c  Q11     D21                ;Z(X29)
AND2  START_c  Q14     D22                ;Z(X30)
AND2  START_c  Q15     D23                ;Z(X31)

```

```

OR3  Q2    Q4    Q13   Q17           ;Z(X32)
OR3  Q5    Q6    Q12   Q15           ;Z(X33)
AND3 A23   A22   NA0    Q4           ;Z(X34)
AND3 NA23 NA22  A21    Q3           ;Z(X35)
AND3 A22   NA21  NA0    Q2           ;Z(X36)
AND3 A23   NA21  NA0    Q1           ;Z(X37)
AND3 NA23 A22   A21    Q5           ;Z(X38)
AND4 A23   NA22  A21    NA0    Q6           ;Z(X39)
OR2  Q1    Q7    Q16           ;Z(X43)
OR2  Q3    Q8    Q13           ;Z(X44)
OR2  Q13   Q9    Q14           ;Z(X45)

```

END

Las combinaciones lógicas aplicadas a los terminales de entrada de este módulo corresponden a los indicados en la Tabla 10.2, los resultados de esta simulación se exhiben en la Fig.10.3.

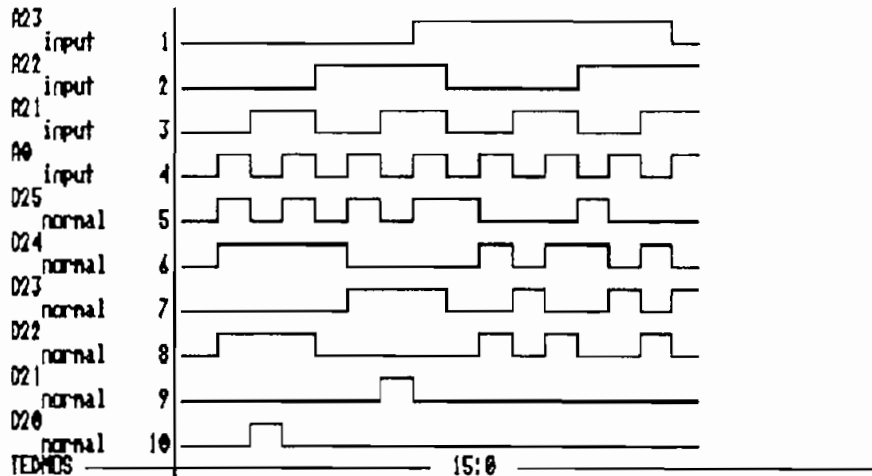


FIGURA 10.3 Simulación lógica del Circuito Combinacional de la MEF2 del Codificador HDBn

Se observa que las señales de salida adoptan valores lógicos coincidentes con los propuestos en la Tabla 10.2 por tanto el circuito lógico del módulo CC2 propuesto en el Plano 1 es correcto.

### iii) Simulación de la etapa de salida

La etapa de salida del circuito lógico del Plano 1, está formada por dos bloques (SALIDA 1 y SALIDA 2) cuya estructura es idéntica, por tanto es suficiente simular uno de ellos para verificar la validez del circuito de salida.

La NETLIST NDL que describe el bloque SALIDA 1 del Codificador HDBn del Plano 1, es la siguiente:

```
X-----
X  SIMULACION LOGICA          OUT1_C.nd1 |
X  CODIFICADOR HDBn          |
X  F. LEMUS/ I. BERNAL      |
X           SALIDA 1         |
X-----
FAMILY CMOS;

X-----
X  DEFINICION DE SUBCIRCUITOS  |
X-----
DEFINE OR2 I1 I2 Out;
,
END;

DEFINE NAND3 I1 I2 I3 Out;
,
END;

DEFINE NAND4 I1 I2 I3 I4 Out;
,
END;

DEFINE FFD_MS D CK CKN QN Q;
,
END;

..

X-----
X  LLAMADA A SUBCIRCUITOS    |
X-----

INPUT    D22 D20 CLK_c NCLK_c HDBn1_c NHDBn1_c HDBn0_c NHDBn0_c;

FFD_MS F12    CLK_c  NCLK_c  NF13    F13    ;Z(X50)
FFD_MS F11    CLK_c  NCLK_c  NF12    F12    ;Z(X51)
FFD_MS D22    CLK_c  NCLK_c  NF11    F11    ;Z(X52)
FFD_MS I16    CLK_c  NCLK_c  NF14    OUT_LSB ;Z(X53)
NAND4 I11     I12     I13     I14     I15    ;I(X54)
```

```

NAND3 NHDBn0_c HDBn1_c F12      I13      ;Z(X57)
NAND3 HDBn0_c NHDBn1_c F11      I12      ;Z(X58)
NAND3 NHDBn0_c NHDBn1_c D22     I11      ;Z(X55)
NAND3 HDBn0_c HDBn1_c F13      I14      ;Z(X56)
OR2   I15      D20      I16      ;Z(X59)

```

END

Para la simulación de este circuito debe aplicarse señales a los siguientes terminales de entrada:

- i) HDBn<sub>1\_c</sub> y HDBn<sub>0\_c</sub>, que mediante el módulo de multiplexación (Multiplexor 4/1) permiten seleccionar el número de celdas FFD\_MS que se colocan en cascada.
- ii) D<sub>22</sub> por el que se aplica la secuencia binaria original, y D<sub>20</sub> por el que se aplica la secuencia de modificación.

La simulación ha sido realizada para los cuatro combinaciones posibles de los terminales de selección HDBn<sub>1\_c</sub> y HDBn<sub>0\_c</sub> como se grafica en la Fig.10.4.

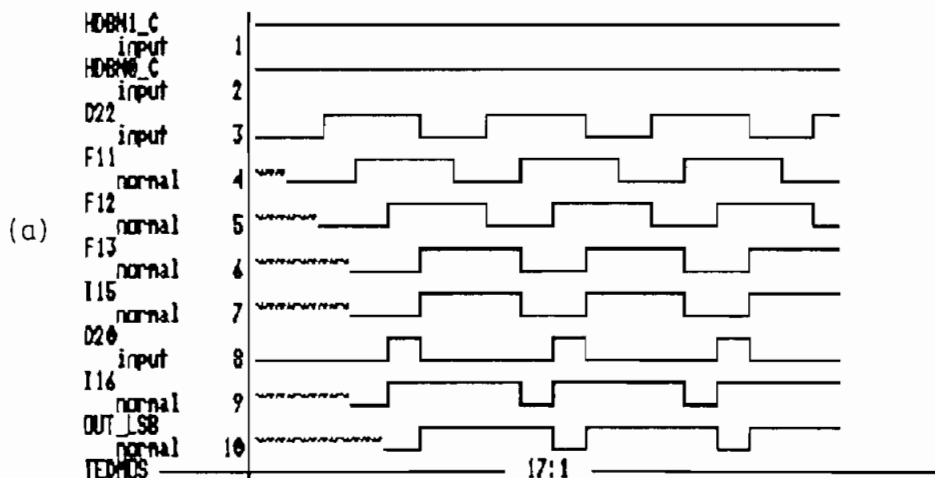


FIGURA 10.4 Simulación del bloque SALIDA 1, (a) caso HDBn<sub>1\_c</sub>=1 HDBn<sub>0\_c</sub>=1,

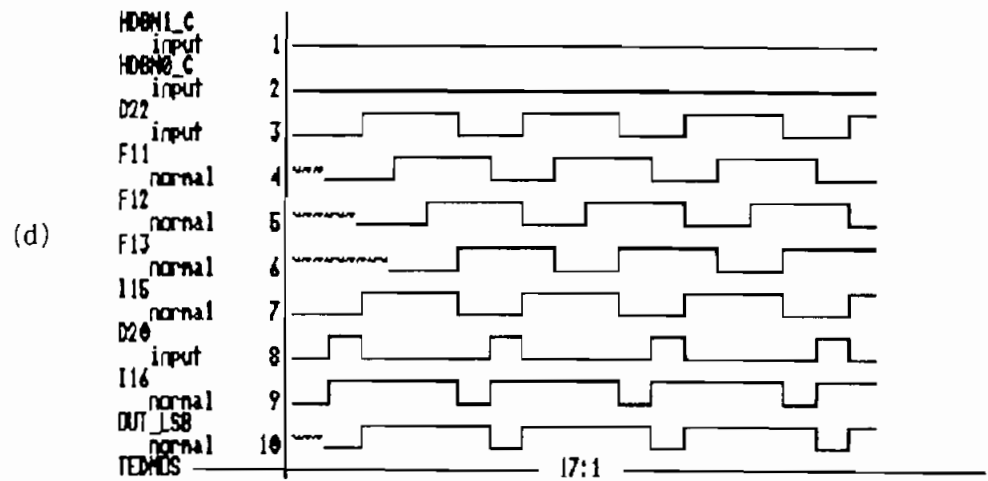
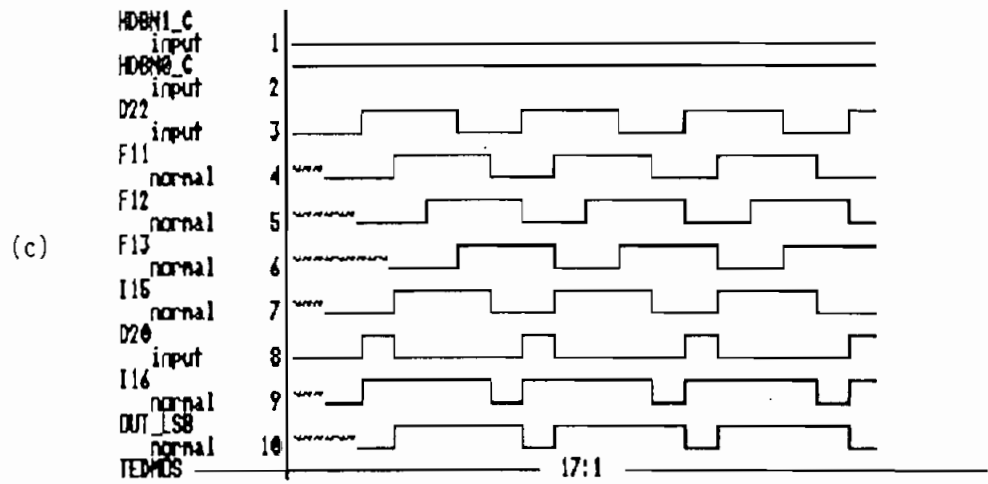
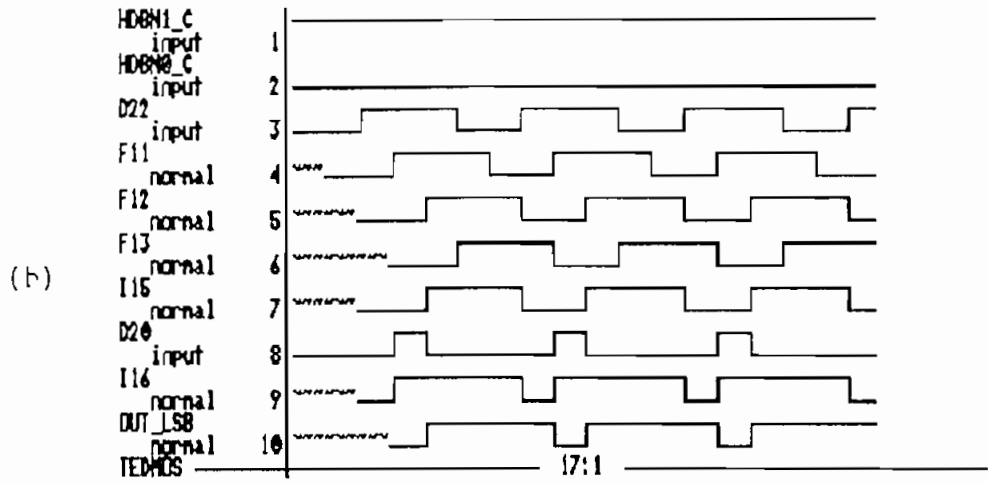


FIGURA 10.4 Cont.. (b) caso  $HDBn_1\_c=1$   $HDBno\_c=0$ , (c) caso  $HDBn_1\_c=0$   $HDBno\_c=1$  y (d) caso  $HDBn_1\_c=0$   $HDBno\_c=0$ .

Los resultados obtenidos para cada caso se indican en la Tabla 10.4:

CASO	HDBn <sub>1_c</sub>	HDBno_c	Salida del Multiplexor	Secuencia Entrante	Secuencia de modificación	Secuencia modificada	Retardo Entrada/Salida
(a)	1	1	$I_{15} = F_{13}$ (3 FFs en cascada)	$D_{22}=0_10_21_31_41_5$	$D_{20}=0_10_20_30_41_5$	$OUT\_LSB=0_11_21_31_41_5$	4 ciclos
(b)	1	0	$I_{15} = F_{12}$ (2 FFs en cascada)	$D_{22}=0_10_21_31_41_5$	$D_{20}=0_10_20_31_40_5$	$OUT\_LSB=0_11_21_31_41_5$	3 ciclos
(c)	0	1	$I_{15} = F_{11}$ (1 FF entre $D_{22}$ e $I_{15}$ )	$D_{22}=0_10_21_31_41_5$	$D_{20}=0_10_21_30_40_5$	$OUT\_LSB=0_11_21_31_41_5$	2 ciclos
(d)	1	1	$I_{15} = D_{22}$ (No hay FFs entre ambas señales)	$D_{22}=0_10_21_31_41_5$	$D_{20}=0_11_20_30_40_5$	$OUT\_LSB=0_11_21_31_41_5$	1 ciclo

TABLA 10.4 Resultados de la simulación del bloque SALIDA 1 del circuito lógico del Plano 1.

En los diagramas de simulación y de la Tabla 10.4 se observa que:

- i) Para los cuatro casos considerados la secuencia periódica aplicada a la entrada es  $D_{22}=0_10_21_31_41_5$  y la secuencia modificada es  $OUT\_LSB=0_11_21_31_41_5$ .
- ii) La secuencia de modificación que se aplica es diferente debido al número de ciclos que debe avanzar  $D_{22}=0_2$  en cada caso para ser modificado en la salida a  $OUT\_LSB=1_2$ .
- iii) Las señales de selección HDBn<sub>1\_c</sub> y HDBno\_c, en efecto permiten seleccionar el número de FFs que se colocan en cascada.

- v) Las salidas de los FFs ( $F_{11}$ ,  $F_{12}$  y  $F_{13}$ ) indican el avance de la secuencia de entrada, la diferencia para cada caso radica en cual de ellas es reproducida a la salida del multiplexor  $I_{15}$ .
- vi) El número de ciclos de retardo entre las señales de entrada/salida es variable debido al número de FFs que se colocan en cascada, nótese que siempre queda a la salida del multiplexor un FF cuya entrada  $I_{16}=D_{20}$  OR  $I_{15}$ .

Se concluye que el circuito lógico de la etapa de salida es correcto puesto que cumple con los requerimientos funcionales para los que fue concebido.

### 10.1.3 Diseño a nivel físico

Idealmente layout del circuito Codificador HDBn debería ser sintetizado "automáticamente" a partir de la descripción en forma de NETLIST SPICE del circuito lógico del Plano 1, sin embargo, los programas de síntesis del sistema TENTOS generan el layout de este circuito en forma "incompleta", por ello se ha optado por dividir al circuito en bloques que permitan construir a partir de síntesis parciales el layout global del circuito Codificador HDBn para luego realizar las interconexiones globales entre estos bloques "manualmente" mediante el editor de máscaras EMA2 a manera de un diseño FULL-CUSTOM.

- a) Definición del plano de base y estrategias de enrutamiento global

Para la determinación de los bloques del plano de base se han considerado los módulos en base a los que se realizó el diseño funcional y que se grafican en el Plano 1.

La distribución de estos módulos y las estrategias de interconexión entre ellos se grafican en la Fig.10.5.

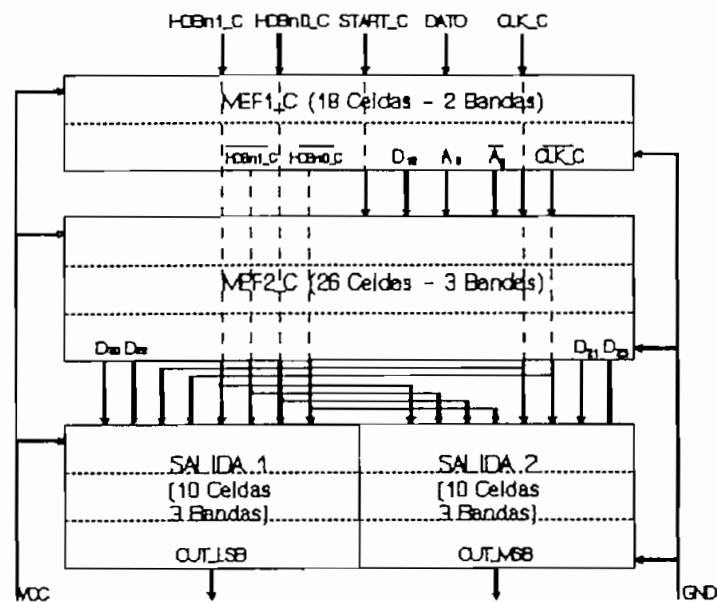


FIGURA 10.5 Plano de base del circuito Codificador HDBn del Plano 1

En el plano de base de la Fig.10.5 se tiene que:

- i) El número de bandas de cada bloque ha sido propuesto de acuerdo al número de celdas que posee, se busca con esta distribución generar bandas de largos similares.



ii) La orientación de los terminales de interconexión deberá ser considerada al asignarse prioridades en la generación del layout de cada bloque.

iii) El plano de base corresponde únicamente al núcleo del layout en la parte que corresponde al Codificador HDBn, mas no al plano de base total del circuito.

NOTA:

En el desarrollo subsiguiente se ha descrito el diseño del bloque MEF1\_C con cierto detalle, se asume que el diseño de los otros bloques tiene características metodológicas similares.

b) Diseño de la Máquina de Estados Finitos 1 (MEF1\_C)

i) Netlist de descripción

En la Netlist que describe el bloque MEF1\_C se debe considerar que los terminales de interfaz (entrada/salida) deben cumplir con las restricciones de orientación indicadas en el plano de base de la Fig.10.5, es decir:

i.1) Las terminales de entrada: HDBn1\_c, HDBn0\_c, DATO, START\_c y CLK\_c deben tener Orientación NORTE.

i.2) Las señales D10, A0, ~A0, ~CLK\_C, ~HDBn1\_c y ~HDBn0\_c, deben tener orientación SUR.

De acuerdo a ello la NETLIST para la síntesis del bloque MEF1\_C es la siguiente:

```

**-----**
** RELATORIO SPICE                MEF1_C.sim **
** CIRCUITO: CODIFICADOR HDBn    **
** BLOQUE: MAQUINA DE ESTADOS FINITOS 1 **
**-----**
X1  A11    P1    P4    vcc    AND2
X2  P3     A12   P6    vcc    AND2
X3  ^HDBn1_c P3    P5    vcc    AND2
X4  P1     ^A11  P7    vcc    AND2
X5  D12    START_c P8    vcc    AND2
X6  D11    START_c P9    vcc    AND2
X7  HDBn0_c ^HDBn0_c vcc    INV
X8  CLK_c  ^CLK_c vcc    INV
X9  HDBn1_c ^HDBn1_c vcc    INV
X10 P6      P2    P5    D10    vcc    OR3
X11 A11    ^HDBn0_c ^A0  P2    vcc    AND3
X12 ^A12   HDBn1_c  ^A0  P1    vcc    AND3
X13 ^A11   HDBn0_c  ^A0  P3    vcc    AND3
X14 DATO  CLK_c    ^CLK_c ^A0  A0    vcc    FFD_MS
X15 P8     CLK_c    ^CLK_c ^A12 A12   vcc    FFD_MS
X16 P9     CLK_c    ^CLK_c ^A11 A11   vcc    FFD_MS
X17 P3     P7     D11    vcc    OR2
X18 P4     P6     D12    vcc    OR2

** SEÑALES DE INTERFAZ DEL CIRCUITO **
** ----- **
† interface: HDBn1_c † orientacao=N †
† interface: HDBn0_c † orientacao=N †
† interface: DATO † orientacao=N †
† interface: START_c † orientacao=N †
† interface: CLK_c † orientacao=N †
† interface: D10 † orientacao=S †
† interface: A0 † orientacao=S †
† interface: ^A0 † orientacao=S †
† interface: ^HDBn1_c † orientacao=S †
† interface: ^HDBn0_c † orientacao=S †
† interface: ^CLK_c † orientacao=S †
.END

```

## ii) Parámetros de particionamiento de celdas y de polarización

Para el particionamiento de las celdas de este bloque se han adoptado los siguientes parámetros:

a: Escolha: Interfaz

A fin de que internamente se generen varias opciones de distribución de celdas y se de preferencia a aquella que resuelva mejor las restricciones de interfaz propuestas.

**b: Número Regioes: 0**

Con el planteamiento modular del plano de base, la distribución en regiones es implícita ya que cada bloque posee sus redes interconexión propias y no existen riesgos de concentración de líneas de enrutamiento en alguna región del layout, en consecuencia no hace falta dividir los bloques en regiones.

**c: Número de Bandas: 2**

De acuerdo con el Plano de Base Fig.10.5, el número de bandas propuesto para este bloque es de 2.

**d: Relacao Aspecto: --**

Al definirse este parámetro se deja al programa en libertad de calcular el número de bandas de acuerdo al que la relación largo/ancho del layout se cumpla, por el contrario si el número de bandas es fijo la relación de aspecto será tan grande cuantas celdas se acumulen en la cantidad de bandas definidas por lo que este parámetro no está al alcance del diseñador.

**e: Percentual Balanco: 100%**

El valor por default de este parámetro es 50%, sin embargo se espera en este caso obtener módulos alargados por lo que es necesario ampliar el margen de variación del largo de las bandas en relación al promedio.

f: Fator de forma: 1.000

El factor de forma permite aumentar el largo de las bandas extremas en proporción a la disminución del largo de las bandas centrales, en el presente caso es preferible que el módulo sea rectangular por lo que este parámetro se ha dejado en su valor por default de "1".

g: Acrescimo Area: 0.000

El incremento del área debido a las inserción de las celdas de interconexión no afecta significativamente a la distribución de las celdas por lo que este factor no ha sido considerado.

De acuerdo al Plano de Base de la Fig.10.5, los parámetros de enrutamiento de polarización adoptados son:

a: Alimentacao Topo: Vcc

b: Alimentacao Esquerda: Vcc

El camino de polarización "Vcc" se orienta en la dirección SUPERIOR-IZQUIERDA, en consecuencia el camino de "Gnd" tendrá orientación INFERIOR-DERECHA.

### iii) Distribución de celdas

De acuerdo a los parámetros indicados, los programas de particionamiento y posicionamiento generan como resultado la organización de celdas que se indica en la Fig.10.6.

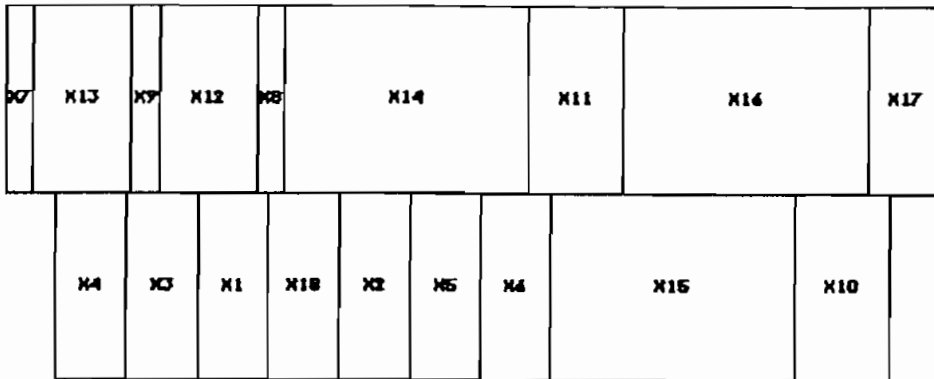


FIGURA 10.6 Distribución de celdas en el bloque MEF1\_C.

La posición final de las celdas dentro de las bandas de la Fig.10.6 es la siguiente:

Posicionamiento do circuito C:\HDBM\MEF1\_C.pos

Banda 1

```

celula= X7 coordenada_simbolica= 0 status rotacional=espelhada
celula= X13 coordenada_simbolica= 21 status rotacional=espelhada
celula= X9 coordenada_simbolica= 94 status rotacional=espelhada
celula= X12 coordenada_simbolica= 115 status rotacional=normal.
celula= X8 coordenada_simbolica= 188 status rotacional=espelhada
celula= X14 coordenada_simbolica= 209 status rotacional=normal
celula= X11 coordenada_simbolica= 392 status rotacional=normal
celula= X16 coordenada_simbolica= 465 status rotacional=normal
celula= X17 coordenada_simbolica= 648 status rotacional=normal

```

Banda 2

```

celula= X4 coordenada_simbolica= 38 status rotacional=espelhada
celula= X3 coordenada_simbolica= 91 status rotacional=espelhada
celula= X1 coordenada_simbolica= 144 status rotacional=espelhada
celula= X18 coordenada_simbolica= 197 status rotacional=espelhada
celula= X2 coordenada_simbolica= 251 status rotacional=espelhada
celula= X5 coordenada_simbolica= 304 status rotacional=espelhada
celula= X6 coordenada_simbolica= 357 status rotacional=normal
celula= X15 coordenada_simbolica= 410 status rotacional=normal
celula= X10 coordenada_simbolica= 593 status rotacional=normal

```

#### iv) Generación del layout

El layout sintetizado automáticamente por el sistema TENTOS se indica en el Plano 2 (a) su descripción en formato RS tiene una estructura jerárquica en que las celdas de biblioteca forman los módulos de menor jerarquía, éstas son integradas en bandas, que a su vez son invocadas para formar el layout global del bloque, como se indica a continuación:

```
DS 1 1 10;
9 "AND_2.CEL";
.
.
DF;
DS 2 1 10;
9 "AND_3.CEL";
.
.
DF;
DS 3 1 10;
9 "FFD_MS.CEL";
.
.
DF;
DS 4 1 10;
9 "INV.CEL";
.
.
DF;
DS 6 1 10;
9 "DR_3.CEL";
.
.
DF;

DS 23 1 10;
9 "BANDA 1";
C 4 T -57 0 MX;
C 2 T -178 0 MX;
C 4 T -223 0 MX;
C 2 T 223 0;
C 4 T -401 0 MX;
C 3 T 401 0;
C 2 T 644 0;
C 3 T 729 0;
C 5 T 924 0;
9 "ENLACES ENTRE CELDAS"
.
.
DF;

DS 24 1 10;
9 "BANDA 2";
C 1 T -158 -106 MX MY;
C 1 T -243 -106 MX MY;
C 1 T -308 -106 MX MY;
C 5 T -434 -106 MX MY;
C 1 T -499 -106 MX MY;
C 1 T -564 -106 MX MY;
C 1 T 620 -106 MY;
C 3 T 685 -106 MY;
C 6 T 880 -106 MY;
9 "ENLACES ENTRE CELDAS"
.
.
DF;

DS 25 1 10;
9 "LAYOUT GLOBAL";
C 23 T 0 100;
C 24
9 "LINEAS DE POLARIZACION";
.
.
DF;
C 25;
E
```

La información general de layout generado es la siguiente:

Roteamento da Banda 1 do Circuito C:\HDBn\MEF1\_C.

Largura Banda=1002

Redes Nao Roteadas:

rede=CLK\_c c\_orig= X14 pino=2 banda=1

Roteamento da Banda 2 do Circuito C:\HDBn\MEF1\_C.

Largura Banda=1002

Redes Nao Roteadas:

rede= D12 c\_vinc= X18 banda=2

rede= P6 c\_vinc= X2 banda=2

rede= P2 c\_vinc= X1 banda=2

rede= D11 c\_orig= X6 pino=2 banda=2

rede= P9 c\_vinc= X6 banda=2

rede=START\_c c\_vinc= X1 banda=2

rede=CLK\_c c\_vinc= X1 banda=2

Informacoes do Modulo Gerado

Area do Circuito = 216300

Relacao de Aspecto = 0.196

En el layout del Plano 2(a) se tiene:

- i) No se han formado todos los enlaces planteados, nótese que en el listado anterior existe el mensaje "REDES NAO ROTEADAS" en el que se advierten los enlaces faltantes en el layout.
  
- ii) Como era de esperarse, este layout presenta los errores de: definición de zonas N y de pasivación, doble definición de capas de Metal1 y Polysilicon, ausencia de zonas activas y formación de transistores falsos que se detallaron para el caso del circuito MED\_SUM (Capítulo 7).

En consecuencia, se deben depurar los caminos de interconexión preexistentes a fin de abrir espacio para realizar los enlaces faltantes, además se deben corregir los errores de generación de layout indicados. Este proceso ha sido realizado en el editor de máscaras EMA2 en forma manual, el layout del bloque MEF1\_C obtenido se indica en el Plano 2(b).

v) Verificación funcional del layout

Para la verificación del layout del bloque MEF1\_C se ha generado el circuito equivalente cuya nominación de nodos y transistores se indican en el Plano 3(a).

La simulación SPICE de este circuito no se ha realizado debido a que la simulación eléctrica únicamente es posible en el caso de circuitos pequeños. En cambio se ha realizado la simulación lógica de acuerdo a la NETLIST NDL correspondiente al layout del Plano 3(a), nótese que en la NETLIST siguiente la numeración de los nodos ha sido precedida por el caracter "X" y se han cambiado los nodos "0" por Gnd y "1" por Vdd para ajustar la NETLIST a la sintaxis del language NDL.

```

Z CIRCUITO: MEF1_C.nd1
Z TECNOLOGIA: ECPD15
Z CIRCUITO CODIFICADOR HD8n
Z MAQUINA DE EST. FINITOS 1
Z
Z Transistores tipo NMOS: 82
Z Transistores tipo PMOS: 82

INPUT x79 x50 x64 x60 x74;
INPUT x62 x72 x48 x84 x61 x86;

NTRANS x89 Gnd x58 ;Z Tr_1
PTRANS x89 Vdd x58 ;Z Tr_2
NTRANS x41 Gnd x89 ;Z Tr_3
NTRANS x56 Gnd x57 ;Z Tr_4
PTRANS x56 Vdd x57 ;Z Tr_5
PTRANS x88 Vdd x39 ;Z Tr_6
PTRANS x41 x39 x89 ;Z Tr_7
NTRANS x88 Gnd x89 ;Z Tr_8
NTRANS x87 Gnd x56 ;Z Tr_9
NTRANS x53 Gnd x56 ;Z Tr_10
PTRANS x53 x38 x37 ;Z Tr_11
PTRANS x87 Vdd x37 ;Z Tr_12
PTRANS x46 x56 x38 ;Z Tr_13
NTRANS x46 Gnd x56 ;Z Tr_14
PTRANS x82 Vdd x36 ;Z Tr_15
NTRANS x82 x61 x35 ;Z Tr_16

NTRANS x85 Gnd x35 ;Z Tr_17
PTRANS x86 x36 x61 ;Z Tr_18
NTRANS x86 Gnd x61 ;Z Tr_19
PTRANS x85 Vdd x36 ;Z Tr_20
NTRANS x61 Gnd x86 ;Z Tr_21
PTRANS x80 Vdd x34 ;Z Tr_22
PTRANS x85 Vdd x31 ;Z Tr_23
NTRANS x80 x33 x48 ;Z Tr_24
NTRANS x85 Gnd x33 ;Z Tr_25
PTRANS x84 x48 x34 ;Z Tr_26
PTRANS x61 x31 x86 ;Z Tr_27
NTRANS x85 Gnd x32 ;Z Tr_28
NTRANS x83 x86 x32 ;Z Tr_29
NTRANS x84 Gnd x48 ;Z Tr_30
PTRANS x85 Vdd x34 ;Z Tr_31
PTRANS x83 Vdd x31 ;Z Tr_32
NTRANS x48 Gnd x84 ;Z Tr_33
PTRANS x78 Vdd x28 ;Z Tr_34
NTRANS x79 Gnd x30 ;Z Tr_35
NTRANS x78 x83 x30 ;Z Tr_36
PTRANS x82 x28 x83 ;Z Tr_37
PTRANS x85 Vdd x27 ;Z Tr_38
NTRANS x82 Gnd x83 ;Z Tr_39
PTRANS x48 x84 x27 ;Z Tr_40
NTRANS x81 x29 x84 ;Z Tr_41
PTRANS x79 Vdd x28 ;Z Tr_42
NTRANS x85 Gnd x29 ;Z Tr_43

PTRANS x81 Vdd x27 ;Z Tr_44
NTRANS x83 Gnd x82 ;Z Tr_45
PTRANS x79 Vdd x24 ;Z Tr_46
PTRANS x52 Vdd x23 ;Z Tr_47
NTRANS x52 x26 x81 ;Z Tr_48
NTRANS x79 Gnd x26 ;Z Tr_49
PTRANS x83 x24 x82 ;Z Tr_50
PTRANS x80 x81 x23 ;Z Tr_51
NTRANS x79 Gnd x25 ;Z Tr_52
NTRANS x55 x82 x25 ;Z Tr_53
NTRANS x80 Gnd x81 ;Z Tr_54
PTRANS x55 Vdd x24 ;Z Tr_55
PTRANS x79 Vdd x23 ;Z Tr_56
NTRANS x78 Gnd x55 ;Z Tr_57
PTRANS x78 Vdd x55 ;Z Tr_58
NTRANS x81 Gnd x80 ;Z Tr_59
PTRANS x79 Vdd x21 ;Z Tr_60
PTRANS x81 x80 x21 ;Z Tr_61
NTRANS x79 Gnd x22 ;Z Tr_62
NTRANS x54 x22 x80 ;Z Tr_63
PTRANS x54 Vdd x21 ;Z Tr_64
NTRANS x77 Gnd x53 ;Z Tr_65
PTRANS x77 Vdd x53 ;Z Tr_66
NTRANS x52 Gnd x54 ;Z Tr_67
PTRANS x52 Vdd x54 ;Z Tr_68
NTRANS x86 x77 x20 ;Z Tr_69
PTRANS x86 Vdd x77 ;Z Tr_70

```



NTRANS	x62	x19	x20	;Z	Tr_71	NTRANS	x50	x69	x11	;Z	Tr_103	PTRANS	x65	Vdd	x67	;Z	Tr_135
NTRANS	x76	Gnd	x78	;Z	Tr_72	NTRANS	x79	Gnd	x11	;Z	Tr_104	NTRANS	x65	Gnd	x67	;Z	Tr_136
PTRANS	x62	Vdd	x77	;Z	Tr_73	PTRANS	x88	Vdd	x68	;Z	Tr_105	NTRANS	x62	x4	x5	;Z	Tr_137
PTRANS	x76	Vdd	x78	;Z	Tr_74	PTRANS	x73	x10	x69	;Z	Tr_106	PTRANS	x62	Vdd	x47	;Z	Tr_138
NTRANS	x75	Gnd	x19	;Z	Tr_75	NTRANS	x73	Gnd	x69	;Z	Tr_107	NTRANS	x64	Gnd	x4	;Z	Tr_139
PTRANS	x75	Vdd	x77	;Z	Tr_76	PTRANS	x79	Vdd	x10	;Z	Tr_108	PTRANS	x64	Vdd	x47	;Z	Tr_140
PTRANS	x58	Vdd	x76	;Z	Tr_77	PTRANS	x68	Vdd	x87	;Z	Tr_109	PTRANS	x45	Vdd	x63	;Z	Tr_141
NTRANS	x58	Gnd	x18	;Z	Tr_78	NTRANS	x68	Gnd	x87	;Z	Tr_110	NTRANS	x45	x3	x63	;Z	Tr_142
NTRANS	x74	x18	x76	;Z	Tr_79	NTRANS	x69	Gnd	x73	;Z	Tr_111	NTRANS	x88	Gnd	x3	;Z	Tr_143
PTRANS	x74	Vdd	x76	;Z	Tr_80	PTRANS	x79	Vdd	x8	;Z	Tr_112	PTRANS	x88	Vdd	x63	;Z	Tr_144
PTRANS	x73	Vdd	x17	;Z	Tr_81	PTRANS	x69	x8	x73	;Z	Tr_113	NTRANS	x64	Gnd	x45	;Z	Tr_145
PTRANS	x72	x17	x62	;Z	Tr_82	NTRANS	x51	x73	x9	;Z	Tr_114	PTRANS	x64	Vdd	x45	;Z	Tr_146
NTRANS	x85	Gnd	x16	;Z	Tr_83	NTRANS	x79	Gnd	x9	;Z	Tr_115	PTRANS	x63	Vdd	x46	;Z	Tr_147
NTRANS	x73	x62	x16	;Z	Tr_84	PTRANS	x51	Vdd	x8	;Z	Tr_116	NTRANS	x63	Gnd	x46	;Z	Tr_148
NTRANS	x72	Gnd	x62	;Z	Tr_85	NTRANS	x87	Gnd	x49	;Z	Tr_117	NTRANS	x62	Gnd	x2	;Z	Tr_149
PTRANS	x85	Vdd	x17	;Z	Tr_86	PTRANS	x67	x49	x7	;Z	Tr_118	PTRANS	x62	Vdd	x44	;Z	Tr_150
PTRANS	x74	Vdd	x70	;Z	Tr_87	NTRANS	x50	Gnd	x51	;Z	Tr_119	NTRANS	x61	x2	x43	;Z	Tr_151
NTRANS	x71	Gnd	x15	;Z	Tr_88	PTRANS	x87	Vdd	x7	;Z	Tr_120	PTRANS	x61	Vdd	x44	;Z	Tr_152
NTRANS	x74	x15	x70	;Z	Tr_89	PTRANS	x50	Vdd	x51	;Z	Tr_121	PTRANS	x66	Vdd	x59	;Z	Tr_153
PTRANS	x71	Vdd	x70	;Z	Tr_90	NTRANS	x67	Gnd	x49	;Z	Tr_122	NTRANS	x61	Gnd	x42	;Z	Tr_154
NTRANS	x62	Gnd	x72	;Z	Tr_91	NTRANS	x79	Gnd	x85	;Z	Tr_123	NTRANS	x66	x42	x59	;Z	Tr_155
PTRANS	x70	Vdd	x52	;Z	Tr_92	NTRANS	x49	Gnd	x71	;Z	Tr_124	PTRANS	x61	Vdd	x59	;Z	Tr_156
PTRANS	x85	Vdd	x13	;Z	Tr_93	PTRANS	x49	Vdd	x71	;Z	Tr_125	NTRANS	x60	x44	x43	;Z	Tr_157
NTRANS	x70	Gnd	x52	;Z	Tr_94	PTRANS	x79	Vdd	x85	;Z	Tr_126	PTRANS	x60	Vdd	x44	;Z	Tr_158
PTRANS	x62	x13	x72	;Z	Tr_95	PTRANS	x86	Vdd	x65	;Z	Tr_127	PTRANS	x59	Vdd	x41	;Z	Tr_159
NTRANS	x85	Gnd	x14	;Z	Tr_96	NTRANS	x66	Gnd	x6	;Z	Tr_128	NTRANS	x44	Gnd	x88	;Z	Tr_160
NTRANS	x69	x72	x14	;Z	Tr_97	NTRANS	x86	x6	x65	;Z	Tr_129	PTRANS	x44	Vdd	x88	;Z	Tr_161
PTRANS	x69	Vdd	x13	;Z	Tr_98	NTRANS	x47	Gnd	x66	;Z	Tr_130	NTRANS	x59	Gnd	x41	;Z	Tr_162
PTRANS	x84	Vdd	x68	;Z	Tr_99	PTRANS	x47	Vdd	x66	;Z	Tr_131	NTRANS	x60	Gnd	x75	;Z	Tr_163
NTRANS	x84	x12	x68	;Z	Tr_100	PTRANS	x66	Vdd	x65	;Z	Tr_132	PTRANS	x60	Vdd	x75	;Z	Tr_164
NTRANS	x88	Gnd	x12	;Z	Tr_101	NTRANS	x48	x47	x5	;Z	Tr_133						
PTRANS	x50	Vdd	x10	;Z	Tr_102	PTRANS	x48	Vdd	x47	;Z	Tr_134						

La simulación de este circuito ha sido realizada con los mismos criterios que se aplicaron en la simulación del circuito lógico a fin de confrontar ambos resultados.

La equivalencia entre la nominación de nodos de terminales del Circuito Combinacional de la MEF1\_C del Plano 1 y su numeración en el Plano 3(a) es la siguiente:

NODOS DEL CIRCUITO LOGICO	NODOS DEL LAYOUT
HDBn1_C (entrada)	x64
HDBnO_C (entrada)	x60
A12 (entrada)	x84
A11 (entrada)	x86
A0 (entrada)	x62
D12 (salida)	x71
D11 (salida)	x58
D10 (salida)	x57

Los nodos de entrada han excitados con las mismas combinaciones lógicas que se aplicaron a los terminales correspondientes del circuito lógico, los resultados obtenidos se indican en la Fig.10.7.

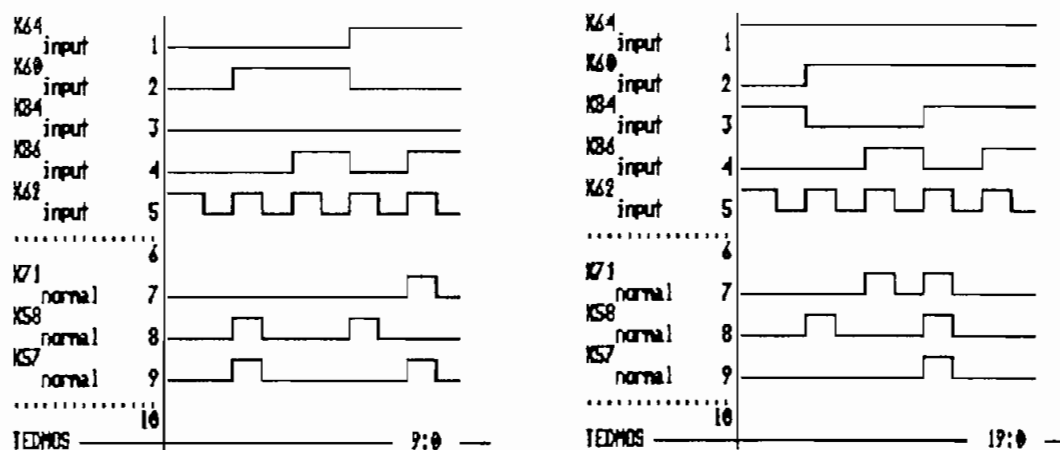


FIGURA 10.7 Simulación lógica del layout del CCI del Plano 3(a)

Los resultados de esta simulación concuerdan con aquellos de la simulación del circuito lógico de la Fig.10.2 y con la descripción funcional de la Tabla 10.1, por lo que se puede concluir que el diseño al nivel funcional es equivalente a los diseños a los niveles superiores: lógico y físico.

c) Diseño la Máquina de Estados Finitos 2 (MEF2\_C)

i) Netlist de descripción

Las restricciones de orientación a que se deben atender los terminales del bloque MEF2\_C de acuerdo con el plano de base de la Fig.10.5, son:

i.1) Las señales de entrada: CLK\_c, ~CLK\_c, START\_c, A0, ~A0 y D10, deben tener orientación NORTE.

i.2) Las señales de salida: D20, D21, D22 y D23 deben tener orientación SUR.

En consecuencia la NETLIST para la síntesis del bloque MEF2\_C es la siguiente:

```

**-----**
** RELATORIO SPICE                MEF2_C.sim **
** CIRCUITO: CODIFICADOR HDBn    **
** BLOQUE: MAQUINA DE ESTADOS FINITOS 2 **
**-----**

X20 START_c Q16    D25    vcc    AND2
X21 START_c Q17    D24    vcc    AND2
X22 Q4      A21    Q9     vcc    AND2
X23 Q3      ~A0    Q10   vcc    AND2
X24 ~A22    A0     Q8     vcc    AND2
X25 ~A0     Q5     Q11   vcc    AND2
X26 ~A23    A0     Q7     vcc    AND2
X27 A22     A0     Q12   vcc    AND2
X28 START_c Q10    D20    vcc    AND2
X29 START_c Q11    D21    vcc    AND2
X30 START_c Q14    D22    vcc    AND2
X31 START_c Q15    D23    vcc    AND2
X32 Q2      Q4     Q13   Q17   vcc    OR3
X33 Q5      Q6     Q12   Q15   vcc    OR3
X34 A23     A22    ~A0   Q4     vcc    AND3
X35 ~A23    ~A22   A21   Q3     vcc    AND3
X36 A22     ~A21   ~A0   Q2     vcc    AND3
X37 A23     ~A21   ~A0   Q1     vcc    AND3
X38 ~A23    A22    A21   Q5     vcc    AND3
X39 A23     ~A22   A21   ~A0   Q6     vcc    AND4
X40 D24     CLK_c  ~CLK_c ~A22   A22    vcc    FFD_MS
```

```

X41 D10    CLK_c    ^CLK_c ^A21    A21    vcc    FFD_MS
X42 D25    CLK_c    ^CLK_c ^A23    A23    vcc    FFD_MS
X43 Q1     Q7       Q16    vcc    DR2
X44 Q3     Q8       Q13    vcc    DR2
X45 Q13    Q9       Q14    vcc    DR2

```

```

## TERMINALES DE INTERFAZ DEL CIRCUITO ##
## ----- ##
# interface: CLK_c    # orientacao=N #
# interface: ^CLK_c  # orientacao=N #
# interface: START_c # orientacao=N #
# interface: A0      # orientacao=N #
# interface: ^A0     # orientacao=N #
# interface: D10     # orientacao=N #
# interface: D20     # orientacao=S #
# interface: D21     # orientacao=S #
# interface: D22     # orientacao=S #
# interface: D23     # orientacao=S #
.END

```

## ii) Parámetros de particionamiento de celdas y de polarización

Los parámetros de particionamiento de celdas y de orientación de polarización son los mismos criterios indicados para el bloque MEF1\_C.

De acuerdo con el plano de base de la Fig.10.5, el único parámetro que varía es:

c: Número de Bandas: 3

debido a que este bloque está formado por mayor cantidad de celdas.

## iii) Distribución de celdas

La distribución de celdas generada por los programas de particionamiento y posicionamiento es la que se indica en la Fig.10.8.

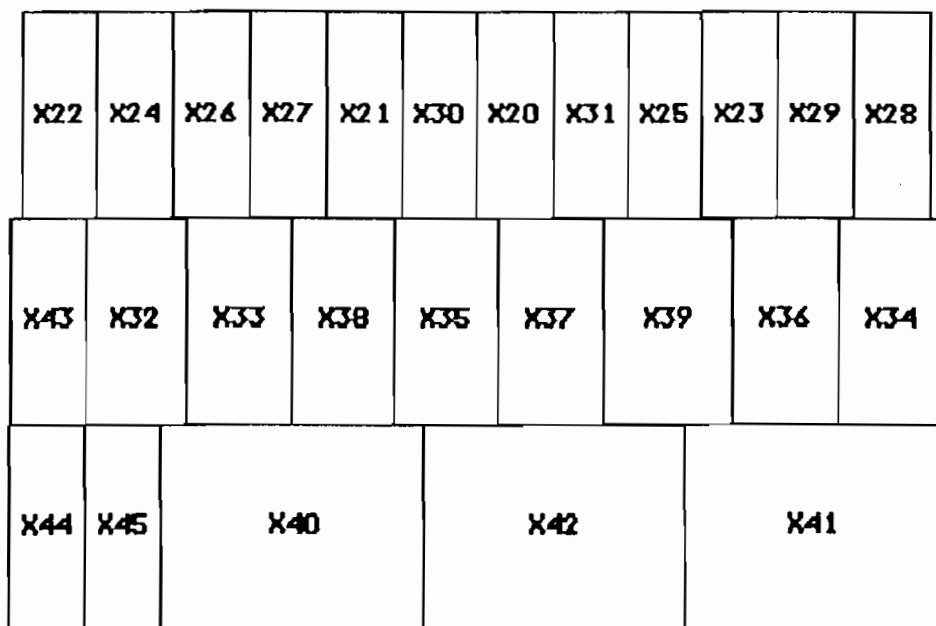


FIGURA 10.8 Distribución de celdas en el bloque MEF2\_C

La posición de las celdas dentro de la distribución de bandas de la Fig.10.8 es la siguiente:

Posicionamiento do circuito C:\HDBN\MEF2\_C.pos

Banda 1

```

celula= X22 coordenada_simbolica= 10 status rotacional=espelhada
celula= X24 coordenada_simbolica= 63 status rotacional=espelhada
celula= X26 coordenada_simbolica= 116 status rotacional=normal
celula= X27 coordenada_simbolica= 169 status rotacional=normal
celula= X21 coordenada_simbolica= 222 status rotacional=espelhada
celula= X30 coordenada_simbolica= 275 status rotacional=espelhada
celula= X20 coordenada_simbolica= 328 status rotacional=espelhada
celula= X31 coordenada_simbolica= 381 status rotacional=espelhada
celula= X25 coordenada_simbolica= 434 status rotacional=espelhada
celula= X23 coordenada_simbolica= 487 status rotacional=espelhada
celula= X29 coordenada_simbolica= 540 status rotacional=normal
celula= X28 coordenada_simbolica= 593 status rotacional=normal

```

Banda 2

```

celula= X43 coordenada_simbolica= 1 status rotacional=espelhada
celula= X32 coordenada_simbolica= 55 status rotacional=espelhada
celula= X33 coordenada_simbolica= 126 status rotacional=espelhada
celula= X38 coordenada_simbolica= 197 status rotacional=espelhada
celula= X35 coordenada_simbolica= 270 status rotacional=espelhada
celula= X37 coordenada_simbolica= 343 status rotacional=espelhada
celula= X39 coordenada_simbolica= 416 status rotacional=espelhada

```

```

celula= X36 coordenada_simbolica= 509 status rotacional=normal
celula= X34 coordenada_simbolica= 582 status rotacional=normal

```

Banda 3

```

celula= X44 coordenada_simbolica= 0 status rotacional=normal
celula= X45 coordenada_simbolica= 54 status rotacional=normal
celula= X40 coordenada_simbolica= 108 status rotacional=normal
celula= X42 coordenada_simbolica= 291 status rotacional=espelhada
celula= X41 coordenada_simbolica= 474 status rotacional=espelhada

```

#### iv) Generación del layout

El layout generado por el sistema TENTOS de acuerdo a la distribución de celdas de la Fig.10.8 se indica en el Plano 2 (b), su estructura jerárquica en formato RS es la siguiente:

```

DS 1 1 10;
9 "AND_2.CEL";
DF;
.
.
DS 2 1 10;
9 "AND_3.CEL";
.
.
DF;
DS 3 1 10;
9 "AND_4.CEL";
.
.
DF;
DS 4 1 10;
9 "FFD_MS.CEL";
.
.
DF;
DS 5 1 10;
9 "DR_2.CEL";
.
.
DF;
DS 6 1 10;
9 "DR_3.CEL";
.
.
DF;

DS 23 1 10;
9 "BANDA 1";
C 1 T -161 0 MX;
C 1 T -287 0 MX;
C 1 T 287 0;
C 1 T 352 0;
C 1 T -597 0 MX;
C 1 T -802 0 MX;
C 1 T -867 0 MX;
C 1 T -932 0 MX;
C 1 T -997 0 MX;
C 1 T -1113 0 MX;
C 1 T 1113 0;
C 1 T 1178 0;
9 "ENLACES ENTRE CELDAS";
.
.
DF;

DS 24 1 10;
9 "BANDA 2";
C 5 T -174 -106 MX MY;
C 6 T -329 -106 MX MY;
C 6 T -412 -106 MX MY;
C 2 T -701 -106 MX MY;
C 2 T -822 -106 MX MY;
C 2 T -907 -106 MX MY;
C 3 T -1012 -106 MX MY;
C 2 T 1060 -106 MY;
C 2 T 1205 -106 MY;
9 "ENLACES ENTRE CELDAS";
.
.
DF;

DS 25 1 10;
9 "BANDA 3";
C 5 T 189 0;
C 5 T 270 0;
C 4 T 336 0;
C 4 T -859 0 MX;
C 4 T -1066 0 MX;
9 "ENLACES ENTRE CELDAS";
.
.
DF;

DS 26 1 10;
9 "LAYOUT GLOBAL";
C 23 T 0 200;
C 24 T 0 100;
C 25;
9 "CAMINOS DE POLARIZACION";
.
.
DF;
C 26;
E

```

Las características generales de las bandas, y las líneas de interconexión no implantadas por el programa de enrutamiento son:

Roteamento da Banda 1 do Circuito C:\HDBn\MEF2\_C.

Largura Banda=1302

Redes Nao Roteadas:

rede= ^A22 c\_orig= X24 pino=2 banda=1

rede= ^A23 c\_orig= X26 pino=2 banda=1

rede= 05 c\_vinc= X20 banda=1

rede= A0 c\_orig= X26 pino=1 banda=1

Roteamento da Banda 2 do Circuito C:\HDBn\MEF2\_C.

Largura Banda=1302

Redes Nao Roteadas:

rede= ^A22 c\_vinc= X32 banda=2

rede= CLK\_c c\_vinc= X32 banda=2

rede= ^CLK\_c c\_vinc= X32 banda=2

rede= D22 c\_vinc= X32 banda=2

rede= ^A0 c\_orig= X37 pino=3 banda=2

rede= D21 c\_vinc= X32 banda=2

Roteamento da Banda 3 do Circuito C:\HDBn\MEF2\_C.

Redes Nao Roteadas:

Informacoes do Modulo Gerado

Area do Circuito = 413100

Relacao de Aspecto = 0.227

El Plano 3 (b) muestra el layout del bloque MEF2\_C en que se han implantado los caminos faltantes (Redes Nao Roteadas) y se han depurado los caminos preexistentes, así como también se han corregido los errores de definición de layout analizados en el diseño del circuito MED\_SUM (Capítulo 7).

#### v) Verificación funcional del layout

La numeración de nodos y de transistores correspondiente al circuito equivalente del layout se indica en el Plano 3 (b). La NETLIST NDL que describe este circuito es la siguiente (se ha antepuesto la letra "y" a la numeración de los nodos y se han cambiado los nodos "1" por "Vdd" y "0" por "Gnd"):

% CIRCUITO: MEF2\_C.nd1  
 % TECNOLOGIA:ECPD15  
 % SUBCIRCUITO CODIFICADOR HD&n  
 %  
 % MAQUINA DE EST. FINITOS 2  
 %  
 % Transistores tipo NMOS: 117  
 % Transistores tipo PMOS: 117

INPUT y125 y64 y113 y112 y73 y105;  
 INPUT y108 y123;

NTRANS y126 Gnd y88 ;% Tr\_1  
 PTRANS y126 Vdd y88 ;% Tr\_2  
 NTRANS y125 y54 y126 ;% Tr\_3  
 PTRANS y125 Vdd y126 ;% Tr\_4  
 NTRANS y105 y54 y53 ;% Tr\_5  
 PTRANS y105 Vdd y126 ;% Tr\_6  
 NTRANS y124 Gnd y85 ;% Tr\_7  
 PTRANS y124 Vdd y85 ;% Tr\_8  
 NTRANS y123 Gnd y53 ;% Tr\_9  
 PTRANS y123 Vdd y126 ;% Tr\_10  
 PTRANS y59 Vdd y124 ;% Tr\_11  
 NTRANS y122 y124 y52 ;% Tr\_12  
 NTRANS y59 Gnd y52 ;% Tr\_13  
 PTRANS y122 Vdd y124 ;% Tr\_14  
 NTRANS y121 Gnd y84 ;% Tr\_15  
 PTRANS y121 Vdd y84 ;% Tr\_16  
 PTRANS y59 Vdd y121 ;% Tr\_17  
 NTRANS y80 y121 y51 ;% Tr\_18  
 NTRANS y59 Gnd y51 ;% Tr\_19  
 PTRANS y80 Vdd y121 ;% Tr\_20  
 NTRANS y120 Gnd y65 ;% Tr\_21  
 PTRANS y120 Vdd y65 ;% Tr\_22  
 NTRANS y125 y49 y120 ;% Tr\_23  
 PTRANS y125 Vdd y120 ;% Tr\_24  
 PTRANS y125 Vdd y119 ;% Tr\_25  
 NTRANS y125 y119 y50 ;% Tr\_26  
 NTRANS y74 Gnd y50 ;% Tr\_27  
 PTRANS y74 Vdd y119 ;% Tr\_28  
 NTRANS y113 y49 y48 ;% Tr\_29  
 PTRANS y113 Vdd y120 ;% Tr\_30  
 NTRANS y105 Gnd y48 ;% Tr\_31  
 PTRANS y105 Vdd y120 ;% Tr\_32  
 PTRANS y119 Vdd y122 ;% Tr\_33  
 NTRANS y119 Gnd y122 ;% Tr\_34  
 PTRANS y118 Vdd y83 ;% Tr\_35  
 NTRANS y118 Gnd y83 ;% Tr\_36  
 PTRANS y83 Vdd y46 ;% Tr\_37  
 NTRANS y83 y116 y47 ;% Tr\_38  
 NTRANS y82 Gnd y47 ;% Tr\_39  
 PTRANS y117 y46 y116 ;% Tr\_40  
 PTRANS y82 Vdd y46 ;% Tr\_41  
 NTRANS y117 Gnd y116 ;% Tr\_42  
 PTRANS y82 Vdd y43 ;% Tr\_43

PTRANS y123 Vdd y79 ;% Tr\_44  
 NTRANS y116 Gnd y117 ;% Tr\_45  
 NTRANS y112 y45 y81 ;% Tr\_46  
 NTRANS y123 Gnd y45 ;% Tr\_47  
 NTRANS y118 y117 y44 ;% Tr\_48  
 PTRANS y116 y43 y117 ;% Tr\_49  
 PTRANS y112 Vdd y79 ;% Tr\_50  
 NTRANS y82 Gnd y44 ;% Tr\_51  
 PTRANS y118 Vdd y43 ;% Tr\_52  
 PTRANS y125 Vdd y114 ;% Tr\_53  
 NTRANS y115 Gnd y42 ;% Tr\_54  
 NTRANS y125 y114 y42 ;% Tr\_55  
 PTRANS y115 Vdd y114 ;% Tr\_56  
 NTRANS y125 y41 y81 ;% Tr\_57  
 PTRANS y125 Vdd y79 ;% Tr\_58  
 PTRANS y117 Vdd y39 ;% Tr\_59  
 PTRANS y114 Vdd y80 ;% Tr\_60  
 NTRANS y117 y112 y40 ;% Tr\_61  
 NTRANS y101 Gnd y40 ;% Tr\_62  
 PTRANS y73 Vdd y79 ;% Tr\_63  
 PTRANS y113 y39 y112 ;% Tr\_64  
 NTRANS y73 y41 y79 ;% Tr\_65  
 NTRANS y114 Gnd y80 ;% Tr\_66  
 PTRANS y101 Vdd y39 ;% Tr\_67  
 NTRANS y113 Gnd y112 ;% Tr\_68  
 NTRANS y79 Gnd y97 ;% Tr\_69  
 PTRANS y79 Vdd y97 ;% Tr\_70  
 PTRANS y59 Vdd y111 ;% Tr\_71  
 NTRANS y112 Gnd y113 ;% Tr\_72  
 NTRANS y59 y111 y38 ;% Tr\_73  
 NTRANS y67 Gnd y38 ;% Tr\_74  
 PTRANS y101 Vdd y36 ;% Tr\_75  
 PTRANS y67 Vdd y111 ;% Tr\_76  
 NTRANS y116 y113 y37 ;% Tr\_77  
 NTRANS y101 Gnd y37 ;% Tr\_78  
 PTRANS y112 y36 y113 ;% Tr\_79  
 PTRANS y116 Vdd y36 ;% Tr\_80  
 NTRANS y113 Gnd y35 ;% Tr\_81  
 PTRANS y113 Vdd y77 ;% Tr\_82  
 PTRANS y111 Vdd y78 ;% Tr\_83  
 NTRANS y111 Gnd y78 ;% Tr\_84  
 PTRANS y125 Vdd y77 ;% Tr\_85  
 NTRANS y125 y34 y35 ;% Tr\_86  
 NTRANS y123 y34 y77 ;% Tr\_87  
 PTRANS y123 Vdd y77 ;% Tr\_88  
 PTRANS y59 Vdd y110 ;% Tr\_89  
 NTRANS y59 y110 y33 ;% Tr\_90  
 NTRANS y63 Gnd y33 ;% Tr\_91  
 PTRANS y63 Vdd y110 ;% Tr\_92  
 PTRANS y72 Vdd y76 ;% Tr\_93  
 NTRANS y72 Gnd y76 ;% Tr\_94  
 NTRANS y77 Gnd y56 ;% Tr\_95  
 PTRANS y77 Vdd y56 ;% Tr\_96  
 PTRANS y76 Vdd y31 ;% Tr\_97  
 NTRANS y82 Gnd y32 ;% Tr\_98

NTRANS y76 y107 y32 ;% Tr\_99  
 PTRANS y109 y31 y107 ;% Tr\_100  
 PTRANS y110 Vdd y72 ;% Tr\_101  
 NTRANS y110 Gnd y72 ;% Tr\_102  
 PTRANS y82 Vdd y31 ;% Tr\_103  
 NTRANS y109 Gnd y107 ;% Tr\_104  
 NTRANS y108 Gnd y30 ;% Tr\_105  
 PTRANS y108 Vdd y75 ;% Tr\_106  
 NTRANS y73 y26 y30 ;% Tr\_107  
 PTRANS y73 Vdd y75 ;% Tr\_108  
 PTRANS y82 Vdd y27 ;% Tr\_109  
 NTRANS y107 Gnd y109 ;% Tr\_110  
 PTRANS y59 Vdd y106 ;% Tr\_111  
 NTRANS y71 Gnd y29 ;% Tr\_112  
 NTRANS y59 y106 y29 ;% Tr\_113  
 NTRANS y72 y109 y28 ;% Tr\_114  
 PTRANS y107 y27 y109 ;% Tr\_115  
 NTRANS y82 Gnd y28 ;% Tr\_116  
 PTRANS y71 Vdd y106 ;% Tr\_117  
 PTRANS y72 Vdd y27 ;% Tr\_118  
 NTRANS y112 y26 y75 ;% Tr\_119  
 PTRANS y112 Vdd y75 ;% Tr\_120  
 PTRANS y106 Vdd y60 ;% Tr\_121  
 NTRANS y75 Gnd y74 ;% Tr\_122  
 PTRANS y75 Vdd y74 ;% Tr\_123  
 NTRANS y106 Gnd y60 ;% Tr\_124  
 PTRANS y109 Vdd y24 ;% Tr\_125  
 NTRANS y109 y123 y25 ;% Tr\_126  
 NTRANS y101 Gnd y25 ;% Tr\_127  
 PTRANS y108 y24 y123 ;% Tr\_128  
 PTRANS y101 Vdd y24 ;% Tr\_129  
 NTRANS y108 Gnd y123 ;% Tr\_130  
 NTRANS y123 Gnd y108 ;% Tr\_131  
 PTRANS y101 Vdd y22 ;% Tr\_132  
 NTRANS y101 Gnd y23 ;% Tr\_133  
 NTRANS y107 y108 y23 ;% Tr\_134  
 PTRANS y123 y22 y108 ;% Tr\_135  
 PTRANS y107 Vdd y22 ;% Tr\_136  
 PTRANS y108 Vdd y104 ;% Tr\_137  
 NTRANS y108 Gnd y21 ;% Tr\_138  
 NTRANS y105 y20 y21 ;% Tr\_139  
 PTRANS y105 Vdd y104 ;% Tr\_140  
 NTRANS y112 y20 y104 ;% Tr\_141  
 PTRANS y112 Vdd y104 ;% Tr\_142  
 NTRANS y104 Gnd y115 ;% Tr\_143  
 PTRANS y104 Vdd y115 ;% Tr\_144  
 PTRANS y59 Vdd y102 ;% Tr\_145  
 NTRANS y59 y102 y19 ;% Tr\_146  
 NTRANS y103 Gnd y19 ;% Tr\_147  
 PTRANS y103 Vdd y102 ;% Tr\_148  
 PTRANS y102 Vdd y70 ;% Tr\_149  
 NTRANS y102 Gnd y70 ;% Tr\_150  
 PTRANS y99 Vdd y18 ;% Tr\_151  
 PTRANS y105 y18 y73 ;% Tr\_152  
 NTRANS y101 Gnd y17 ;% Tr\_153



```

NTRANS y99 y73 y17 ;Z Tr_154
NTRANS y105 Gnd y73 ;Z Tr_155
PTRANS y101 Vdd y18 ;Z Tr_156
NTRANS y73 Gnd y105 ;Z Tr_157
PTRANS y101 Vdd y15 ;Z Tr_158
PTRANS y73 y15 y105 ;Z Tr_159
NTRANS y100 y105 y16 ;Z Tr_160
NTRANS y101 Gnd y16 ;Z Tr_161
PTRANS y100 Vdd y15 ;Z Tr_162
PTRANS y70 Vdd y14 ;Z Tr_163
NTRANS y70 y100 y13 ;Z Tr_164
NTRANS y82 Gnd y13 ;Z Tr_165
PTRANS y99 y14 y100 ;Z Tr_166
NTRANS y99 Gnd y100 ;Z Tr_167
PTRANS y82 Vdd y14 ;Z Tr_168
NTRANS y115 Gnd y68 ;Z Tr_169
NTRANS y98 Gnd y69 ;Z Tr_170
PTRANS y98 Vdd y69 ;Z Tr_171
PTRANS y69 Vdd y11 ;Z Tr_172
PTRANS y97 y12 y11 ;Z Tr_173
PTRANS y115 y68 y12 ;Z Tr_174
NTRANS y97 Gnd y68 ;Z Tr_175
NTRANS y100 Gnd y99 ;Z Tr_176
NTRANS y69 Gnd y68 ;Z Tr_177
PTRANS y82 Vdd y8 ;Z Tr_178
PTRANS y105 Vdd y98 ;Z Tr_179
NTRANS y64 y98 y10 ;Z Tr_180
NTRANS y105 Gnd y10 ;Z Tr_181
PTRANS y100 y8 y99 ;Z Tr_182
NTRANS y82 Gnd y9 ;Z Tr_183
NTRANS y96 y99 y9 ;Z Tr_184
PTRANS y64 Vdd y98 ;Z Tr_185
PTRANS y96 Vdd y8 ;Z Tr_186
NTRANS y70 Gnd y96 ;Z Tr_187
NTRANS y68 Gnd y67 ;Z Tr_188
PTRANS y70 Vdd y96 ;Z Tr_189
PTRANS y68 Vdd y67 ;Z Tr_190
NTRANS y94 Gnd y62 ;Z Tr_191
PTRANS y94 Vdd y62 ;Z Tr_192
NTRANS y66 Gnd y71 ;Z Tr_193
NTRANS y88 Gnd y57 ;Z Tr_194
PTRANS y66 Vdd y71 ;Z Tr_195
PTRANS y65 Vdd y6 ;Z Tr_196
PTRANS y93 y7 y6 ;Z Tr_197
PTRANS y88 y57 y7 ;Z Tr_198
NTRANS y93 Gnd y57 ;Z Tr_199
PTRANS y108 Vdd y94 ;Z Tr_200
NTRANS y64 y94 y5 ;Z Tr_201
NTRANS y108 Gnd y5 ;Z Tr_202
PTRANS y64 Vdd y94 ;Z Tr_203
NTRANS y58 Gnd y66 ;Z Tr_204
NTRANS y65 Gnd y57 ;Z Tr_205
PTRANS y93 Vdd y4 ;Z Tr_206
PTRANS y58 y4 y66 ;Z Tr_207
NTRANS y93 Gnd y66 ;Z Tr_208
PTRANS y64 Vdd y91 ;Z Tr_209
NTRANS y73 Gnd y3 ;Z Tr_210
NTRANS y64 y91 y3 ;Z Tr_211
NTRANS y57 Gnd y103 ;Z Tr_212
PTRANS y73 Vdd y91 ;Z Tr_213
PTRANS y57 Vdd y103 ;Z Tr_214
NTRANS y91 Gnd y89 ;Z Tr_215
NTRANS y91 Gnd y89 ;Z Tr_216
NTRANS y90 Gnd y93 ;Z Tr_217
PTRANS y90 Vdd y93 ;Z Tr_218
NTRANS y74 Gnd y90 ;Z Tr_219
PTRANS y89 Vdd y2 ;Z Tr_220
PTRANS y74 y2 y90 ;Z Tr_221
NTRANS y89 Gnd y90 ;Z Tr_222
NTRANS y56 Gnd y87 ;Z Tr_223
PTRANS y62 y87 y61 ;Z Tr_224
PTRANS y56 Vdd y61 ;Z Tr_225
NTRANS y62 Gnd y87 ;Z Tr_226
PTRANS y112 Vdd y86 ;Z Tr_227
NTRANS y88 Gnd y55 ;Z Tr_228
NTRANS y112 y86 y55 ;Z Tr_229
PTRANS y88 Vdd y86 ;Z Tr_230
PTRANS y87 Vdd y63 ;Z Tr_231
NTRANS y87 Gnd y63 ;Z Tr_232
PTRANS y86 Vdd y58 ;Z Tr_233
NTRANS y86 Gnd y58 ;Z Tr_234
END

```

La equivalencia entre la nominación de los nodos correspondientes a los terminales del Circuito Combinacional (CC2) de la MEF2\_C del Plano 1 y su numeración en el Plano 3(b) es la siguiente:

NODOS DEL CIRCUITO LOGICO	NODOS DEL LAYOUT
A23(entrada)	y123
A22(entrada)	y105
A21(entrada)	y112
A0 (entrada)	y64
D25(salida)	y72
D24(salida)	y70
D23(salida)	y78
D22(salida)	y60
D21(salida)	y84
D20(salida)	y85

Las combinaciones lógicas a aplicarse a los nodos de entrada del circuito combinacional del bloque MEF2\_C en el circuito equivalente del layout son las mismas que se aplicaron a sus terminales de entrada en el circuito lógico, los resultados de esta simulación se indican en la Fig.10.9.

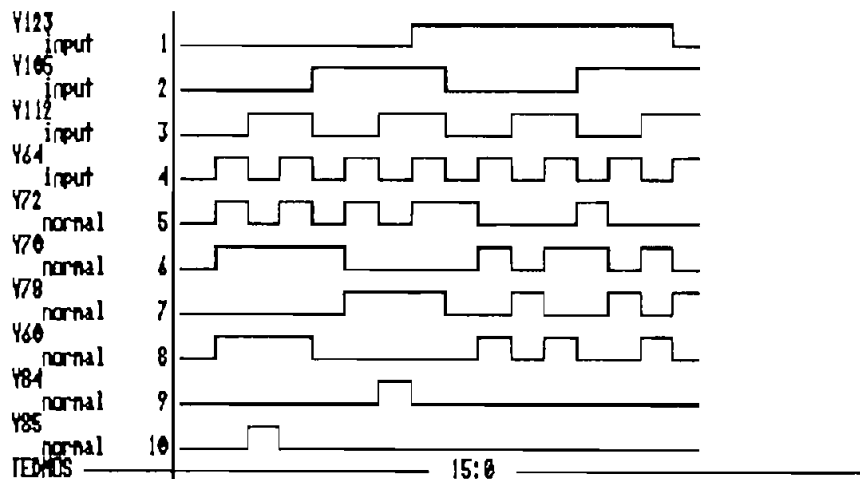


FIGURA 10.9 Simulación lógica del layout del CC2 del Plano 3 (b)

Los valores que adoptan los nodos de salida del circuito combinacional coinciden con aquellos de la simulación de su circuito lógico (Fig.10.3) y con los de su tabla funcional (Tabla 10.2) 10.4), en consecuencia la formulación de este bloque al nivel de diseño físico es equivalente con la formulación a los niveles lógico y funcional.

#### d) Diseño de los bloques de salida

En el Plano 1, se puede observar que los bloques de la etapa de salida: Salida 1 y Salida 2 son topológicamente idénticos, su única diferencia radica en la nominación de los terminales, por ello es suficiente diseñar uno de estos bloques (por ejemplo: Salida 1) y luego duplicar su estructura con lo que se construiría en su totalidad la etapa de salida del plano de base de la Fig.10.5.

#### i) Netlist de descripción

La NETLIST SPICE que describe las interconexiones entre las celdas del bloque Salida 1 del Plano 1 se indica a continuación. La orientación de los terminales de interfaz de este bloque de acuerdo con el plano de base de la Fig.10.5 debe atenerse a las siguientes restricciones:

i.1) Las señales de entrada: D20, D22, HDBn0\_c, ~HDBn0\_c, HDBn1\_c, ~HDBn1\_c, CLK\_c y ~CLK\_c, deben tener orientación NORTE.

i.2) La señal de salida: OUT\_LSB, debe tener orientación SUR.

En consecuencia, la NETLIST para la síntesis del módulo Salida 1 es la siguiente:

```

**-----**
** RELATORIO SPICE                OUT1_C.sim **
** CIRCUITO: CODIFICADOR HDBn    **
** BLOQUE: SALIDA 1              **
**-----**

X52 D22    CLK_c    ^CLK_c ^F11  F11    vcc    FFD_MS
X55 ^HDBn0_c ^HDBn1_c D22    I11    vcc    NAND3
X56 HDBn0_c HDBn1_c F13    I14    vcc    NAND3
X57 ^HDBn0_c HDBn1_c F12    I13    vcc    NAND3
X58 HDBn0_c ^HDBn1_c F11    I12    vcc    NAND3
X50 F12    CLK_c    ^CLK_c ^F13  F13    vcc    FFD_MS
X51 F11    CLK_c    ^CLK_c ^F12  F12    vcc    FFD_MS
X53 I16    CLK_c    ^CLK_c ^F14  OUT_LSB vcc    FFD_MS
X54 I11    I12     I13    I14    I15    vcc    NAND4
X59 I15    D20     I16    vcc    OR2

** TERMINALES DE INTERFAZ DEL MODULO **
** ----- **

# interface: D20      # orientacao=N #
# interface: D22      # orientacao=N #
# interface: HDBn0_c # orientacao=N #
# interface: ^HDBn0_c # orientacao=N #
# interface: HDBn1_c # orientacao=N #
# interface: ^HDBn1_c # orientacao=N #
# interface: CLK_c    # orientacao=N #
# interface: ^CLK_c   # orientacao=N #
# interface: OUT_LSB  # orientacao=S #

.END

```

## ii) Parámetros de particionamiento de celdas y de polarización

Los parámetros para el particionamiento de las celdas y de orientación de las líneas de polarización en este bloque son los mismos que se adoptaron para los bloques anteriores, a excepción del número de bandas..

De acuerdo con el plano de base (Fig.10.5) el número de bandas requerido para la distribución de las celdas es de "3", así:

c: Número de Bandas: 3

es de esperarse que las bandas sean más cortas que en los casos anteriores ya que el número de celdas es menor, no obstante no se debe olvidar que este bloque será duplicado al construirse el bloque Salida 2 de modo que el largo de las bandas se doblará.

### iii) Distribución de celdas

La distribución de las celdas en este bloque, generada por los programas de particionamiento y posicionamiento de acuerdo con los parámetros asignados se indica en la Fig.10.10.

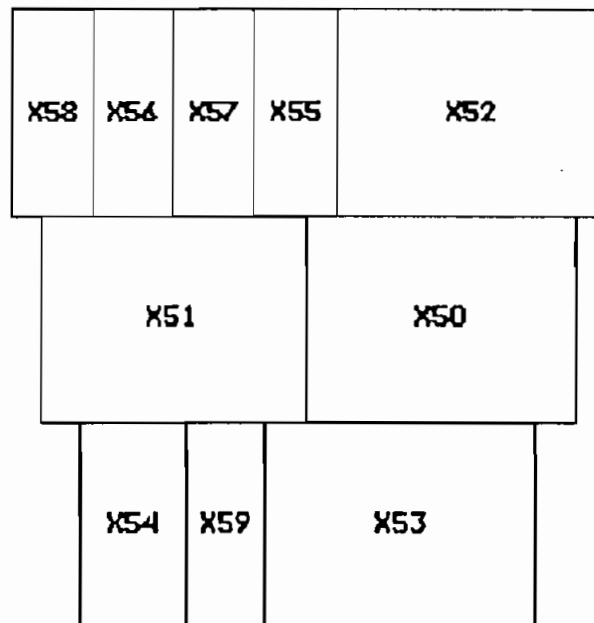


FIGURA 10.10 Distribución de celdas en el bloque Salida 1.

La posición final en que se disponen las celdas dentro de la distribución de la Fig.10.10 es el siguiente:

Posicionamiento de circuito C:\HDBn\OUT1\_C.pos

Banda 1

celula= X58 coordenada\_simbolica= 0 status rotacional=espejada

celula= X56 coordenada\_simbolica= 56 status rotacional=espelhada  
 celula= X57 coordenada\_simbolica= 112 status rotacional=normal  
 celula= X55 coordenada\_simbolica= 168 status rotacional=normal  
 celula= X52 coordenada\_simbolica= 224 status rotacional=normal

Banda 2

celula= X51 coordenada\_simbolica= 20 status rotacional=normal  
 celula= X50 coordenada\_simbolica= 203 status rotacional=normal

Banda 3

celula= X54 coordenada\_simbolica= 48 status rotacional=normal  
 celula= X59 coordenada\_simbolica= 121 status rotacional=normal  
 celula= X53 coordenada\_simbolica= 175 status rotacional=normal

iv) Generación del layout

El layout del bloque Salida 1 sintetizado a partir de la distribución de celdas de la Fig.10.10 se indica en el Plano 2 (c), su estructura jerárquica en formato RS es la que se indica a continuación:

DS 1 1 10;	DS 23 1 10;	DS 25 1 10;
9 "FFD_MS.CEL";	9 "BANDA 1";	9 "BANDA 3";
,	C 2 T -152 0 MX;	C 3 T 168 0;
,	C 2 T -260 0 MX;	C 4 T 253 0;
DF;	C 2 T 260 0;	C 1 T 319 0;
DS 2 1 10;	C 2 T 328 0;	9 "ENLACES ENTRE CELDAS";
9 "NAND_3.CEL";	C 1 T 396 0;	,
,	9 "ENLACES ENTRE CELDAS";	,
,	,	DF;
DF;	,	DS 26 1 10;
DS 3 1 10;	DF;	9 "LAYOUT GLOBAL";
9 "NAND_4.CEL";		C 23 T 0 200;
,	DS 24 1 10;	C 24 T 0 100;
,	9 "BANDA 2";	C 25;
DF;	C 1 T 204 -106 MY;	9 "CAMINOS DE POLARIZACION"
DS 4 1 10;	C 1 T 399 -106 MY;	,
9 "OR_2.CEL";	9 "ENLACES ENTRE CELDAS";	,
,	,	DF;
,	,	C 26;
DF;	DF;	E

Las características generales del layout del Plano 2(c) y las líneas de interconexión no formadas en este son las siguientes:

Roteamento da Banda 1 do Circuito C:\HDBn\OUT1\_C  
 Largura Banda=606  
 Redes Nao Roteadas:  
     rede=~HDBn0\_c\_c\_orig= X57 pino=3 banda=1

Roteamento da Banda 2 do Circuito C:\HDBn\OUT1\_C  
 Largura Banda=606  
 Redes Nao Roteadas:

Roteamento da Banda 3 do Circuito C:\HDBn\OUT1\_C  
 Redes Nao Roteadas:

Informacoes do Modulo Gerado  
 Area do Circuito = 200124  
 Relacao de Aspecto = 0.468

Se concluye que únicamente falta un camino de interconexión en la banda 1. El Plano 3(c) muestra el layout del módulo Salida 1 en que se ha realizado el enlace faltante y se han corregido los errores característicos de los layouts generados por el sistema TENTOS.

#### v) Verificación funcional del layout

El circuito equivalente al layout del bloque Salida 1 tiene la estructura de nodos y transistores cuya numeración se indica en el layout del Plano 3 (c), la NETLIST NDL que describe este circuito es la siguiente:

```

Z CIRCUITO: OUT1_C.ndl
Z TECNOLOGIA: ECPD15
Z CIRCUITO CODIFICADOR HDBn
Z SALIDA 1
Z
Z Transistores tipo NMOS: 71
Z Transistores tipo PMOS: 71

INPUT #73 #78 #71 #43 #50 #52;
INPUT #68 #64;

PTRANS #75 Vdd #42 ;Z Tr_1
PTRANS #74 Vdd #41 ;Z Tr_2
NTRANS #78 Gnd #40 ;Z Tr_3
NTRANS #75 #40 #62 ;Z Tr_4

PTRANS #80 #62 #42 ;Z Tr_5
NTRANS #74 #45 #39 ;Z Tr_6
NTRANS #78 Gnd #39 ;Z Tr_7
PTRANS #79 #41 #45 ;Z Tr_8
NTRANS #80 Gnd #62 ;Z Tr_9
NTRANS #79 Gnd #45 ;Z Tr_10
PTRANS #78 Vdd #42 ;Z Tr_11
PTRANS #78 Vdd #41 ;Z Tr_12
NTRANS #62 Gnd #80 ;Z Tr_13
NTRANS #45 Gnd #79 ;Z Tr_14
PTRANS #78 Vdd #36 ;Z Tr_15
PTRANS #78 Vdd #35 ;Z Tr_16
PTRANS #62 #80 #36 ;Z Tr_17
NTRANS #78 Gnd #38 ;Z Tr_18
NTRANS #60 #38 #80 ;Z Tr_19

PTRANS #45 #35 #79 ;Z Tr_20
NTRANS #78 Gnd #37 ;Z Tr_21
NTRANS #77 #79 #37 ;Z Tr_22
PTRANS #60 Vdd #36 ;Z Tr_23
PTRANS #77 Vdd #35 ;Z Tr_24
PTRANS #70 Vdd #30 ;Z Tr_25
PTRANS #76 #30 #61 ;Z Tr_26
NTRANS #78 Gnd #33 ;Z Tr_27
NTRANS #70 #61 #33 ;Z Tr_28
PTRANS #69 Vdd #34 ;Z Tr_29
NTRANS #73 Gnd #32 ;Z Tr_30
NTRANS #69 #32 #60 ;Z Tr_31
PTRANS #71 Vdd #29 ;Z Tr_32
PTRANS #75 #60 #34 ;Z Tr_33
NTRANS #73 Gnd #31 ;Z Tr_34

```

NTRANS	■71	■77	■31	;% Tr_35	PTRANS	■71	Vdd	■57	;% Tr_72	NTRANS	■52	Gnd	■9	;% Tr_109
NTRANS	■76	Gnd	■61	;% Tr_36	PTRANS	■66	Vdd	■19	;% Tr_73	PTRANS	■52	Vdd	■49	;% Tr_110
NTRANS	■75	Gnd	■60	;% Tr_37	NTRANS	■78	Gnd	■20	;% Tr_74	PTRANS	■73	Vdd	■11	;% Tr_111
PTRANS	■74	■29	■77	;% Tr_38	NTRANS	■66	■20	■56	;% Tr_75	PTRANS	■43	■8	■67	;% Tr_112
PTRANS	■78	Vdd	■30	;% Tr_39	PTRANS	■69	■56	■19	;% Tr_76	PTRANS	■65	Vdd	■8	;% Tr_113
NTRANS	■74	Gnd	■77	;% Tr_40	NTRANS	■58	Gnd	■70	;% Tr_77	NTRANS	■65	Gnd	■67	;% Tr_114
PTRANS	■73	Vdd	■34	;% Tr_41	NTRANS	■69	Gnd	■56	;% Tr_78	PTRANS	■80	Vdd	■48	;% Tr_115
PTRANS	■73	Vdd	■29	;% Tr_42	PTRANS	■73	Vdd	■16	;% Tr_79	NTRANS	■80	Gnd	■7	;% Tr_116
NTRANS	■61	Gnd	■76	;% Tr_43	PTRANS	■78	Vdd	■19	;% Tr_80	NTRANS	■53	Gnd	■66	;% Tr_117
NTRANS	■60	Gnd	■75	;% Tr_44	NTRANS	■68	■15	■18	;% Tr_81	PTRANS	■73	Vdd	■5	;% Tr_118
NTRANS	■77	Gnd	■74	;% Tr_45	PTRANS	■58	■16	■70	;% Tr_82	PTRANS	■52	Vdd	■48	;% Tr_119
PTRANS	■78	Vdd	■26	;% Tr_46	PTRANS	■68	Vdd	■57	;% Tr_83	NTRANS	■52	■7	■3	;% Tr_120
PTRANS	■73	Vdd	■24	;% Tr_47	NTRANS	■55	■70	■17	;% Tr_84	PTRANS	■53	■66	■5	;% Tr_121
PTRANS	■73	Vdd	■23	;% Tr_48	NTRANS	■73	Gnd	■17	;% Tr_85	NTRANS	■51	■6	■66	;% Tr_122
PTRANS	■61	■26	■76	;% Tr_49	PTRANS	■55	Vdd	■16	;% Tr_86	NTRANS	■73	Gnd	■6	;% Tr_123
NTRANS	■58	■76	■28	;% Tr_50	NTRANS	■50	Gnd	■15	;% Tr_87	PTRANS	■51	Vdd	■5	;% Tr_124
PTRANS	■60	■75	■24	;% Tr_51	PTRANS	■50	Vdd	■57	;% Tr_88	NTRANS	■57	■65	■4	;% Tr_125
NTRANS	■78	Gnd	■28	;% Tr_52	NTRANS	■54	Gnd	■55	;% Tr_89	PTRANS	■57	Vdd	■65	;% Tr_126
NTRANS	■73	Gnd	■27	;% Tr_53	PTRANS	■54	Vdd	■55	;% Tr_90	NTRANS	■79	Gnd	■51	;% Tr_127
NTRANS	■59	■27	■75	;% Tr_54	NTRANS	■56	Gnd	■69	;% Tr_91	PTRANS	■79	Vdd	■51	;% Tr_128
PTRANS	■77	■23	■74	;% Tr_55	PTRANS	■78	Vdd	■13	;% Tr_92	PTRANS	■48	Vdd	■65	;% Tr_129
NTRANS	■72	■74	■25	;% Tr_56	PTRANS	■56	■69	■13	;% Tr_93	NTRANS	■48	■44	■4	;% Tr_130
PTRANS	■58	Vdd	■26	;% Tr_57	NTRANS	■78	Gnd	■14	;% Tr_94	NTRANS	■64	■48	■3	;% Tr_131
NTRANS	■73	Gnd	■25	;% Tr_58	NTRANS	■53	■14	■69	;% Tr_95	PTRANS	■64	Vdd	■48	;% Tr_132
PTRANS	■59	Vdd	■24	;% Tr_59	PTRANS	■53	Vdd	■13	;% Tr_96	PTRANS	■49	Vdd	■65	;% Tr_133
PTRANS	■72	Vdd	■23	;% Tr_60	NTRANS	■68	■49	■12	;% Tr_97	NTRANS	■49	■44	■2	;% Tr_134
NTRANS	■69	Gnd	■59	;% Tr_61	PTRANS	■68	Vdd	■49	;% Tr_98	NTRANS	■63	Gnd	■2	;% Tr_135
PTRANS	■69	Vdd	■59	;% Tr_62	NTRANS	■67	Gnd	■54	;% Tr_99	PTRANS	■63	Vdd	■65	;% Tr_136
NTRANS	■71	Gnd	■72	;% Tr_63	PTRANS	■67	Vdd	■54	;% Tr_100	PTRANS	■50	Vdd	■63	;% Tr_137
PTRANS	■71	Vdd	■72	;% Tr_64	NTRANS	■69	■9	■12	;% Tr_101	NTRANS	■50	Gnd	■46	;% Tr_138
PTRANS	■54	Vdd	■21	;% Tr_65	PTRANS	■79	Vdd	■11	;% Tr_102	PTRANS	■64	Vdd	■63	;% Tr_139
NTRANS	■73	Gnd	■22	;% Tr_66	NTRANS	■73	Gnd	■10	;% Tr_103	NTRANS	■64	■46	■47	;% Tr_140
NTRANS	■54	■58	■22	;% Tr_67	NTRANS	■79	■10	■53	;% Tr_104	NTRANS	■79	■63	■47	;% Tr_141
PTRANS	■70	■21	■58	;% Tr_68	PTRANS	■69	Vdd	■49	;% Tr_105	PTRANS	■79	Vdd	■63	;% Tr_142
NTRANS	■70	Gnd	■58	;% Tr_69	PTRANS	■66	■53	■11	;% Tr_106	END				
PTRANS	■73	Vdd	■21	;% Tr_70	NTRANS	■66	Gnd	■53	;% Tr_107					
NTRANS	■71	■57	■18	;% Tr_71	NTRANS	■43	Gnd	■67	;% Tr_108					

La correspondencia entre la numeración de los nodos del Plano 3(c) y su nominación en el bloque Salida 1 del Plano 1 para los nodos a graficarse en los diagramas de simulación es la siguiente:



NODOS DEL CIRCUITO LOGICO	NODOS DEL LAYOUT
HDBn1_C (entrada)	m52
HDBn0_C (entrada)	m64
D22 (entrada)	m71
D20 (entrada)	m43
F11 (nodo interno)	m79
F12 (nodo interno)	m69
F13 (nodo interno)	m80
I15 (nodo interno)	m65
I16 (nodo interno)	m54
OUT_LSB (salida)	m76

A fin de verificar la correspondencia entre el circuito lógico del Plano 1 y el circuito equivalente del layout del Plano 3(c) se han aplicado a los nodos de entrada m71(D22) y m43(D20) y de selección m52(HDBn1\_C) y m65(HDBn0\_C) de este último el mismo conjunto de señales que se emplearon en la simulación del circuito lógico, los resultados obtenidos para los cuatro casos de selección se indican en la Fig.10.11.

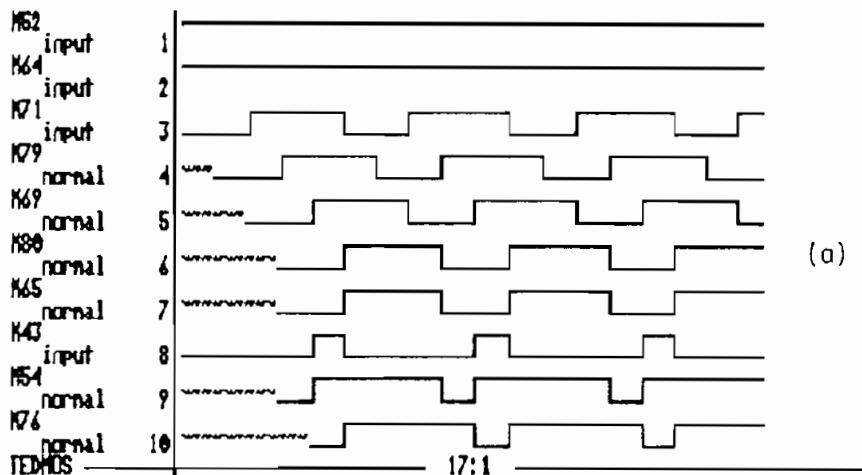


FIGURA 10.11 Simulación del layout del plano 3(c), (a) caso #52=1 #64=1.

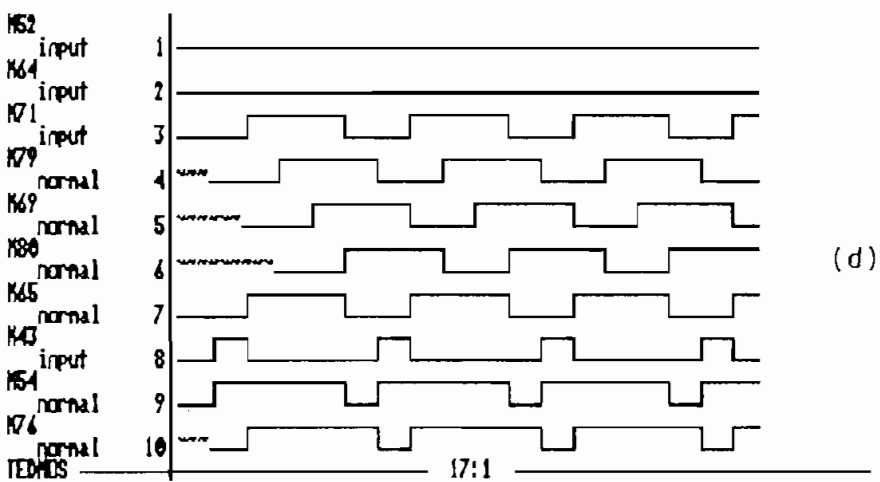
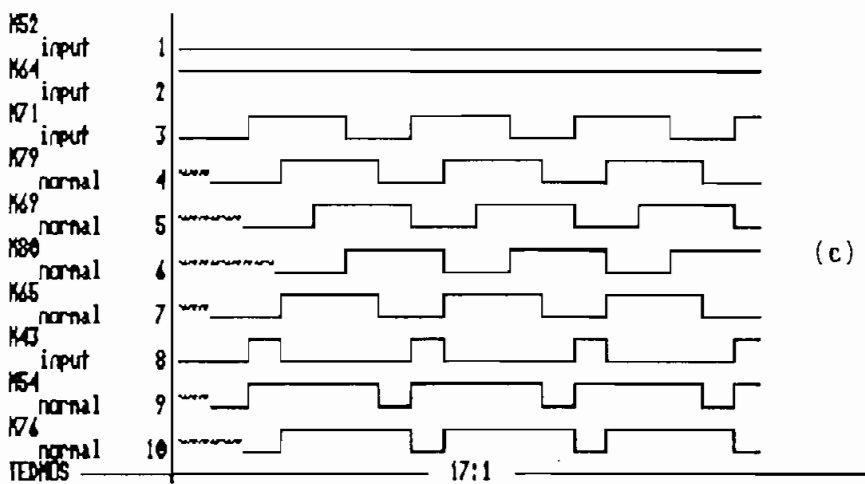
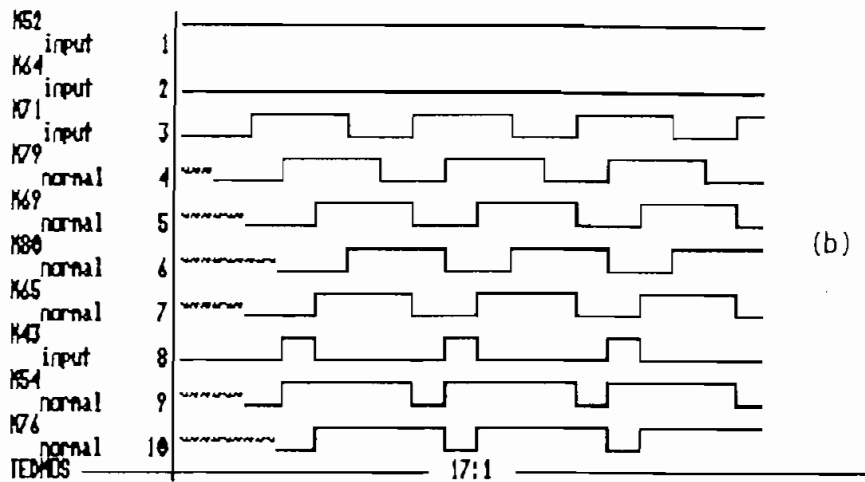


FIGURA 10.11 Cont.. (b) caso  $m52=1$   $m64=0$  (c) caso  $m52=0$   $m64=1$ ,  
 (d) caso  $m52=0$   $m64=0$ .

Los diagramas de simulación del circuito equivalente del layout indicados (Fig.10.11) coinciden con aquellos de la simulación del circuito lógico (Fig.10.4, y Tabla 10.4), en consecuencia los diseños del bloque Salida 1 a los niveles físico y lógico son equivalentes.

vi) **Generación del layout del módulo de salida**

Los módulos Salida 1 y Salida 2 del Plano 1 tienen un circuito lógico similar por lo que físicamente son idénticos, por tanto una vez construido uno de ellos, el layout del otro es igual y el layout del módulo de salida se forma colocando ambos bloques de manera adyacente y reconstruyendo los caminos de polarización como se indica en la Fig.10.12.

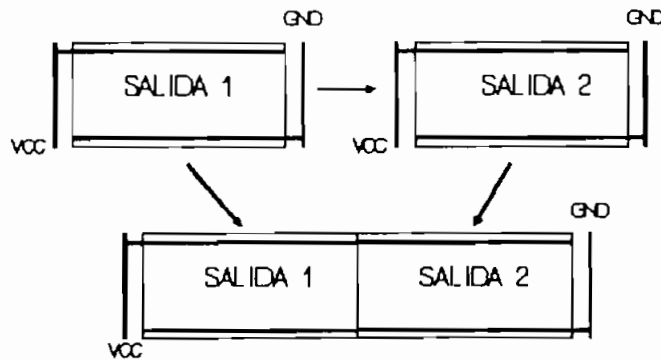


FIGURA 10.12 Generación del módulo de salida.

e) Generación del layout del Codificador HDBn

Una vez diseñados los bloques del plano de base de la Fig.10.5, el layout del Codificador HDBn se construye siguiendo los siguientes pasos:

- i) Se integran los bloques diseñados en un solo layout, según su posición en el plano de base.
- ii) Se construyen las líneas de interconexión siguiendo las estrategias de enrutamiento global formulada en el plano de base, los caminos horizontales han sido construidos en Metal 1 y los verticales en Polysilicon a excepción de las líneas de polarización que también son en Metal1.

No habiendo programa en el sistema TENTOS que realice este proceso de manera automática, se lo ha hecho en el Editor de máscaras EMA2 en forma manual a manera de un diseño "FULL-CUSTOM", el layout del circuito Codificador HDBn así obtenido se indica en el Plano 4 del Anexo H.

## 10.2 DISEÑO DEL DECODIFICADOR PROGRAMABLE HDBn

### 10.2.1 Diseño a nivel funcional

Los algoritmos y ecuaciones que describen funcionalmente al Decodificador HDBn se estudian en el Capítulo 9, el esquema deducido para su circuito se indica en la Fig.9.55, que con ciertas modificaciones de nomenclatura a fin de evitar duplicación de nombres se indica en la Fig.10.13.

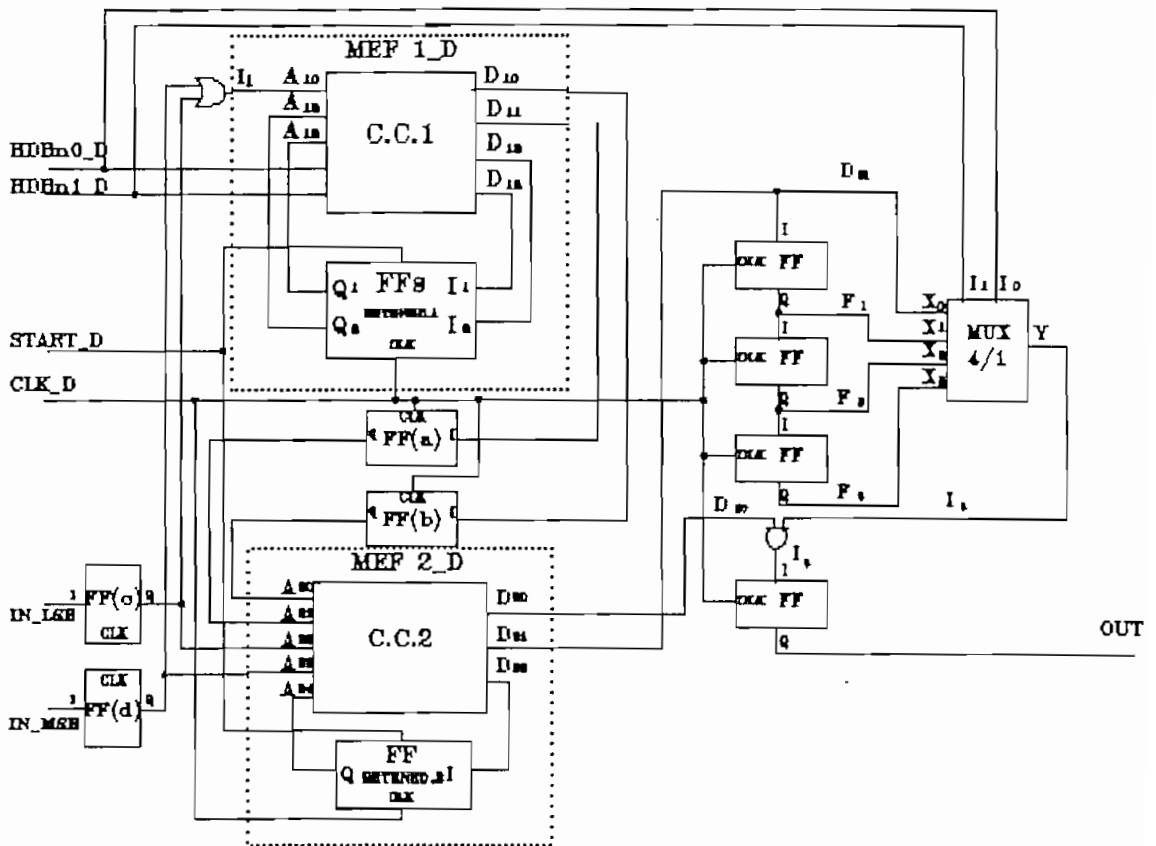


FIGURA 10.13 Esquema funcional del DECODIFICADOR HDBn correspondiente al Plano 5.

### 10.2.2 Diseño a nivel lógico

#### a) Formulación del diseño

La interpretación al nivel lógico de los bloques funcionales de la Fig.10.13, se ha basado en la misma biblioteca de celdas empleada en los diseños anteriores (Anexo F.1).

La circuito lógico deducido para el circuito Decodificador HD8n se grafica en el Plano 5 del Anexo H, su estructura es la siguiente:

##### i) Circuito Combinacional 1 (CC1)

La relación lógica entre los terminales de salida (D) del CC1 y las de entrada (A) está dada por la Tabla 9.28 que adaptada a la nomenclatura del Plano 5 se indica en la Tabla 10.5, las ecuaciones que interpretan esta relación lógica fueron deducidas en el numeral 9.3.2 y para la Tabla 10.5 son las siguientes:

$$D_{13} = \overline{HDEN_0_D} \overline{A_{11}} \overline{A_{10}} + HDEN_1_D \overline{HDEN_0_D} \overline{A_{12}} A_{11} \overline{A_{10}} + HDEN_0_D A_{12} \overline{A_{11}} \overline{A_{10}}$$

$$D_{12} = HDEN_0_D \overline{A_{11}} \overline{A_{10}} + HDEN_1_D \overline{HDEN_0_D} \overline{A_{12}} \overline{A_{11}} \overline{A_{10}}$$

$$D_{11} = \overline{HDEN_1_D} \overline{HDEN_0_D} \overline{A_{11}} \overline{A_{10}} + \overline{HDEN_0_D} A_{11} \overline{A_{10}} + HDEN_0_D A_{12} \overline{A_{11}} \overline{A_{10}}$$

$$D_{10} = \overline{HDEN_1_D} A_{11} + \overline{HDEN_1_D} \overline{HDEN_0_D} A_{10} + HDEN_0_D \overline{A_{12}} A_{11} \overline{A_{10}} \\ + HDEN_1_D \overline{HDEN_0_D} \overline{A_{12}} \overline{A_{11}} \overline{A_{10}}$$

ENTRADAS					SALIDAS			
HDBn <sub>1</sub> _D	HDBn <sub>0</sub> _D	A <sub>12</sub>	A <sub>11</sub>	A <sub>10</sub>	D <sub>13</sub>	D <sub>12</sub>	D <sub>11</sub>	D <sub>10</sub>
0	0	0	0	0	0	0	0	0
0	0	0	0	1	0	0	0	0
0	1	0	0	0	0	1	1	0
0	1	0	0	1	0	0	0	1
0	1	0	1	0	0	0	0	1
0	1	0	1	1	0	0	0	1
1	0	0	0	0	0	1	0	1
1	0	0	0	1	0	0	0	0
1	0	0	1	0	1	0	1	0
1	0	0	1	1	0	0	0	0
1	0	1	0	0	0	0	0	0
1	0	1	0	1	0	0	0	0
1	1	0	0	0	0	1	0	0
1	1	0	0	1	0	0	0	0
1	1	0	1	0	1	0	0	1
1	1	0	1	1	0	0	0	0
1	1	1	0	0	1	1	1	0
1	1	1	0	1	0	0	0	0
1	1	1	1	0	0	0	0	0
1	1	1	1	1	0	0	0	0
1	1	1	1	1	0	0	0	0

TABLA 10.5 Formulación funcional del Circuito Combinacional de la MEF1 del Decodificador HDBn.

ii) FFs Retenedores 1

Este módulo está formado por dos celdas FFD\_MS (X24 y X25) sincronizan y enlazan las señales de realimentación de los estados lógicos al CC1 de la máquina de estados finitos (MEF1\_D). Las celdas AND2 (X5 y X6) a la entrada de cada FFD\_MS permiten incluir la señal de habilitación START\_D de modo que mientras START\_D=0, la salida de los FFs sea Q=0 y el CC1 se mantenga en su estado inicial, hasta que se especifique lo contrario (START\_D=1).

iii) Circuito Combinacional 2 (CC2)

La relación lógica entre los terminales de entrada (A) y de salida (D) del CC2 del Plano 5 está dada por la Tabla 10.6 (reformulada a partir de la Tabla 9.41), las ecuaciones que relacionan estos terminales se dedujeron en el numeral 9.3.3 y son las siguientes:

$$D_{12} = \overline{HDEn_{1,D}} \overline{HDEno_{,D}} + \overline{HDEn_{1,D}} A_{12} \overline{A_{11}} \overline{A_{10}} + \overline{HDEn_{1,D}} \overline{HDEno_{,D}} \overline{A_{12}} A_{11} \\ + \overline{HDEn_{1,D}} \overline{A_{12}} A_{10} + HDEno_{,D} A_{11} + \overline{HDEn_{1,D}} \overline{HDEno_{,D}} \overline{A_{12}} \overline{A_{11}}$$

$$D_{11} = \overline{HDEn_{1,D}} \overline{HDEno_{,D}} + \overline{HDEn_{1,D}} A_{12} + \overline{HDEno_{,D}} \overline{A_{12}} A_{11} + A_{12} \overline{A_{11}} \overline{A_{10}} + \\ + HDEno_{,D} \overline{A_{11}} \overline{A_{10}}$$

$$D_{10} = \overline{A_{10}} + \overline{HDEn_{1,D}} \overline{A_{12}} + \overline{HDEn_{1,D}} \overline{HDEno_{,D}}$$

ENTRADAS					SALIDAS		
A <sub>24</sub>	A <sub>23</sub>	A <sub>22</sub>	A <sub>21</sub>	A <sub>20</sub>	D <sub>22</sub>	D <sub>21</sub>	D <sub>20</sub>
0	0	0	0	0	0	0	1
0	0	0	0	1	0	0	1
0	0	0	1	0	1	1	1
0	0	1	0	0	1	1	1
0	0	1	0	1	0	0	0
0	0	1	1	0	0	0	1
0	1	0	0	0	1	1	1
0	1	0	0	1	1	1	1
0	1	0	1	0	1	1	1
1	0	0	0	0	1	0	1
1	0	0	0	1	1	0	1
1	0	0	1	0	0	1	1
1	0	1	0	0	0	1	1
1	0	1	0	1	0	1	1
1	0	1	1	0	0	1	1
1	1	0	0	0	0	1	1
1	1	0	0	1	1	0	0
1	1	0	1	0	1	0	1

TABLA 10.6 Formulación funcional del Circuito Combinacional de la MEF2 del Decodificador HDBn.



En los terminales de salida del módulo CC2 se han colocado las celdas AND2 (X37, X38 y X39) para mantener las señales de salida de este módulo en "0" mientras START\_D=0 durante la inicialización del funcionamiento de la máquina de estados finitos 2 (MEF2\_D).

iv) FFs Retenedores 2

Este módulo lo forman las celdas FFD\_MS (X53 y X54) que sincronizan y enlazan las señales de realimentación al circuito combinacional 2 (CC2) de la máquina de estados finitos (MEF2\_D), no cuenta con la señal de inicialización START\_D puesto que esta ya fue considerada en el módulo CC2

v) Multiplexor de salida

El módulo de multiplexación 4/1 de la Fig.10.13 debe tener el comportamiento funcional que se indica en la Tabla 10.7.

ENTRADA				SELECCION		SALIDA
F <sub>1</sub>	F <sub>2</sub>	F <sub>3</sub>	F <sub>4</sub>	HDBn <sub>1_p</sub>	HDBn <sub>0_p</sub>	I <sub>5</sub>
F <sub>1</sub>	.	.	.	0	0	F <sub>1</sub>
.	F <sub>2</sub>	.	.	0	1	F <sub>2</sub>
.	.	F <sub>3</sub>	.	1	0	F <sub>3</sub>
.	.	.	F <sub>4</sub>	1	1	F <sub>4</sub>

TABLA 10.7 Formulación funcional del Multiplexor 4/1 del Decodificador.

Su construcción en base a celdas estándar se ha realizado en base a las siguientes ecuaciones:

$$I_5 = \overline{F_1} \overline{HDBn_{1_p}} \overline{HDBn_{0_p}} + \overline{F_2} \overline{HDBn_{1_p}} HDBn_{0_p} + F_3 \overline{HDBn_{1_p}} \overline{HDBn_{0_p}} + F_4 HDBn_{1_p} HDBn_{0_p}$$

$$I_5 = (F_1 \overline{HDBn_{1_p}} \overline{HDBn_{0_p}})' (F_2 \overline{HDBn_{1_p}} HDBn_{0_p})' (F_3 HDBn_{1_p} \overline{HDBn_{0_p}})' (F_4 HDBn_{1_p} HDBn_{0_p})$$

## b) Simulación lógica

Debido a las limitaciones en la capacidad del simulador NDL el circuito lógico del Plano 5 ha sido dividido en módulos que serán simulados individualmente a fin de verificar su correspondencia con las tablas que definen su comportamiento funcional.

De la misma manera que en las NETLISTs del Codificador HDBn, en las NETLISTs subsiguientes el caracter "~" en la nominación de los nodos del Plano 5 ha sido cambiado por "N", y se ha omitido la descripción de la red de cada subcircuito que se detalla en el Anexo F.1

### i) Simulación del Circuito Combinacional 1 (CC1)

La NETLIST NDL que describe el Circuito Combinacional 1 del Decodificador HDBn del Plano 5 es la siguiente:

```
-----
I SIMULACION LOGICA          CC1_D.nd1 |
I DECODIFICADOR HDBn        |
I F. LEMUS / I. BERNAL     |
I      CIRCUITO COMBINACIONAL 1 |
I-----
FAMILY CMOS;

-----
I DEFINICION DE SUBCIRCUITOS |
I-----

DEFINE AND2 I1 I2 Out;
.
END;

DEFINE AND3 In1 In2 In3 Out;
.
END;
```

```

DEFINE AND4 In1 In2 In3 In4 Out;
,
END;

DEFINE OR2 I1 I2 Out;
,
END;

DEFINE OR3 I1 I2 I3 Out;
,
END;

I-----
Z  LLAMADA A SUBCIRCUITOS          ;
Z-----
INPUT  A10 NA10 A11 NA11 A12 NA12 HDBn0_D NHDBn0_D HDBn1_D NHDBn1_D;

AND2  NHDBn0_D HDBn1_D P5                ;Z(X7)
AND2  P2      HDBn1_D P7                ;Z(X8)
AND2  P3      A12      P8                ;Z(X9)
AND2  P3      NHDBn1_D P9                ;Z(X10)
AND2  P4      P5      P10               ;Z(X11)
AND3  NA10    A11     NHDBn0_D P1       ;Z(X12)
AND3  NA10    NA11    HDBn0_D P3       ;Z(X13)
AND3  NA10    NA11    NA12    P4       ;Z(X14)
AND3  A10     HDBn0_D NHDBn1_D P6       ;Z(X15)
AND4  NA10    A11     NA12    HDBn0_D P2 ;Z(X16)
OR2   P3      P10     D12              ;Z(X17)
OR2   A23     A22     A10              ;Z(X18)
OR3   P1      P7      P8      D13      ;Z(X19)
OR3   P1      P8      P9      D11      ;Z(X20)
OR3   P2      P10     P6      D10      ;Z(X21)

END

```

Para la simulación de este circuito, las combinaciones lógicas han sido aplicadas a sus terminales de entrada de acuerdo con la Tabla 10.5 que define su comportamiento funcional, los valores lógicos que adoptan los terminales de salida (D) frente a estas excitaciones se grafican en la Fig.10.14, se observa en esta figura que estos valores coinciden con sus correspondientes en la Tabla 10.5, en consecuencia el circuito lógico CC1 del Decodificador HDBn del Plano 5 es funcionalmente correcto.

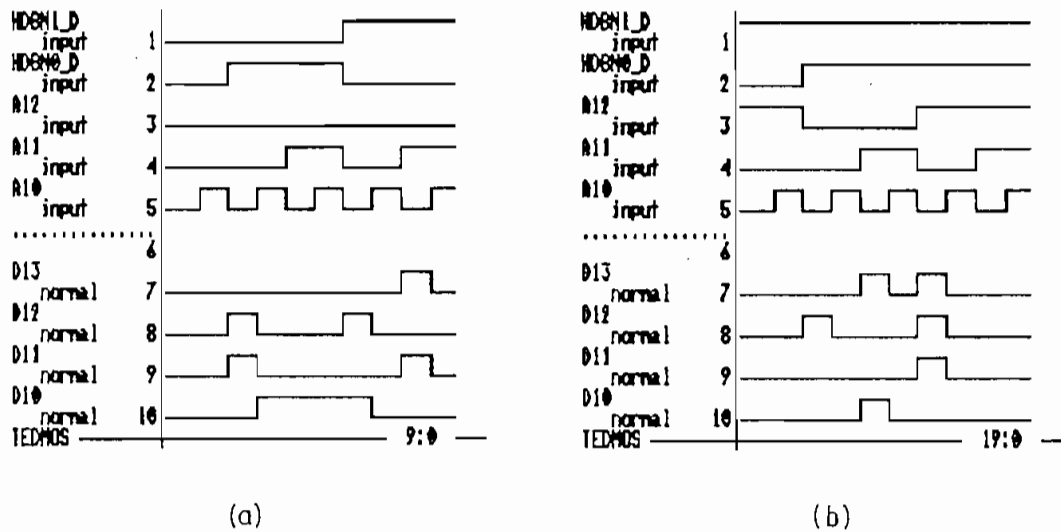


FIGURA 10.14 Simulación lógica del CCI del Decodificador HDBn.

ii) Simulación del Circuito Combinacional 2

.La NETLIST NDL que describe del Circuito Combinacional 2 (CC2) del Plano 5 es la siguiente:

```

X-----
X  SIMULACION LOGICA          CC2_D.NDL  ;
X  DECODIFICADOR HDBn
X  F. LEMUS / I. BERNAL
X  CIRCUITO COMBINACIONAL 2
X-----
FAMILY CMOS;

X-----
X  DEFINICION DE SUBCIRCUITOS
X-----
DEFINE AND2 I1 I2 Out;
.
END;

DEFINE AND3 In1 In2 In3 Out;
.
END;

DEFINE AND4 In1 In2 In3 In4 Out;
.
END;

```

```

DEFINE OR2 I1 I2 Out;
.
END;

DEFINE OR3 I1 I2 I3 Out;
.
END;

!-----
!  LLAMADA A SUBCIRCUITOS          ;
!-----
INPUT A20 NA20 A21 NA21 A22 NA22 A23 NA23 A24 NA24 START_D;

AND2  A23    NA24    Q1                ;Z(X30)
AND2  A21    A23     Q5                ;Z(X31)
AND2  A22    A24     Q7                ;Z(X32)
AND2  NA22   NA24    Q9                ;Z(X33)
AND2  NA23   A24    Q10               ;Z(X34)
AND2  Q2     NA24    Q11               ;Z(X35)
AND2  Q3     NA24    Q12               ;Z(X36)
AND2  Q19    START_D D22               ;Z(X37)
AND2  START_D Q15    D21               ;Z(X38)
AND2  START_D Q14    D20               ;Z(X39)
AND3  NA20   NA21    A22     Q2        ;Z(X40)
AND3  A21    NA22   NA23    Q3        ;Z(X41)
AND3  A20    NA22   A24     Q4        ;Z(X42)
AND3  NA20   NA21    A23     Q8        ;Z(X43)
AND4  NA21   NA22   NA23    A24     Q6  ;Z(X44)
OR2   Q7     Q8     Q13                ;Z(X45)
OR2   Q16    Q13    Q15                ;Z(X46)
OR2   Q17    Q18    Q19                ;Z(X47)
OR3   Q1     Q11    Q12     Q17       ;Z(X48)
OR3   Q4     Q5     Q6     Q18       ;Z(X49)
OR3   Q1     Q2     Q3     Q16       ;Z(X50)
OR3   NA20   Q9     Q10    Q14       ;Z(X51)

END

```

Las combinaciones lógicas aplicadas durante la simulación a los terminales de entrada de este circuito corresponden a las indicadas en la Tabla 10.6, los resultados de esta simulación se exhiben en la Fig.10.15, en ella se observa que los valores lógicos que adoptan los terminales de salida coinciden con sus correspondientes de la Tabla 10.6, en consecuencia el circuito lógico del módulo CC2 del Plano 5 es funcionalmente correcto.

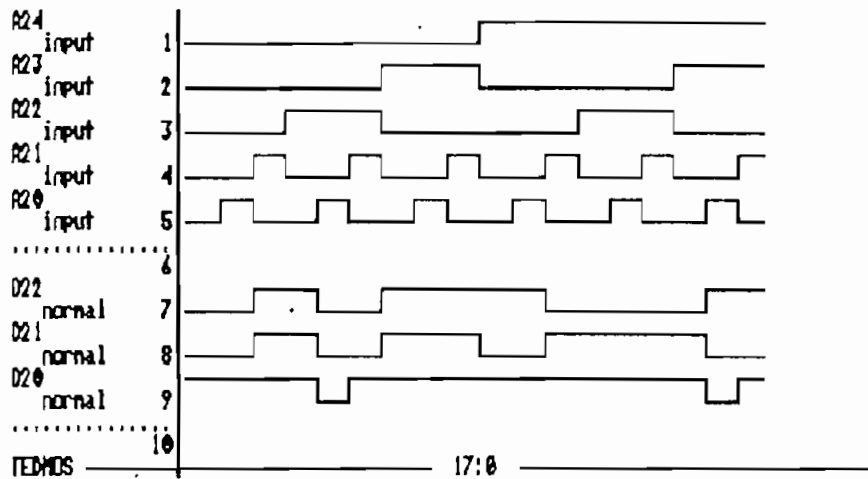


FIGURA 10.15 Simulación lógica del CC2 del Decodificador HDBn

### iii) Simulación de la Etapa de Salida

La NETLIST NDL que describe la etapa de salida del circuito lógico del Plano 6 es la siguiente:

```

X-----
X  SIMULACION LOGICA          OUT_D.NDL  ;
X  MULTIPLEXOR DE SALIDA      ;
X  F. LEMUS / I. BERNAL      ;
X  ETAPA DE SALIDA           ;
X-----
FAMILY CMOS;

X-----
X  DEFINICION DE SUBCIRCUITOS ;
X-----

DEFINE INV In Out;
.
END;

DEFINE AND2 I1 I2 Out;
.
END;

```

```

DEFINE NAND3 I1 I2 I3 Out;
,
END;

DEFINE NAND4 I1 I2 I3 I4 Out;
,
END;

DEFINE FFD_MS D CK CKM QN Q;
,
END;

X-----
X  LLAMADA A SUBCIRCUITOS                               X
X-----
INPUT  CLK_D NCLK_D HDBn1_D NHDBN1_D HDBn0_D NHDBN0_D D21 D20;

AND2  I5      D20      I6                               ;X(X60)
NAND3 NHDBn1_D NHDBn0_D D21      I1                     ;X(X61)
NAND3 NHDBn1_D HDBn0_D  F1       I2                     ;X(X62)
NAND3 HDBn1_D  NHDBn0_D  F2       I3                     ;X(X63)
NAND3 HDBn1_D  HDBn0_D  F3       I4                     ;X(X64)
NAND4 I1      I2 I3  I4      I5                         ;X(X65)
FFD_MS D21    CLK_D  NCLK_D  NF1    F1                  ;X(X66)
FFD_MS F1     CLK_D  NCLK_D  NF2    F2                  ;X(X67)
FFD_MS F2     CLK_D  NCLK_D  NF3    F3                  ;X(X68)
FFD_MS I6     CLK_D  NCLK_D  NF4    OUT                 ;X(X67)

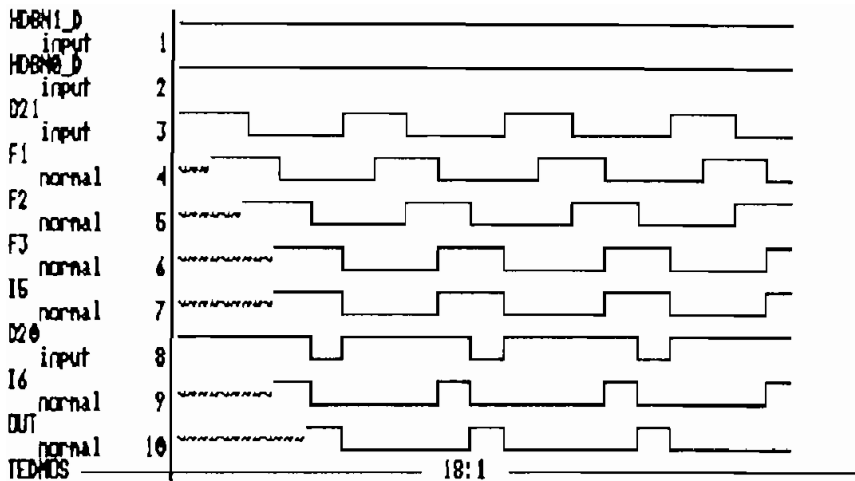
END

```

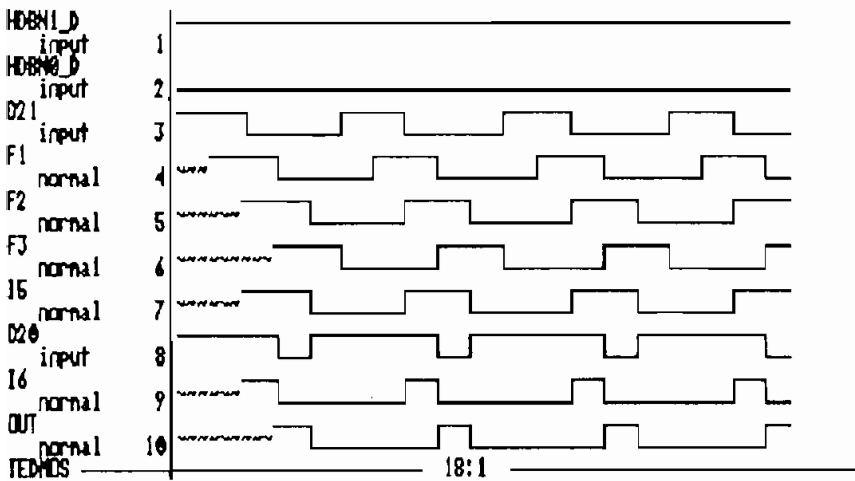
Los terminales de entrada para la simulación de este circuito son:

- i) HDBn1\_D y HDBn0\_D que permiten seleccionar en el multiplexor 4/1 el número de celdas FFD\_MS que se colocan en cascada.
- ii) D21 y D20 por los que se aplican la secuencia binaria original y la secuencia de modificación, respectivamente.

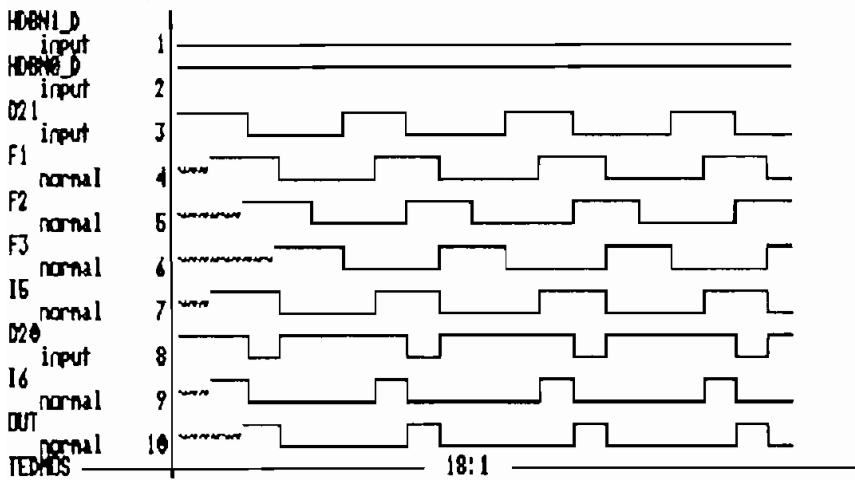
En la simulación se han considerado las cuatro opciones de selección en el multiplexor, obteniéndose los resultados que se indican en la Fig.10.16.



(a)



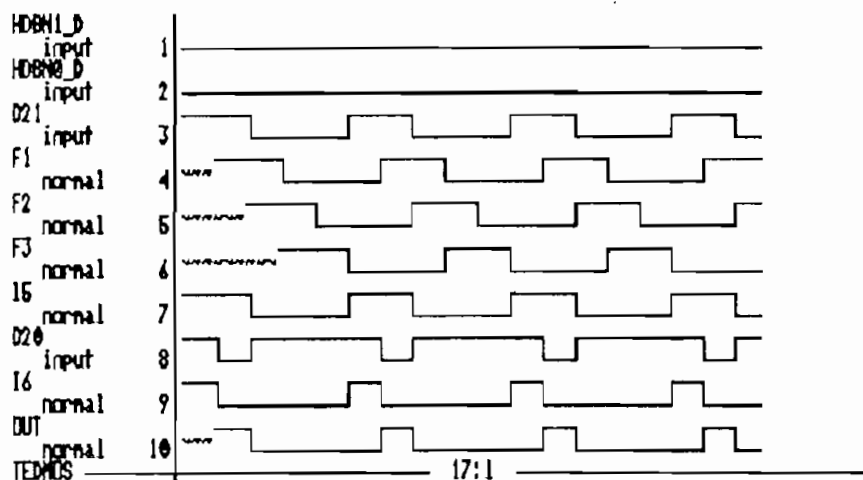
(b)



(c)

FIGURA 10.16 Simulación del bloque de salida (a) Caso  $HDBn_{1,p}=1$   $HDBn_{0,p}=1$   
 (b) Caso  $HDBn_{1,p}=1$   $HDBn_{0,p}=0$  (c) Caso  $HDBn_{1,p}=0$   $HDBn_{0,p}=1$





(d)

FIGURA 10.16 Cont.. (d) Caso HDBn<sub>1\_p</sub>=0 HDBn<sub>o\_p</sub>=0

Los resultados obtenidos en la simulación de este circuito se indican en la Tabla 10.8.

CASO	HDBn <sub>1_p</sub>	HDBn <sub>o_p</sub>	Salida del Multiplexor	Secuencia Entrante	Secuencia de modificación	Secuencia modificada	Retardo Entrada/Salida
(a)	1	1	I <sub>5</sub> = F <sub>3</sub> (3 FFs en cascada)	D <sub>21</sub> =1 <sub>1</sub> 1 <sub>2</sub> 0 <sub>3</sub> 0 <sub>4</sub> 0 <sub>5</sub>	D <sub>20</sub> =1 <sub>1</sub> 1 <sub>2</sub> 1 <sub>3</sub> 1 <sub>4</sub> 0 <sub>5</sub>	OUT=1 <sub>1</sub> 0 <sub>2</sub> 0 <sub>3</sub> 0 <sub>4</sub> 0 <sub>5</sub>	4 ciclos
(b)	1	0	I <sub>5</sub> = F <sub>2</sub> (2 FFs en cascada)	D <sub>22</sub> =1 <sub>1</sub> 1 <sub>2</sub> 0 <sub>3</sub> 0 <sub>4</sub> 0 <sub>5</sub>	D <sub>20</sub> =1 <sub>1</sub> 1 <sub>2</sub> 1 <sub>3</sub> 0 <sub>4</sub> 1 <sub>5</sub>	OUT=1 <sub>1</sub> 0 <sub>2</sub> 0 <sub>3</sub> 0 <sub>4</sub> 0 <sub>5</sub>	3 ciclos
(c)	0	1	I <sub>5</sub> = F <sub>1</sub> (1 FF entre D <sub>21</sub> e I <sub>5</sub> )	D <sub>22</sub> =1 <sub>1</sub> 1 <sub>2</sub> 0 <sub>3</sub> 0 <sub>4</sub> 0 <sub>5</sub>	D <sub>20</sub> =1 <sub>1</sub> 1 <sub>2</sub> 0 <sub>3</sub> 1 <sub>4</sub> 1 <sub>5</sub>	OUT=1 <sub>1</sub> 0 <sub>2</sub> 0 <sub>3</sub> 0 <sub>4</sub> 0 <sub>5</sub>	2 ciclos
(d)	1	1	I <sub>5</sub> = D <sub>21</sub> (No hay FFs entre ambas señales)	D <sub>22</sub> =1 <sub>1</sub> 1 <sub>2</sub> 0 <sub>3</sub> 0 <sub>4</sub> 0 <sub>5</sub>	D <sub>20</sub> =1 <sub>1</sub> 0 <sub>2</sub> 1 <sub>3</sub> 1 <sub>4</sub> 1 <sub>5</sub>	OUT=1 <sub>1</sub> 0 <sub>2</sub> 0 <sub>3</sub> 0 <sub>4</sub> 0 <sub>5</sub>	1 ciclo

TABLA 10.4 Resultados de la etapa de salida del circuito lógico del Plano 5.

En los diagramas de simulación y de la Tabla 10.4 se observa que:

- i) La secuencia periódica aplicada a la entrada, en los cuatro casos es  $D_{21}=1_11_20_30_40_5$  y la secuencia modificada a la salida es  $OUT=1_10_20_30_40_5$ .
- ii) La secuencia de modificación que se aplica en cada caso es diferente debido al número de ciclos de reloj que debe avanzar  $D_{21}=1_2$  caso para ser modificado a  $OUT=0_2$ .
- iii) Las señales aplicadas a los terminales  $HDBn_{1\_D}$  y  $HDBn_{0\_D}$ , efectivamente permiten seleccionar el número de FFs que se colocan en cascada.
- v) El avance de la secuencia de entrada puede ser observado en las salidas de los FFs ( $F_1$ ,  $F_2$  y  $F_3$ ), la diferencia para cada caso radica en cual de ellas es reproducida a la salida del multiplexor  $I_5$ .
- vi) El número de ciclos de retardo entre las señales de entrada/salida es variable debido al número de FFs que se colocan en cascada, nótese que además de los FFs entre las señales  $D_{21}$  e  $I_5$  existe otro FF que se halla a la salida del multiplexor, la entrada a este FF es  $I_6=D_{20} \text{ AND } I_5$ .

En consecuencia, el circuito lógico de la etapa de salida es correcto puesto que cumple con los requerimientos propuestos en su formulación funcional.

### 10.2.3 Diseño a nivel físico

- a) Definición del plano de base y estrategias de enrutamiento global

De igual manera que el diseño del circuito Codificador, la síntesis del circuito lógico del Decodificador HDBn del Plano 6 ha sido realizada en bloques parciales, estos bloques son los mismos que se propusieron en el diseño funcional del Decodificador en el Capítulo 9.

La distribución e interconexión de los layouts parciales que se generen se plantea en el plano de base del circuito que se indica en la Fig.10.17.

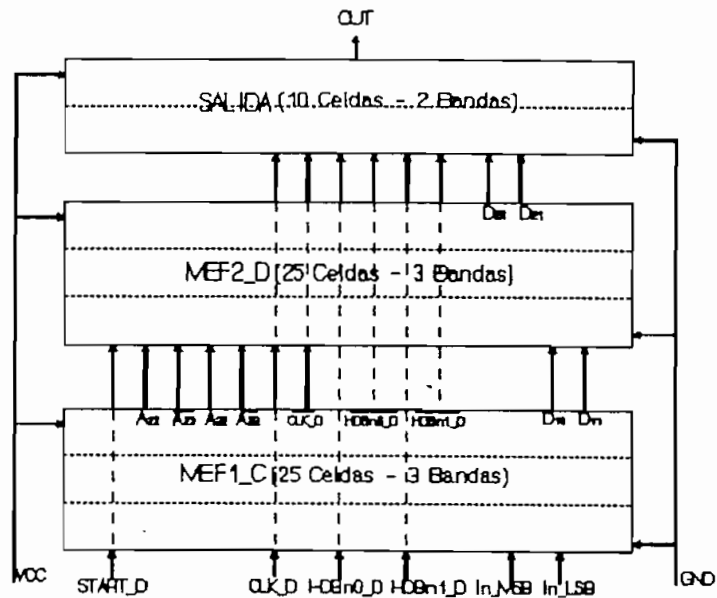


FIGURA 10.17 Plano de Base del circuito Decodificador HDBn

En el plano de base de la Fig.10.17 conviene notar que:

- i) El número de bandas de cada bloque ha sido propuesto en atención a la cantidad de celdas que posee, con el criterio de que las bandas que lo formen tengan largos similares.
  - ii) La orientación de los terminales de interfaz de cada bloque debe ser considerada al definirse las prioridades en la generación del layout de cada bloque.
  - iii) El plano de base corresponde únicamente al núcleo del layout en la parte que corresponde al Circuito Decodificador HDBn.
- b) Diseño de la Máquina de Estados Finitos 1' (MEF1\_D)

i) Netlist de descripción

La NETLIST SPICE que se describe al bloque MEF1\_D del Plano 5 debe considerar las siguientes restricciones de orientación de los terminales interfaz de acuerdo al plano de base de la Fig.10.17:

- i) Los terminales de entrada: START\_D, CLK\_D, HDBn1\_D, HDBn0\_D, IN\_MSB, IN\_LSB; deben tener orientación SUR.
- ii) Los terminales de salida: A23, ~A23, A22, ~A22, D10, D11, ~HDBn1\_D, ~HDBn0\_d y ~CLK\_D; deben tener orientación NORTE.

En consecuencia la NETLIST SPICE para la síntesis del bloque MEF1\_D es la siguiente:

```

**-----**
** RELATORIO SPICE           MEF1_D.SIM **
** CIRCUITO:  DECODIFICADOR HDBn      **
** BLOQUE:    MAQUINA DE ESTADOS FINITOS 1 **
**-----**

X1  CLK_D    ^CLK_D  vcc      INV
X2  HDBn1_D  ^HDBn1_D vcc      INV
X3  HDBn0_D  ^HDBn0_D vcc      INV
X4  A10      ^A10    vcc      INV
X5  D13      START_D P13     vcc      AND2
X6  D12      START_D P12     vcc      AND2
X7  ^HDBn0_D HDBn1_D P5      vcc      AND2
X8  P2       HDBn1_D P7      vcc      AND2
X9  P3       A12      P8      vcc      AND2
X10 P3       ^HDBn1_D P9      vcc      AND2
X11 P4       P5       P10     vcc      AND2
X12 ^A10     A11      ^HDBn0_D P1     vcc      AND3
X13 ^A10     ^A11     HDBn0_D P3     vcc      AND3
X14 ^A10     ^A11     ^A12    P4     vcc      AND3
X15 A10      HDBn0_D  ^HDBn1_D P6     vcc      AND3
X16 ^A10     A11      ^A12    HDBn0_D P2     vcc      AND4
X17 P3       P10     D12     vcc      OR2
X18 A23     A22     A10     vcc      OR2
X19 P1      P7      P8      D13     vcc      OR3
X20 P1      P8      P9      D11     vcc      OR3
X21 P2      P10     P6      D10     vcc      OR3
X22 IN_MSB  CLK_D    ^CLK_D  ^A23   A23     vcc      FFD_MS
X23 IN_LSB  CLK_D    ^CLK_D  ^A22   A22     vcc      FFD_MS
X24 P13     CLK_D    ^CLK_D  ^A12   A12     vcc      FFD_MS
X25 P12     CLK_D    ^CLK_D  ^A11   A11     vcc      FFD_MS

** TERMINALES DE INTERFAZ DEL MODULO **
** ----- **
# interface: ^CLK_D # orientacao=S #
# interface: ^HDBn1_D # orientacao=S #
# interface: ^HDBn0_D # orientacao=S #
# interface: A23 # orientacao=N #
# interface: ^A23 # orientacao=N #
# interface: A22 # orientacao=N #
# interface: ^A22 # orientacao=N #
# interface: D10 # orientacao=N #
# interface: D11 # orientacao=N #
# interface: START_D # orientacao=S #
# interface: CLK_D # orientacao=S #
# interface: HDBn1_D # orientacao=S #
# interface: HDBn0_D # orientacao=S #
# interface: IN_MSB # orientacao=S #
# interface: IN_LSB # orientacao=S #

```

.END

## ii) Parámetros de particionamiento de celdas y de polarización

Los parámetros para los procesos de particionamiento de celdas han sido definidos con los mismos criterios que fueron adoptados en los parámetros de los módulos del Codificador, así:

a:	Escolha:	Interfaz
b:	Número Regioes:	0
c:	Número Bandas:	2
d:	Relacao Aspecto:	--
e:	Percentual Balanco:	100%
f:	Fator de Forma:	1.000
g:	Acrescimo de Area:	0.000

Atendiendo a la orientación de las líneas de polarización en el plano de base de la Fig.10.17, los parámetros de enrutamiento son:

a:	Alimentacao Topo:	Vcc
b:	Alimentacao Esquerda:	Vcc

Es decir que la línea de Vcc tendrá orientación SUPERIOR-IZQUIERDA y la línea de tierra (Gnd) tendrá orientación: INFERIOR-DERECHA.

## iii) Distribución de celdas

De acuerdo a los parámetros anteriores, la distribución de celdas generada por los programas de particionamiento y posicionamiento es la que se indica en la Fig.10.18.

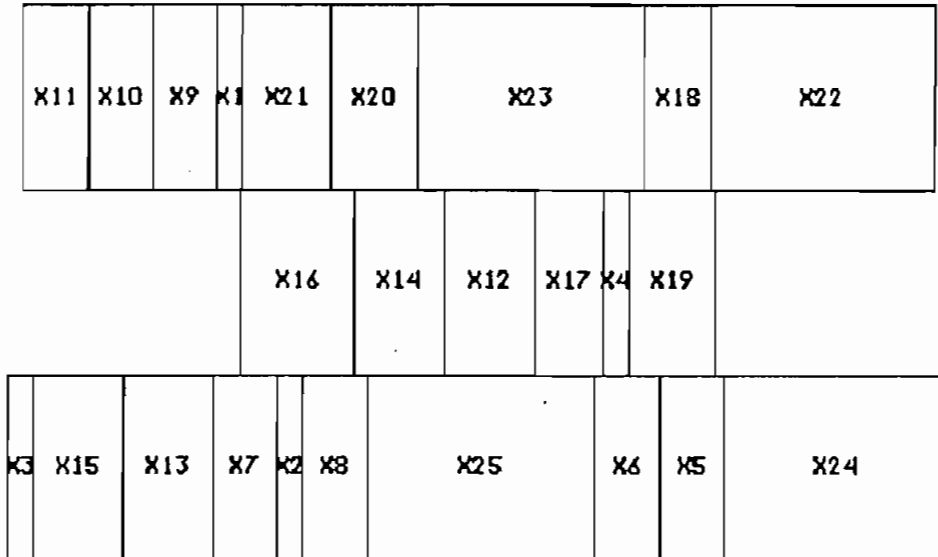


FIGURA 10.18 Distribución de celdas del bloque MEF1\_D.

La posición final de las celdas dentro de las bandas de la Fig.10.18 es el siguiente:

Posicionamiento do circuito C:\HDBn\MEF1\_D.pos

Banda 1

celula= X11 coordenada\_simbolica= 12 status rotacional=espelhada  
celula= X10 coordenada\_simbolica= 65 status rotacional=espelhada  
celula= X9 coordenada\_simbolica= 118 status rotacional=espelhada  
celula= X1 coordenada\_simbolica= 171 status rotacional=espelhada  
celula= X21 coordenada\_simbolica= 192 status rotacional=espelhada  
celula= X20 coordenada\_simbolica= 263 status rotacional=normal  
celula= X23 coordenada\_simbolica= 334 status rotacional=normal  
celula= X18 coordenada\_simbolica= 517 status rotacional=normal  
celula= X22 coordenada\_simbolica= 571 status rotacional=espelhada

Banda 2

celula= X16 coordenada\_simbolica= 190 status rotacional=normal  
celula= X14 coordenada\_simbolica= 283 status rotacional=normal  
celula= X12 coordenada\_simbolica= 356 status rotacional=normal  
celula= X17 coordenada\_simbolica= 429 status rotacional=normal  
celula= X4 coordenada\_simbolica= 483 status rotacional=normal  
celula= X19 coordenada\_simbolica= 504 status rotacional=normal

```

Banda 3
celula= X3 coordenada_simbolica= 0 status rotacional=espelhada
celula= X15 coordenada_simbolica= 21 status rotacional=espelhada
celula= X13 coordenada_simbolica= 94 status rotacional=espelhada
celula= X7 coordenada_simbolica= 167 status rotacional=espelhada
celula= X2 coordenada_simbolica= 220 status rotacional=normal
celula= X8 coordenada_simbolica= 241 status rotacional=normal
celula= X25 coordenada_simbolica= 294 status rotacional=espelhada
celula= X6 coordenada_simbolica= 477 status rotacional=espelhada
celula= X5 coordenada_simbolica= 530 status rotacional=normal
celula= X24 coordenada_simbolica= 583 status rotacional=normal

```

#### iv) Generación del layout

El layout del bloque MEF1\_D se indica en el Plano 6(a), su descripción en formato RS tiene la estructura jerárquica de celdas, bandas y estructura global que se indica a continuación:

```

DS 1 1 10;
9 "AND_2.CEL";
,
DF;
DS 2 1 10;
9 "AND_3.CEL";
,
DF;
DS 3 1 10;
9 "AND_4.CEL";
,
DF;
DS 4 1 10;
9 "FFD_MS.CEL";
,
DF;
DS 5 1 10;
9 "INV.CEL";
,
DF;
DS 6 1 10;
9 "OR_2.CEL";
,
DF;
DS 7 1 10;
9 "OR_3.CEL";
,
DF;

DS 23 1 10;
9 "BANDA 1";
C 1 T -113 0 MX;
C 1 T -178 0 MX;
C 1 T -243 0 MX;
C 5 T -312 0 MX;
C 7 T -395 0 MX;
C 7 T 395 0;
C 4 T 653 0;
C 6 T 848 0;
C 4 T -1243 0 MX;
9 "ENLACES ENTRE CELDAS"
,
DF;

DS 24 1 10;
9 "BANDA 2";
C 3 T 283 -106 MY;
C 2 T 472 -106 MY;
C 2 T 659 -106 MY;
C 6 T 744 -106 MY;
C 5 T 848 -106 MY;
C 7 T 881 -106 MY;
,
DF;

DS 25 1 10;
9 "BANDA 3";
C 5 T -129 0 MX;
C 2 T -226 0 MX;
C 2 T -323 0 MX;
C 1 T -388 0 MX;
C 5 T 472 0;
C 1 T 505 0;
C 4 T -812 0 MX;
C 1 T -901 0 MX;
C 1 T 901 0;
C 4 T 1012 0;
9 "ENLACES ENTRE CELDAS"
,
DF;

DS 26 1 10;
9 "hd-mq1.rs";
C 23 T 0 200;
C 24 T 0 100;
C 25;
9 "ENLACES ENTRE BANDAS"
,
DF;
C 26;
E

```



Las características generales del módulo generado y las líneas de enrutamiento no formadas (Redes Nao Roteadas) son:

Roteamento da Banda 1 do Circuito C:\HDBn\MEF1\_D  
Largura Banda=1219  
Redes Nao Roteadas:

Roteamento da Banda 2 do Circuito C:\HDBn\MEF1\_D  
Largura Banda=1219  
Redes Nao Roteadas:  
rede= IN\_LSB c\_vinc= X4 banda=2

Roteamento da Banda 3 do Circuito C:\HDBn\MEF1\_D  
Largura Banda=1219  
Redes Nao Roteadas:  
rede= ^A12 c\_vinc= X2 banda=3  
rede= A11 c\_vinc= X2 banda=3  
rede=HDBn0\_D c\_orig= X13 pino=2 banda=3

Informacoes do Modulo Gerado  
Area do Circuito = 387702  
Relacao de Aspecto = 0.242

En consecuencia en el layout del Plano 6(a) deben realizarse las siguientes modificaciones:

- iv.1) Depurar las interconexiones entre celdas a fin de optimizar el uso de los espacios de enrutamiento para construir los enlaces que los programas de síntesis y enrutamiento no generaron.
  
- iv.3) Corregir los errores característicos de los layouts generados por el sistema TENTOS, que fueron estudiados en el Capítulo 7.

El layout del bloque MEF1\_D en que se han realizado las modificaciones indicadas se indica en el Plano 7(a).

## v) Verificación funcional del layout

El circuito equivalente al layout del bloque MEF2\_D tiene la numeración de nodos y de transistores que se indica en el Plano 7(a), la NETLIST NDL que describe este circuito se indica a continuación, en ella se ha antecedido el caracter "X" a la numeración de los nodos y se ha cambiado los nodos "0" por "Gnd" y "1" por "Vdd", así:

```

% CIRCUITO: MEF1_D.nd1
% TECNOLOGIA: ECPD15
% DECODIFICADOR HD8n
% MAQUINA DE EST. FINITOS 1
%
% Transistores tipo NMOS: 116
% Transistores tipo PMOS: 116

INPUT x84 x106 x72 x123;
INPUT x103 x113 x87 x95;

PTRANS x126 Vdd x88 ;% Tr_1
NTRANS x126 Gnd x88 ;% Tr_2
PTRANS x88 Vdd x55 ;% Tr_3
NTRANS x88 x124 x56 ;% Tr_4
NTRANS x118 Gnd x56 ;% Tr_5
PTRANS x125 x55 x124 ;% Tr_6
PTRANS x118 Vdd x55 ;% Tr_7
NTRANS x125 Gnd x124 ;% Tr_8
PTRANS x118 Vdd x52 ;% Tr_9
PTRANS x119 Vdd x51 ;% Tr_10
NTRANS x124 Gnd x125 ;% Tr_11
NTRANS x119 x87 x54 ;% Tr_12
NTRANS x122 Gnd x54 ;% Tr_13
PTRANS x123 x51 x87 ;% Tr_14
NTRANS x126 x125 x53 ;% Tr_15
PTRANS x124 x52 x125 ;% Tr_16
NTRANS x118 Gnd x53 ;% Tr_17
NTRANS x123 Gnd x87 ;% Tr_18
PTRANS x126 Vdd x52 ;% Tr_19
PTRANS x122 Vdd x51 ;% Tr_20
NTRANS x87 Gnd x123 ;% Tr_21
PTRANS x125 Vdd x49 ;% Tr_22
NTRANS x125 x120 x50 ;% Tr_23
NTRANS x122 Gnd x50 ;% Tr_24
PTRANS x86 x49 x120 ;% Tr_25
PTRANS x122 Vdd x47 ;% Tr_26
PTRANS x122 Vdd x49 ;% Tr_27
PTRANS x87 x47 x123 ;% Tr_28

NTRANS x121 x123 x48 ;% Tr_29
NTRANS x122 Gnd x48 ;% Tr_30
NTRANS x86 Gnd x120 ;% Tr_31
PTRANS x121 Vdd x47 ;% Tr_32
NTRANS x120 Gnd x86 ;% Tr_33
PTRANS x122 Vdd x44 ;% Tr_34
PTRANS x116 Vdd x43 ;% Tr_35
NTRANS x118 Gnd x46 ;% Tr_36
NTRANS x116 x121 x46 ;% Tr_37
PTRANS x119 x43 x121 ;% Tr_38
NTRANS x124 x86 x45 ;% Tr_39
NTRANS x122 Gnd x45 ;% Tr_40
NTRANS x119 Gnd x121 ;% Tr_41
PTRANS x120 x44 x86 ;% Tr_42
PTRANS x124 Vdd x44 ;% Tr_43
PTRANS x118 Vdd x43 ;% Tr_44
NTRANS x121 Gnd x119 ;% Tr_45
PTRANS x118 Vdd x41 ;% Tr_46
PTRANS x121 x41 x119 ;% Tr_47
NTRANS x117 x119 x42 ;% Tr_48
NTRANS x118 Gnd x42 ;% Tr_49
PTRANS x117 Vdd x41 ;% Tr_50
NTRANS x116 Gnd x117 ;% Tr_51
PTRANS x116 Vdd x117 ;% Tr_52
NTRANS x115 Gnd x116 ;% Tr_53
NTRANS x60 Gnd x85 ;% Tr_54
PTRANS x115 Vdd x116 ;% Tr_55
PTRANS x60 Vdd x85 ;% Tr_56
PTRANS x85 Vdd x115 ;% Tr_57
NTRANS x113 x115 x40 ;% Tr_58
NTRANS x85 Gnd x40 ;% Tr_59
NTRANS x100 Gnd x60 ;% Tr_60
PTRANS x113 Vdd x115 ;% Tr_61
NTRANS x64 Gnd x60 ;% Tr_62
NTRANS x83 Gnd x84 ;% Tr_63
PTRANS x82 x60 x39 ;% Tr_64
PTRANS x100 Vdd x38 ;% Tr_65
PTRANS x64 x39 x38 ;% Tr_66
PTRANS x83 Vdd x84 ;% Tr_67

NTRANS x82 Gnd x60 ;% Tr_68
PTRANS x113 Vdd x111 ;% Tr_69
NTRANS x112 Gnd x37 ;% Tr_70
NTRANS x113 x111 x37 ;% Tr_71
PTRANS x112 Vdd x111 ;% Tr_72
NTRANS x120 Gnd x83 ;% Tr_73
PTRANS x120 x36 x83 ;% Tr_74
PTRANS x110 Vdd x36 ;% Tr_75
NTRANS x110 Gnd x83 ;% Tr_76
PTRANS x111 Vdd x81 ;% Tr_77
NTRANS x84 Gnd x102 ;% Tr_78
PTRANS x84 Vdd x102 ;% Tr_79
NTRANS x111 Gnd x81 ;% Tr_80
PTRANS x107 Vdd x35 ;% Tr_81
PTRANS x110 x35 x79 ;% Tr_82
NTRANS x122 Gnd x34 ;% Tr_83
NTRANS x107 x79 x34 ;% Tr_84
NTRANS x110 Gnd x79 ;% Tr_85
PTRANS x122 Vdd x35 ;% Tr_86
NTRANS x81 Gnd x80 ;% Tr_87
PTRANS x81 Vdd x80 ;% Tr_88
NTRANS x78 Gnd x112 ;% Tr_89
PTRANS x78 Vdd x112 ;% Tr_90
PTRANS x80 Vdd x32 ;% Tr_91
NTRANS x118 Gnd x33 ;% Tr_92
NTRANS x80 x108 x33 ;% Tr_93
PTRANS x76 x32 x108 ;% Tr_94
NTRANS x79 Gnd x110 ;% Tr_95
PTRANS x118 Vdd x32 ;% Tr_96
NTRANS x76 Gnd x108 ;% Tr_97
PTRANS x122 Vdd x30 ;% Tr_98
NTRANS x63 Gnd x78 ;% Tr_99
PTRANS x79 x30 x110 ;% Tr_100
NTRANS x77 x110 x31 ;% Tr_101
NTRANS x122 Gnd x31 ;% Tr_102
PTRANS x109 Vdd x29 ;% Tr_103
PTRANS x77 Vdd x30 ;% Tr_104
PTRANS x63 x78 x29 ;% Tr_105
NTRANS x109 Gnd x78 ;% Tr_106

```

```

PTRANS x118 Vdd x27 ;% Tr_107
NTRANS x108 Gnd x76 ;% Tr_108
NTRANS x118 Gnd x28 ;% Tr_109
PTRANS x108 x27 x76 ;% Tr_110
NTRANS x81 x76 x28 ;% Tr_111
PTRANS x73 Vdd x25 ;% Tr_112
PTRANS x81 Vdd x27 ;% Tr_113
NTRANS x73 x77 x26 ;% Tr_114
NTRANS x118 Gnd x26 ;% Tr_115
PTRANS x107 x25 x77 ;% Tr_116
NTRANS x107 Gnd x77 ;% Tr_117
NTRANS x75 Gnd x100 ;% Tr_118
PTRANS x75 Vdd x100 ;% Tr_119
PTRANS x118 Vdd x25 ;% Tr_120
NTRANS x106 x22 x75 ;% Tr_121
PTRANS x106 Vdd x25 ;% Tr_122
PTRANS x76 Vdd x23 ;% Tr_123
NTRANS x76 x106 x24 ;% Tr_124
NTRANS x122 Gnd x24 ;% Tr_125
PTRANS x72 x23 x106 ;% Tr_126
NTRANS x77 Gnd x107 ;% Tr_127
PTRANS x122 Vdd x23 ;% Tr_128
NTRANS x72 Gnd x106 ;% Tr_129
PTRANS x118 Vdd x20 ;% Tr_130
PTRANS x77 x20 x107 ;% Tr_131
NTRANS x98 x22 x19 ;% Tr_132
NTRANS x74 x107 x21 ;% Tr_133
NTRANS x118 Gnd x21 ;% Tr_134
PTRANS x98 Vdd x75 ;% Tr_135
PTRANS x74 Vdd x20 ;% Tr_136
NTRANS x102 Gnd x19 ;% Tr_137
PTRANS x102 Vdd x75 ;% Tr_138
NTRANS x73 Gnd x74 ;% Tr_139
PTRANS x73 Vdd x74 ;% Tr_140
NTRANS x106 Gnd x72 ;% Tr_141
PTRANS x122 Vdd x17 ;% Tr_142
NTRANS x122 Gnd x18 ;% Tr_143
NTRANS x108 x72 x18 ;% Tr_144
PTRANS x106 x17 x72 ;% Tr_145
PTRANS x108 Vdd x17 ;% Tr_146
NTRANS x105 Gnd x82 ;% Tr_147
PTRANS x105 Vdd x82 ;% Tr_148
NTRANS x104 Gnd x71 ;% Tr_149
PTRANS x104 Vdd x71 ;% Tr_150
PTRANS x69 Vdd x105 ;% Tr_151
NTRANS x103 x105 x16 ;% Tr_152
NTRANS x69 Gnd x16 ;% Tr_153
NTRANS x87 x15 x104 ;% Tr_154
PTRANS x87 Vdd x104 ;% Tr_155
PTRANS x103 Vdd x105 ;% Tr_156
NTRANS x72 x15 x14 ;% Tr_157
PTRANS x72 Vdd x104 ;% Tr_158
NTRANS x102 Gnd x14 ;% Tr_159
NTRANS x103 Gnd x92 ;% Tr_160
PTRANS x103 Vdd x92 ;% Tr_161
PTRANS x102 Vdd x104 ;% Tr_162
NTRANS x59 Gnd x70 ;% Tr_163
PTRANS x59 Vdd x70 ;% Tr_164
NTRANS x100 Gnd x59 ;% Tr_165
NTRANS x64 Gnd x59 ;% Tr_166
PTRANS x100 Vdd x13 ;% Tr_167
PTRANS x64 x13 x12 ;% Tr_168
PTRANS x99 x12 x59 ;% Tr_169
NTRANS x99 Gnd x59 ;% Tr_170
NTRANS x69 Gnd x58 ;% Tr_171
PTRANS x97 Vdd x11 ;% Tr_172
PTRANS x63 x11 x10 ;% Tr_173
PTRANS x69 x10 x58 ;% Tr_174
NTRANS x63 Gnd x58 ;% Tr_175
PTRANS x103 Vdd x96 ;% Tr_176
NTRANS x98 Gnd x9 ;% Tr_177
NTRANS x103 x96 x9 ;% Tr_178
NTRANS x68 Gnd x69 ;% Tr_179
PTRANS x68 Vdd x69 ;% Tr_180
PTRANS x98 Vdd x96 ;% Tr_181
NTRANS x97 Gnd x58 ;% Tr_182
PTRANS x96 Vdd x90 ;% Tr_183
NTRANS x96 Gnd x90 ;% Tr_184
PTRANS x102 Vdd x68 ;% Tr_185
NTRANS x102 x8 x68 ;% Tr_186
NTRANS x58 Gnd x66 ;% Tr_187
PTRANS x106 Vdd x68 ;% Tr_188
PTRANS x58 Vdd x66 ;% Tr_189
NTRANS x106 x8 x65 ;% Tr_190
NTRANS x102 Gnd x7 ;% Tr_191
PTRANS x102 Vdd x94 ;% Tr_192
PTRANS x87 Vdd x68 ;% Tr_193
NTRANS x118 Gnd x122 ;% Tr_194
NTRANS x95 Gnd x6 ;% Tr_195
NTRANS x72 x7 x5 ;% Tr_196
PTRANS x72 Vdd x94 ;% Tr_197
NTRANS x87 x6 x65 ;% Tr_198
PTRANS x118 Vdd x122 ;% Tr_199
PTRANS x95 Vdd x68 ;% Tr_200
NTRANS x95 x94 x5 ;% Tr_201
PTRANS x95 Vdd x94 ;% Tr_202
NTRANS x94 Gnd x109 ;% Tr_203
PTRANS x94 Vdd x109 ;% Tr_204
PTRANS x123 Vdd x93 ;% Tr_205
NTRANS x109 Gnd x4 ;% Tr_206
NTRANS x123 x93 x4 ;% Tr_207
PTRANS x109 Vdd x93 ;% Tr_208
NTRANS x92 Gnd x3 ;% Tr_209
PTRANS x92 Vdd x57 ;% Tr_210
PTRANS x84 Vdd x57 ;% Tr_211
NTRANS x84 x3 x2 ;% Tr_212
PTRANS x93 Vdd x64 ;% Tr_213
NTRANS x93 Gnd x64 ;% Tr_214
NTRANS x95 x57 x2 ;% Tr_215
PTRANS x95 Vdd x57 ;% Tr_216
PTRANS x109 Vdd x91 ;% Tr_217
NTRANS x57 Gnd x97 ;% Tr_218
NTRANS x92 Gnd x61 ;% Tr_219
NTRANS x109 x91 x61 ;% Tr_220
PTRANS x57 Vdd x97 ;% Tr_221
PTRANS x92 Vdd x91 ;% Tr_222
PTRANS x91 Vdd x99 ;% Tr_223
NTRANS x91 Gnd x99 ;% Tr_224
NTRANS x95 Gnd x98 ;% Tr_225
PTRANS x95 Vdd x98 ;% Tr_226
PTRANS x71 Vdd x89 ;% Tr_227
NTRANS x71 x89 x62 ;% Tr_228
NTRANS x90 Gnd x62 ;% Tr_229
PTRANS x90 Vdd x89 ;% Tr_230
PTRANS x89 Vdd x63 ;% Tr_231
NTRANS x89 Gnd x63 ;% Tr_232
END

```

Para simular el circuito equivalente del layout con los mismos criterios empleados en la simulación del circuito lógico, ha sido necesario identificar la correspondencia entre la numeración de los nodos del layout del Plano 7(a) y

su nominación en el circuito del Plano 5, así para los nodos terminales del Circuito Combinacional 1 (CC1) se tiene:

NODOS DEL CIRCUITO LOGICO	NODOS DEL LAYOUT
HDBn1_D (entrada)	x103
HDBn0_D (entrada)	x95
A12 (entrada)	x123
A11 (entrada)	x106
A10 (entrada)	x84
D13 (salida)	x85
D12 (salida)	x112
D11 (salida)	x70
D10 (salida)	x66

Los nodos de entrada del layout han sido excitados con las mismas combinaciones lógicas que fueron aplicadas a los terminales del circuito lógico (Fig.10.14) los resultados obtenidos se indican en la Fig.10.19.

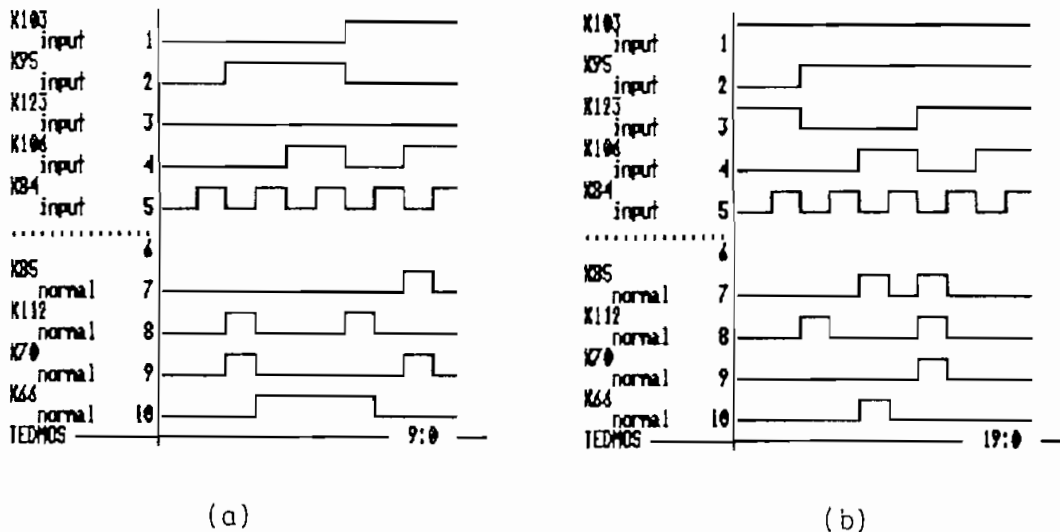


FIGURA 10.19 Simulación lógica del layout del CC1 del Plano 7(a).

Los valores que adoptan los terminales de salida en la Fig.10.19 concuerdan con sus correspondientes en la simulación del circuito lógico de la Fig.10.14 y con la descripción funcional de la Tabla 10.5 por lo que los diseños de este bloque a los niveles físico, lógico y funcional son equivalentes.

c) Diseño de la Máquina de Estados Finitos 2 (MEF2\_D)

i) Netlist de descripción

De acuerdo al plano de base de la Fig,10,17, los terminales de interfaz del bloque MEF2\_D deben atenerse a las siguientes restricciones de orientación:

- i) Los terminales de entrada: START\_D, CLK\_D, ~CLK\_D, D11, D10, A23, ~A23, A22, y ~A22 deben tener orientación SUR.
- ii) Los terminales de salida: D20 y D21 deben tener orientación NORTE.

Por tanto la NETLIST para la descripción del bloque MEF2\_D del Plano 5 es la siguiente:

```

**-----**
** RELATORIO SPICE           MEF2_D.SIM  **
** CIRCUITO:  DECODIFICADOR HDBn        **
** BLOQUE:    MAQUINA DE ESTADOS FINITOS 2  **
**-----**
X30 A23  ^A24  Q1    vcc    AND2
X31 A21   A23   Q5    vcc    AND2
X32 A22   A24   Q7    vcc    AND2
X33 ^A22  ^A24   Q9    vcc    AND2
X34 ^A23  A24   Q10   vcc    AND2
X35 Q2    ^A24  Q11   vcc    AND2
X36 Q3    ^A24  Q12   vcc    AND2

```

```

X37 Q19    START_D D22    vcc    AND2
X38 START_D Q15    D21    vcc    AND2
X39 START_D Q14    D20    vcc    AND2
X40 ^A20    ^A21    A22    Q2     vcc    AND3
X41 A21     ^A22    ^A23    Q3     vcc    AND3
X42 A20     ^A22    A24    Q4     vcc    AND3
X43 ^A20     ^A21    A23    Q8     vcc    AND3
X44 ^A21     ^A22    ^A23    A24    Q6     vcc    AND4
X45 Q7      Q8      Q13    vcc    OR2
X46 Q16    Q13    Q15    vcc    OR2
X47 Q17    Q18    Q19    vcc    OR2
X48 Q1     Q11    Q12    Q17    vcc    OR3
X49 Q4     Q5     Q6     Q18    vcc    OR3
X50 Q1     Q2     Q3     Q16    vcc    OR3
X51 ^A20    Q9     Q10    Q14    vcc    OR3
X52 D22    CLK_D   ^CLK_D ^A24    A24    vcc    FFD_MS
X53 D11    CLK_D   ^CLK_D ^A21    A21    vcc    FFD_MS
X54 D10    CLK_D   ^CLK_D ^A20    A20    vcc    FFD_MS

```

```

** TERMINALES DE INTERFAZ DEL MODULO **

```

```

** ----- **

```

```

‡ interface: START_D ‡ orientacao=S ‡
‡ interface: CLK_D   ‡ orientacao=S ‡
‡ interface: ^CLK_D  ‡ orientacao=S ‡
‡ interface: D11     ‡ orientacao=S ‡
‡ interface: D10     ‡ orientacao=S ‡
‡ interface: A23     ‡ orientacao=S ‡
‡ interface: ^A23    ‡ orientacao=S ‡
‡ interface: A22     ‡ orientacao=S ‡
‡ interface: ^A22    ‡ orientacao=S ‡
‡ interface: D20     ‡ orientacao=N ‡
‡ interface: D21     ‡ orientacao=N ‡

```

```

.END

```

## ii) Parámetros de particionamiento de celdas y de polarización

Los parámetros para el particionamiento de celdas y orientación de los caminos de polarización son los mismos que se definieron para los otros bloques, a excepción del número de bandas que de acuerdo al plano de base (Fig.10.17) ha sido fijado como:

c: Número de bandas: 3

debido a la cantidad de celdas que posee este bloque.

### iii) Distribución de Celdas

De acuerdo a los parámetros definidos la distribución de celdas generada por los programas de particionamiento y posicionamiento del sistema TENTOS es la que se indica en la Fig.10.20.

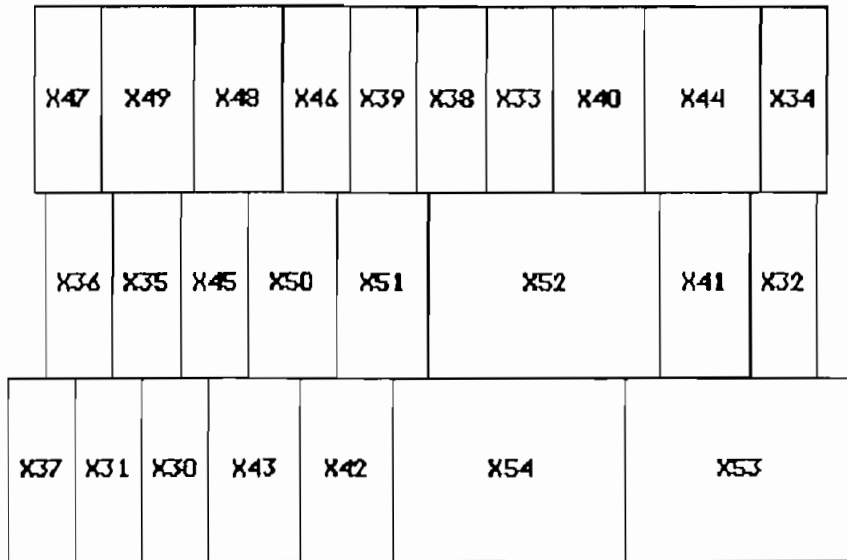


FIGURA 10.20 Distribución de celdas en el bloque MEF2\_D

La posición final de las celdas dentro de sus respectivas bandas es el siguiente:

Posicionamiento do circuito C:\HDBn\MEF2\_D.pos

Banda 1

```

celula= X47 coordenada_simbolica= 21 status rotacional=normal
celula= X49 coordenada_simbolica= 75 status rotacional=normal
celula= X48 coordenada_simbolica= 146 status rotacional=normal
celula= X46 coordenada_simbolica= 217 status rotacional=normal
celula= X39 coordenada_simbolica= 271 status rotacional=normal
celula= X38 coordenada_simbolica= 324 status rotacional=normal
celula= X33 coordenada_simbolica= 377 status rotacional=espelhada
celula= X40 coordenada_simbolica= 430 status rotacional=espelhada
celula= X44 coordenada_simbolica= 503 status rotacional=espelhada
celula= X34 coordenada_simbolica= 596 status rotacional=normal
    
```

Banda 2

```

celula= X36 coordenada_simbolica= 30 status rotacional=normal
celula= X35 coordenada_simbolica= 83 status rotacional=normal
celula= X45 coordenada_simbolica= 136 status rotacional=normal
    
```

```

celula= X50 coordenada_simbolica= 190 status rotacional=normal
celula= X51 coordenada_simbolica= 261 status rotacional=normal
celula= X52 coordenada_simbolica= 332 status rotacional=normal
celula= X41 coordenada_simbolica= 515 status rotacional=normal
celula= X32 coordenada_simbolica= 588 status rotacional=normal

```

Banda 3

```

celula= X37 coordenada_simbolica= 0 status rotacional=espelhada
celula= X31 coordenada_simbolica= 53 status rotacional=espelhada
celula= X30 coordenada_simbolica= 106 status rotacional=normal
celula= X43 coordenada_simbolica= 159 status rotacional=normal
celula= X42 coordenada_simbolica= 232 status rotacional=espelhada
celula= X54 coordenada_simbolica= 305 status rotacional=normal
celula= X53 coordenada_simbolica= 488 status rotacional=espelhada

```

#### iv) Generación del layout

El layout generado por programas de síntesis del sistema TENTOS en función de la distribución de celdas de la Fig.10.20 es el que se indica en el Plano 6(b), su descripción jerárquica en formato RS tiene la siguiente estructura:

```

DS 1 1 10;
9 "AND_2.CEL";
.
DF;
DS 2 1 10;
9 "AND_3.CEL";
.
DF;
DS 3 1 10;
9 "AND_4.CEL";
.
DF;
DS 4 1 10;
9 "FFD_MS.CEL";
.
DF;
DS 5 1 10;
9 "OR_2.CEL";
.
DF;
DS 6 1 10;
9 "OR_3.CEL";
.
DF;

DS 23 1 10;
9 "BANDA 1";
C 5 T 24 0;
C 6 T 90 0;
C 6 T 353 0;
C 5 T 472 0;
C 1 T 634 0;
C 1 T 699 0;
C 1 T -829 0 MX;
C 2 T -914 0 MX;
C 3 T -1131 0 MX;
C 1 T 1155 0;
9 "ENLACES ENTRE CELDAS";
.
DF;
DS 24 1 10;
9 "BANDA 2";
C 1 T 55 -106 MY;
C 1 T 120 -106 MY;
C 5 T 365 -106 MY;
C 6 T 460 -106 MY;
C 6 T 646 -106 MY;
C 4 T 729 -106 MY;
C 2 T 950 -106 MY;
C 1 T 1155 -106 MY;
9 "ENLACES ENTRE CELDAS";
.
DF;

DS 25 1 10;
9 "BANDA 3";
C 1 T -197 0 MX;
C 1 T -281 0 MX;
C 1 T 351 0;
C 2 T 416 0;
C 2 T -586 0 MX;
C 4 T 645 0;
C 4 T -1035 0 MX;
9 "ENLACES ENTRE CELDAS";
.
DF;
DS 26 1 10;
9 "LAYOUT GLOBAL";
C 23 T 0 200;
C 24 T 0 100;
C 25;
9 "ENLACES ENTRE BANDAS";
.
DF;
C 26;
E

```



Las características generales de las bandas y las redes no formadas (Redes Nao Enrutadas) son las siguientes:

```
Roteamento da Banda 1 do Circuito C:\HDBn\MEF2_D
Largura Banda=1196
Redes Nao Roteadas:
rede= Q13 c_yinc= X33 banda=1
rede= Q14 c_yinc= X33 banda=1
rede=START_D c_yinc= X33 banda=1
rede=START_D c_orig= X39 pino=1 banda=1
```

```
Roteamento da Banda 2 do Circuito C:\HDBn\MEF2_D
Largura Banda=1196
Redes Nao Roteadas:F
rede= ^A22 c_orig= X41 pino=2 banda=2
```

```
Roteamento da Banda 3 do Circuito C:\HDBn\MEF2_D
Largura Banda=1196
Redes Nao Roteadas:
rede= ^A20 c_orig= X43 pino=2 banda=3
rede= ^A21 c_orig= X43 pino=3 banda=3
rede= A24 c_orig= X42 pino=3 banda=3
rede= A23 c_orig= X30 pino=1 banda=3
rede= ^A22 c_orig= X42 pino=4 banda=3
```

```
Informacoes do Modulo Gerado
Area do Circuito = 380664
Relacao de Aspecto = 0.246
```

El Plano 7(b) muestra el layout del bloque MEF2\_D en que se han optimizado las interconexiones del Plano 6 (b) y se han construido las interconexiones faltantes, además se han corregido los errores característicos de los layouts generados por el sistema TENTOS.

#### v) Verificación funcional del layout

El circuito equivalente del layout del módulo MEF2\_D tiene la numeración de nodos y transistores que se indica en el plano 7(b), la NETLIST NDL que describe este circuito es la siguiente (se ha antepuesto la letra "y" a la numeración

de nodos y se han cambiado los nodos "1" por "Vdd" y "0" por "gnd" con respecto a la nomenclatura del plano 7(b))

```

% CIRCUITO: MEF2_D.nd1
% TECNOLOGIA: ECPD15
% DECODIFICADOR HD8n
% MAQUINA DE EST. FINITOS 2
%
% Transistores tipo NMOS: 115
% Transistores tipo PMOS: 115

INPUT y116 y119 y120 y125 y92;
INPUT y126 y104 y95 y127 y91;

NTRANS y129 Gnd y96 ;X Tr_1
NTRANS y128 Gnd y58 ;X Tr_2
PTRANS y129 Vdd y96 ;X Tr_3
PTRANS y128 Vdd y58 ;X Tr_4
PTRANS y95 Vdd y129 ;X Tr_5
PTRANS y92 Vdd y128 ;X Tr_6
NTRANS y127 y53 y128 ;X Tr_7
NTRANS y127 y129 y54 ;X Tr_8
NTRANS y95 Gnd y54 ;X Tr_9
NTRANS y92 Gnd y53 ;X Tr_10
PTRANS y127 Vdd y129 ;X Tr_11
PTRANS y127 Vdd y128 ;X Tr_12
PTRANS y127 Vdd y124 ;X Tr_13
NTRANS y126 y61 y52 ;X Tr_14
NTRANS y127 Gnd y52 ;X Tr_15
PTRANS y126 Vdd y124 ;X Tr_16
NTRANS y95 y61 y51 ;X Tr_17
PTRANS y95 Vdd y124 ;X Tr_18
PTRANS y125 Vdd y124 ;X Tr_19
NTRANS y125 y124 y51 ;X Tr_20
NTRANS y124 Gnd y100 ;X Tr_21
PTRANS y124 Vdd y100 ;X Tr_22
PTRANS y121 Vdd y94 ;X Tr_23
NTRANS y121 Gnd y94 ;X Tr_24
NTRANS y93 Gnd y98 ;X Tr_25
PTRANS y93 Vdd y98 ;X Tr_26
PTRANS y94 Vdd y49 ;X Tr_27
NTRANS y94 y122 y50 ;X Tr_28
NTRANS y115 Gnd y50 ;X Tr_29
PTRANS y123 y49 y122 ;X Tr_30
NTRANS y95 y48 y93 ;X Tr_31
PTRANS y95 Vdd y93 ;X Tr_32
PTRANS y115 Vdd y49 ;X Tr_33
NTRANS y123 Gnd y122 ;X Tr_34
NTRANS y126 y48 y47 ;X Tr_35
PTRANS y126 Vdd y93 ;X Tr_36
PTRANS y115 Vdd y45 ;X Tr_37
NTRANS y120 Gnd y47 ;X Tr_38
NTRANS y122 Gnd y123 ;X Tr_39

PTRANS y120 Vdd y93 ;X Tr_40
NTRANS y115 Gnd y46 ;X Tr_41
PTRANS y122 y45 y123 ;X Tr_42
NTRANS y121 y123 y46 ;X Tr_43
PTRANS y121 Vdd y45 ;X Tr_44
PTRANS y123 Vdd y43 ;X Tr_45
NTRANS y60 Gnd y44 ;X Tr_46
NTRANS y123 y120 y44 ;X Tr_47
PTRANS y125 y43 y120 ;X Tr_48
PTRANS y60 Vdd y43 ;X Tr_49
NTRANS y125 Gnd y120 ;X Tr_50
PTRANS y125 Vdd y90 ;X Tr_51
NTRANS y125 Gnd y41 ;X Tr_52
PTRANS y117 Vdd y42 ;X Tr_53
NTRANS y92 y41 y37 ;X Tr_54
PTRANS y92 Vdd y90 ;X Tr_55
PTRANS y127 y91 y42 ;X Tr_56
NTRANS y60 Gnd y40 ;X Tr_57
NTRANS y117 y40 y91 ;X Tr_58
NTRANS y120 Gnd y125 ;X Tr_59
PTRANS y60 Vdd y38 ;X Tr_60
NTRANS y127 Gnd y91 ;X Tr_61
PTRANS y60 Vdd y42 ;X Tr_62
NTRANS y60 Gnd y39 ;X Tr_63
NTRANS y122 y125 y39 ;X Tr_64
PTRANS y120 y38 y125 ;X Tr_65
PTRANS y122 Vdd y38 ;X Tr_66
NTRANS y119 y90 y37 ;X Tr_67
PTRANS y119 Vdd y90 ;X Tr_68
NTRANS y91 Gnd y127 ;X Tr_69
NTRANS y90 Gnd y89 ;X Tr_70
PTRANS y90 Vdd y89 ;X Tr_71
PTRANS y60 Vdd y35 ;X Tr_72
PTRANS y91 y127 y35 ;X Tr_73
NTRANS y60 Gnd y36 ;X Tr_74
NTRANS y118 y36 y127 ;X Tr_75
PTRANS y118 Vdd y35 ;X Tr_76
PTRANS y126 Vdd y114 ;X Tr_77
NTRANS y91 Gnd y34 ;X Tr_78
NTRANS y126 y114 y34 ;X Tr_79
PTRANS y112 Vdd y33 ;X Tr_80
PTRANS y111 Vdd y32 ;X Tr_81
PTRANS y91 Vdd y114 ;X Tr_82
NTRANS y115 Gnd y31 ;X Tr_83
NTRANS y112 y31 y118 ;X Tr_84
NTRANS y60 Gnd y30 ;X Tr_85
NTRANS y111 y119 y30 ;X Tr_86
PTRANS y117 y118 y33 ;X Tr_87
PTRANS y116 y32 y119 ;X Tr_88
NTRANS y117 Gnd y118 ;X Tr_89

NTRANS y116 Gnd y119 ;X Tr_90
PTRANS y60 Vdd y32 ;X Tr_91
PTRANS y115 Vdd y33 ;X Tr_92
PTRANS y114 Vdd y88 ;X Tr_93
NTRANS y114 Gnd y88 ;X Tr_94
NTRANS y118 Gnd y117 ;X Tr_95
NTRANS y119 Gnd y116 ;X Tr_96
PTRANS y115 Vdd y27 ;X Tr_97
PTRANS y60 Vdd y26 ;X Tr_98
PTRANS y118 y117 y27 ;X Tr_99
NTRANS y115 Gnd y29 ;X Tr_100
NTRANS y86 y29 y117 ;X Tr_101
PTRANS y119 y26 y116 ;X Tr_102
NTRANS y85 y116 y28 ;X Tr_103
NTRANS y60 Gnd y28 ;X Tr_104
PTRANS y86 Vdd y27 ;X Tr_105
PTRANS y85 Vdd y26 ;X Tr_106
NTRANS y113 Gnd y87 ;X Tr_107
NTRANS y112 Gnd y86 ;X Tr_108
PTRANS y113 Vdd y87 ;X Tr_109
PTRANS y112 Vdd y86 ;X Tr_110
PTRANS y59 Vdd y25 ;X Tr_111
NTRANS y59 y85 y24 ;X Tr_112
NTRANS y115 Gnd y24 ;X Tr_113
PTRANS y68 Vdd y113 ;X Tr_114
PTRANS y111 y25 y85 ;X Tr_115
NTRANS y81 y113 y23 ;X Tr_116
NTRANS y68 Gnd y23 ;X Tr_117
NTRANS y111 Gnd y85 ;X Tr_118
PTRANS y115 Vdd y25 ;X Tr_119
PTRANS y81 Vdd y113 ;X Tr_120
NTRANS y82 Gnd y108 ;X Tr_121
PTRANS y82 Vdd y108 ;X Tr_122
NTRANS y85 Gnd y111 ;X Tr_123
NTRANS y96 Gnd y82 ;X Tr_124
NTRANS y109 Gnd y84 ;X Tr_125
PTRANS y115 Vdd y21 ;X Tr_126
PTRANS y109 Vdd y84 ;X Tr_127
PTRANS y85 y21 y111 ;X Tr_128
NTRANS y83 y111 y22 ;X Tr_129
NTRANS y115 Gnd y22 ;X Tr_130
PTRANS y83 Vdd y21 ;X Tr_131
NTRANS y88 Gnd y82 ;X Tr_132
PTRANS y96 Vdd y19 ;X Tr_133
PTRANS y88 y20 y19 ;X Tr_134
PTRANS y119 y82 y20 ;X Tr_135
NTRANS y59 Gnd y83 ;X Tr_136
NTRANS y119 Gnd y82 ;X Tr_137
PTRANS y59 Vdd y83 ;X Tr_138
PTRANS y68 Vdd y109 ;X Tr_139

```

NTRANS	y108	y109	y18	;Z	Tr_140	NTRANS	y119	Gnd	y11	;Z	Tr_171	PTRANS	y68	Vdd	y101	;Z	Tr_202
NTRANS	y68	Gnd	y18	;Z	Tr_141	PTRANS	y119	Vdd	y75	;Z	Tr_172	NTRANS	y102	Gnd	y69	;Z	Tr_203
PTRANS	y108	Vdd	y109	;Z	Tr_142	NTRANS	y57	Gnd	y56	;Z	Tr_173	PTRANS	y102	Vdd	y69	;Z	Tr_204
PTRANS	y127	Vdd	y80	;Z	Tr_143	PTRANS	y57	Vdd	y56	;Z	Tr_174	PTRANS	y101	Vdd	y112	;Z	Tr_205
NTRANS	y127	Gnd	y17	;Z	Tr_144	NTRANS	y106	Gnd	y73	;Z	Tr_175	NTRANS	y66	Gnd	y97	;Z	Tr_206
NTRANS	y116	y17	y16	;Z	Tr_145	PTRANS	y106	Vdd	y73	;Z	Tr_176	PTRANS	y66	Vdd	y97	;Z	Tr_207
PTRANS	y116	Vdd	y80	;Z	Tr_146	NTRANS	y105	Gnd	y74	;Z	Tr_177	NTRANS	y101	Gnd	y112	;Z	Tr_208
NTRANS	y126	y80	y16	;Z	Tr_147	PTRANS	y105	Vdd	y74	;Z	Tr_178	PTRANS	y89	Vdd	y102	;Z	Tr_209
PTRANS	y126	Vdd	y80	;Z	Tr_148	NTRANS	y74	Gnd	y57	;Z	Tr_179	NTRANS	y91	y4	y102	;Z	Tr_210
NTRANS	y76	Gnd	y79	;Z	Tr_149	NTRANS	y58	Gnd	y106	;Z	Tr_180	NTRANS	y89	Gnd	y4	;Z	Tr_211
PTRANS	y76	Vdd	y79	;Z	Tr_150	PTRANS	y58	y106	y10	;Z	Tr_181	PTRANS	y91	Vdd	y102	;Z	Tr_212
NTRANS	y80	Gnd	y78	;Z	Tr_151	PTRANS	y71	Vdd	y10	;Z	Tr_182	NTRANS	y100	Gnd	y66	;Z	Tr_213
NTRANS	y107	Gnd	y81	;Z	Tr_152	NTRANS	y71	Gnd	y106	;Z	Tr_183	NTRANS	y78	Gnd	y66	;Z	Tr_214
PTRANS	y80	Vdd	y78	;Z	Tr_153	NTRANS	y69	Gnd	y57	;Z	Tr_184	PTRANS	y100	Vdd	y3	;Z	Tr_215
PTRANS	y107	Vdd	y81	;Z	Tr_154	PTRANS	y91	Vdd	y105	;Z	Tr_185	PTRANS	y78	y3	y2	;Z	Tr_216
NTRANS	y98	Gnd	y76	;Z	Tr_155	NTRANS	y104	y105	y9	;Z	Tr_186	PTRANS	y65	y2	y66	;Z	Tr_217
NTRANS	y79	Gnd	y107	;Z	Tr_156	NTRANS	y91	Gnd	y9	;Z	Tr_187	NTRANS	y99	Gnd	y70	;Z	Tr_218
PTRANS	y73	Vdd	y15	;Z	Tr_157	PTRANS	y70	y7	y57	;Z	Tr_188	NTRANS	y65	Gnd	y66	;Z	Tr_219
PTRANS	y79	y15	y107	;Z	Tr_158	PTRANS	y69	y8	y7	;Z	Tr_189	PTRANS	y99	Vdd	y70	;Z	Tr_220
NTRANS	y73	Gnd	y107	;Z	Tr_159	PTRANS	y74	Vdd	y8	;Z	Tr_190	PTRANS	y91	Vdd	y99	;Z	Tr_221
NTRANS	y75	Gnd	y71	;Z	Tr_160	NTRANS	y70	Gnd	y57	;Z	Tr_191	NTRANS	y98	y62	y99	;Z	Tr_222
PTRANS	y75	Vdd	y71	;Z	Tr_161	PTRANS	y104	Vdd	y105	;Z	Tr_192	NTRANS	y91	Gnd	y62	;Z	Tr_223
NTRANS	y89	Gnd	y76	;Z	Tr_162	PTRANS	y120	Vdd	y103	;Z	Tr_193	NTRANS	y55	Gnd	y64	;Z	Tr_224
PTRANS	y98	Vdd	y13	;Z	Tr_163	NTRANS	y104	Gnd	y6	;Z	Tr_194	PTRANS	y55	Vdd	y64	;Z	Tr_225
PTRANS	y89	y14	y13	;Z	Tr_164	NTRANS	y120	y103	y6	;Z	Tr_195	PTRANS	y98	Vdd	y99	;Z	Tr_226
PTRANS	y74	y76	y14	;Z	Tr_165	PTRANS	y104	Vdd	y103	;Z	Tr_196	NTRANS	y56	Gnd	y55	;Z	Tr_227
NTRANS	y74	Gnd	y76	;Z	Tr_166	PTRANS	y103	Vdd	y65	;Z	Tr_197	PTRANS	y97	Vdd	y63	;Z	Tr_228
NTRANS	y125	y75	y12	;Z	Tr_167	NTRANS	y103	Gnd	y65	;Z	Tr_198	PTRANS	y56	y63	y55	;Z	Tr_229
PTRANS	y125	Vdd	y75	;Z	Tr_168	PTRANS	y64	Vdd	y101	;Z	Tr_199	NTRANS	y97	Gnd	y55	;Z	Tr_230
NTRANS	y104	y11	y12	;Z	Tr_169	NTRANS	y64	y101	y5	;Z	Tr_200	END					
PTRANS	y104	Vdd	y75	;Z	Tr_170	NTRANS	y68	Gnd	y5	;Z	Tr_201						

La correspondencia entre la nominación de los nodos terminales del Circuito Combinacional (CC2) del la Máquina de Estados Finitos 2 en el circuito lógico del Plano 5 y su numeración en el layout del Plano 7(b) es la siguiente:

NODOS DEL CIRCUITO LOGICO	NODOS DEL LAYOUT
A24 (entrada)	y127
A23 (entrada)	y104
A22 (entrada)	y92
A21 (entrada)	y120
A20 (entrada)	y116
D22 (salida)	y112
D21 (salida)	y87
D20 (salida)	y84

estos nodos han sido excitados con las mismas combinaciones lógicas consideradas en la simulación del circuito lógico, los resultados obtenidos en la simulación del circuito equivalente del layout se indican en la Fig.10.21.

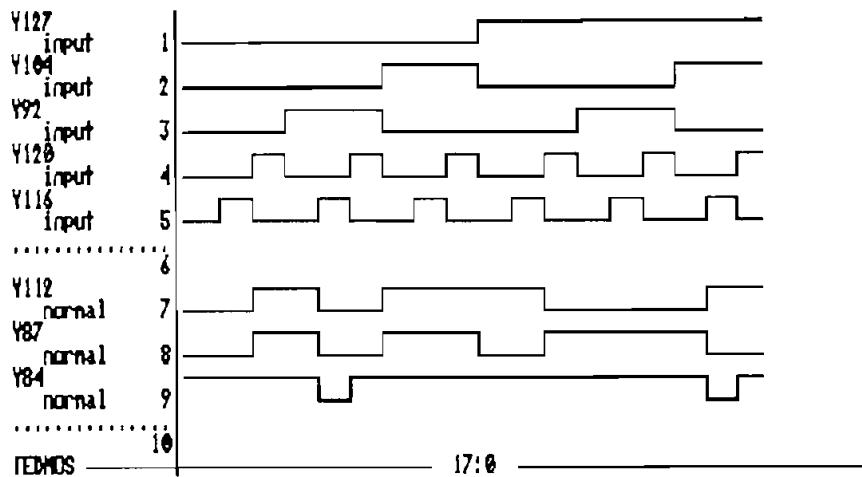


FIGURA 10.21 Simulación lógica del layout del CC2 del Plano 7(b)

Las señales obtenidas en los terminales de salida coinciden con aquellas obtenidas en la simulación del circuito lógico de la Fig.10.15 y con la formulación funcional de la Tabla 10.7, en consecuencia los diseños a los niveles físico, lógico y funcional de la Máquina de Estados Finitos 2 del Decodificador HDBn son equivalentes.

d) Diseño del bloque de salida

i) Formulación del módulo

Las restricciones a ser formuladas en la NETLIST SPICE para la orientación de los terminales de interfaz del bloque de salida de acuerdo al plano de base de la Fig.10.17 son las siguientes:

i.1) Los terminales de entrada: CLK\_D, ~CLK\_D, HDBn1\_D, ~HDBn1\_D, HDBn0\_D, ~HDBn0\_D, D20 y D21, deben tener orientación SUR.

i.2) El terminal de salida OUT debe tener orientación NORTE. en consecuencia la NETLIST SPICE para la síntesis del bloque de salida del Plano 5 es:

```
##-----##
## RELATORIO SPICE          SALIDA.sim ##
## DECODIFICADOR HDBn      ##
## ETAPA DE SALIDA         ##
##-----##
X60 I5      D20      I6      vcc      AND2
X61 ~HDBn1_D ~HDBn0_D D21      I1      vcc      NAND3
X62 ~HDBn1_D HDBn0_D F1      I2      vcc      NAND3
X63 HDBn1_D  ~HDBn0_D F2      I3      vcc      NAND3
X64 HDBn1_D  HDBn0_D F3      I4      vcc      NAND3
X65 I1      I2 I3  I4      I5      vcc      NAND4
X66 D21     CLK_D   ~CLK_D  ~F1    F1     vcc     FFD_MS
X67 F1      CLK_D   ~CLK_D  ~F2    F2     vcc     FFD_MS
X68 F2      CLK_D   ~CLK_D  ~F3    F3     vcc     FFD_MS
X69 I6      CLK_D   ~CLK_D  ~F4    OUT    vcc     FFD_MS

## TERMINALES DE SALIDA DEL MODULO ##
## ----- ##
# interface: CLK_D  # orientacao=S #
# interface: ~CLK_D # orientacao=S #
# interface: HDBn1_D # orientacao=S #
# interface: ~HDBn1_D # orientacao=S #
# interface: HDBn0_D # orientacao=S #
# interface: ~HDBn0_D # orientacao=S #
# interface: D20     # orientacao=S #
# interface: D21     # orientacao=S #
# interface: OUT     # orientacao=N #
.END
```

ii) Parámetros de distribución de celdas

Los parámetros para la distribución de las celdas en bandas son los mismos que se definieron en los módulos anteriores, el número de bandas ha sido establecido en:

c: Número de Bandas: 2

debido a que se tienen menos celdas que en los otros bloques del Decodificador.

iii) Distribución de celdas

La distribución de celdas generada por los programas de particionamiento y posicionamiento en función de los parámetros definidos se indica en la Fig.10.22.

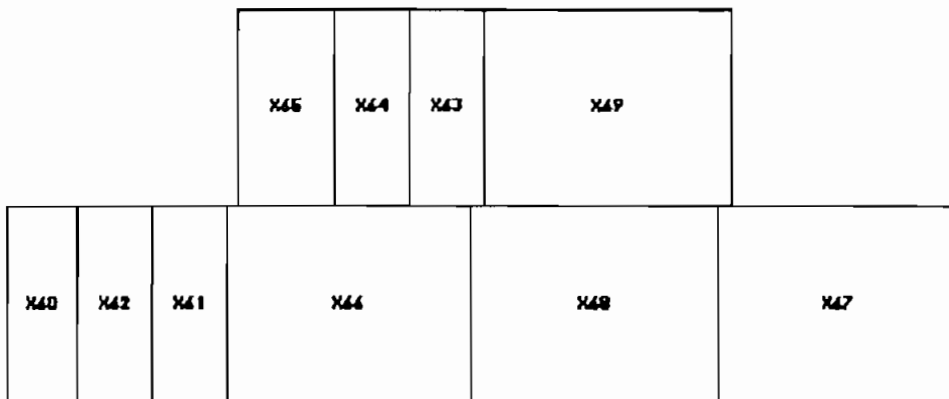


FIGURA 10.22 Distribución de celdas en la etapa de salida.

La posición final de cada celda dentro de su respectiva banda de posicionamiento es:

Posicionamiento do circuito C:\HD9n\SALIDA.pos

Banda 1

celula= X65 coordenada\_simbolica= 173 status rotacional=normal  
celula= X64 coordenada\_simbolica= 246 status rotacional=normal  
celula= X63 coordenada\_simbolica= 302 status rotacional=normal  
celula= X69 coordenada\_simbolica= 358 status rotacional=normal

Banda 2

celula= X60 coordenada\_simbolica= 0 status rotacional=espehada  
celula= X62 coordenada\_simbolica= 53 status rotacional=espehada  
celula= X61 coordenada\_simbolica= 109 status rotacional=espehada  
celula= X66 coordenada\_simbolica= 165 status rotacional=normal  
celula= X68 coordenada\_simbolica= 348 status rotacional=normal  
celula= X67 coordenada\_simbolica= 531 status rotacional=espehada

iv) Generación del layout

El layout del bloque de salida sintetizado a partir de la distribución de celdas de la Fig.10.22 se indica en el Plano 6(c), su descripción en formato RS y tiene la estructura jerárquica que se indica a continuación:

```
DS 1 1 10;          DS 23 1 10;          DS 25 1 10;
9 "AND_2.CEL";      9 "BANDA 1";          9 "hd-mx.rs";
.                   C 4 T 253 0;        C 23 T 0 100;
.                   C 3 T 338 0;        C 24;
DF;                 C 3 T 406 0;        9 "ENLACES ENTRE BANDAS";
DS 2 1 10;          C 2 T 474 0;          .
9 "FFD_MS.CEL";    9 "ENLACES ENTRE CELDAS";
.                   .
.                   .
DF;                 DF;
DS 3 1 10;          DS 24 1 10;          C 25;
9 "NAND_3.CEL";    9 "BANDA 2";          E
.                   C 1 T -149 -106 MX MY;
.                   C 3 T -217 -106 MX MY;
DF;                 C 3 T -309 -106 MX MY;
DS 4 1 10;          C 2 T 309 -106 MY;
9 "NAND_4.CEL";    C 2 T 504 -106 MY;
.                   C 2 T -966 -106 MX MY;
.                   9 "ENLACES ENTRE CELDAS";
DF;                 .
.                   .
.                   DF;
```

Las características de las bandas del layout del Plano 6(c) son las siguientes:

Roteamento da Banda 1 do Circuito C:\HDBn\SALIDA.  
Largura Banda=942  
Redes Nao Roteadas:

Roteamento da Banda 2 do Circuito C:\HDBn\SALIDA.  
Largura Banda=942  
Redes Nao Roteadas:

Informacoes do Modulo Gerado  
Area do Circuito = 203940  
Relacao de Aspecto = 0.208

En este caso se han formado todas las redes de interconexión (Redes Nao Roteadas=0), no obstante en el layout del Plano 6(c) deben corregirse los errores característicos del sistema TENTOS que fueron estudiados en el Capítulo 9. El layout corregido se indica en el Plano 7(c)

#### v) Verificación funcional del layout

El circuito equivalente al layout de la etapa de salida tiene la numeración de nodos y transistores que se indica en el Plano 7(c), para la formulación de la NETLIST NDL de este circuito se ha antecedido a la numeración de los nodos la letra "m", y se han cambiado los nodos "0" por Gnd y "1" por Vdd, así:

% CIRCUITO: SALIDA.nd1	NTRANS	m80	Gnd	m55	;% Tr_1	NTRANS	m70	Gnd	m40	;% Tr_11
% TECNOLOGIA: ECPD15	PTRANS	m80	Vdd	m55	;% Tr_2	PTRANS	m78	m79	m39	;% Tr_12
% CIRCUITO DECODIFICADOR HDBn	PTRANS	m55	Vdd	m41	;% Tr_3	NTRANS	m80	m40	m79	;% Tr_13
% ETAPA DE SALIDA	NTRANS	m55	m42	m78	;% Tr_4	PTRANS	m80	Vdd	m39	;% Tr_14
%	NTRANS	m70	Gnd	m42	;% Tr_5	PTRANS	m79	Vdd	m37	;% Tr_15
% Transistores tipo NMOS: 71	PTRANS	m79	m78	m41	;% Tr_6	NTRANS	m79	m38	m76	;% Tr_16
% Transistores tipo PMOS: 71	PTRANS	m70	Vdd	m41	;% Tr_7	NTRANS	m74	Gnd	m38	;% Tr_17
	NTRANS	m79	Gnd	m78	;% Tr_8	PTRANS	m77	m76	m37	;% Tr_18
INPUT m70 m74 m61 m57;	PTRANS	m70	Vdd	m39	;% Tr_9	PTRANS	m74	Vdd	m37	;% Tr_19
INPUT m58 m60 m63 m59;	NTRANS	m78	Gnd	m79	;% Tr_10	NTRANS	m77	Gnd	m76	;% Tr_20

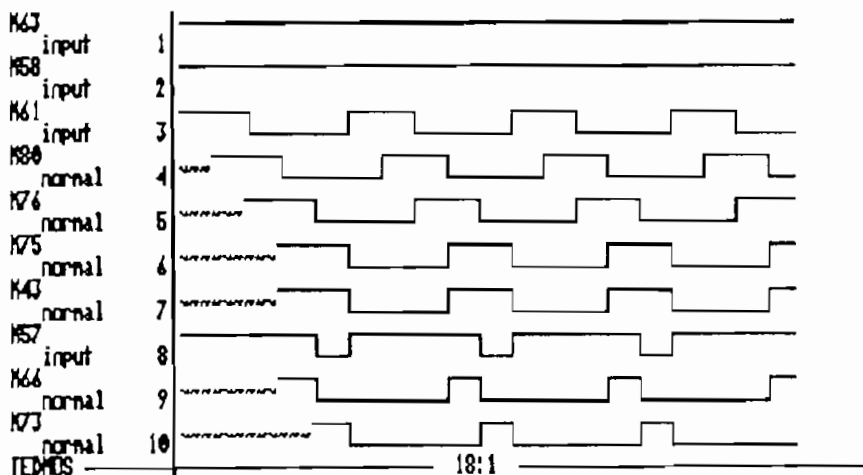


NTRANS	■76	Gnd	■77	;X	Tr_21	NTRANS	■72	Gnd	■71	;X	Tr_62	NTRANS	■64	Gnd	■65	;X	Tr_103
PTRANS	■74	Vdd	■35	;X	Tr_22	PTRANS	■70	Vdd	■21	;X	Tr_63	PTRANS	■70	Vdd	■11	;X	Tr_104
NTRANS	■74	Gnd	■36	;X	Tr_23	PTRANS	■70	Vdd	■23	;X	Tr_64	NTRANS	■63	■9	■10	;X	Tr_105
NTRANS	■78	■36	■77	;X	Tr_24	PTRANS	■72	■71	■21	;X	Tr_65	PTRANS	■63	Vdd	■52	;X	Tr_106
PTRANS	■76	■77	■35	;X	Tr_25	NTRANS	■70	Gnd	■22	;X	Tr_66	NTRANS	■58	Gnd	■9	;X	Tr_107
PTRANS	■78	Vdd	■35	;X	Tr_26	NTRANS	■68	■22	■71	;X	Tr_67	PTRANS	■58	Vdd	■52	;X	Tr_108
PTRANS	■71	Vdd	■34	;X	Tr_27	PTRANS	■68	Vdd	■21	;X	Tr_68	NTRANS	■65	Gnd	■64	;X	Tr_109
PTRANS	■75	■48	■34	;X	Tr_28	NTRANS	■76	Gnd	■68	;X	Tr_69	PTRANS	■70	Vdd	■7	;X	Tr_110
NTRANS	■74	Gnd	■33	;X	Tr_29	PTRANS	■76	Vdd	■68	;X	Tr_70	PTRANS	■65	■64	■7	;X	Tr_111
NTRANS	■71	■33	■48	;X	Tr_30	NTRANS	■47	Gnd	■69	;X	Tr_71	NTRANS	■70	Gnd	■8	;X	Tr_112
NTRANS	■75	Gnd	■48	;X	Tr_31	PTRANS	■70	Vdd	■19	;X	Tr_72	NTRANS	■62	■8	■64	;X	Tr_113
PTRANS	■74	Vdd	■34	;X	Tr_32	PTRANS	■47	■19	■69	;X	Tr_73	PTRANS	■62	Vdd	■7	;X	Tr_114
PTRANS	■69	Vdd	■32	;X	Tr_33	NTRANS	■70	Gnd	■20	;X	Tr_74	NTRANS	■61	Gnd	■62	;X	Tr_115
NTRANS	■69	■54	■31	;X	Tr_34	NTRANS	■67	■69	■20	;X	Tr_75	PTRANS	■61	Vdd	■62	;X	Tr_116
NTRANS	■74	Gnd	■31	;X	Tr_35	PTRANS	■67	Vdd	■19	;X	Tr_76	NTRANS	■46	■43	■6	;X	Tr_117
PTRANS	■73	■32	■54	;X	Tr_36	NTRANS	■66	Gnd	■67	;X	Tr_77	PTRANS	■46	Vdd	■43	;X	Tr_118
NTRANS	■73	Gnd	■54	;X	Tr_37	PTRANS	■66	Vdd	■67	;X	Tr_78	PTRANS	■52	Vdd	■43	;X	Tr_119
NTRANS	■48	Gnd	■75	;X	Tr_38	PTRANS	■64	Vdd	■18	;X	Tr_79	NTRANS	■52	■45	■6	;X	Tr_120
PTRANS	■74	Vdd	■32	;X	Tr_39	NTRANS	■64	■17	■53	;X	Tr_80	NTRANS	■60	Gnd	■5	;X	Tr_121
PTRANS	■74	Vdd	■29	;X	Tr_40	NTRANS	■74	Gnd	■17	;X	Tr_81	PTRANS	■60	Vdd	■51	;X	Tr_122
PTRANS	■48	■75	■29	;X	Tr_41	PTRANS	■80	■53	■18	;X	Tr_82	PTRANS	■61	Vdd	■51	;X	Tr_123
NTRANS	■74	Gnd	■30	;X	Tr_42	NTRANS	■80	Gnd	■53	;X	Tr_83	NTRANS	■61	■3	■5	;X	Tr_124
NTRANS	■72	■30	■75	;X	Tr_43	PTRANS	■74	Vdd	■18	;X	Tr_84	PTRANS	■51	Vdd	■43	;X	Tr_125
PTRANS	■72	Vdd	■29	;X	Tr_44	NTRANS	■76	■46	■16	;X	Tr_85	NTRANS	■51	■45	■4	;X	Tr_126
NTRANS	■54	Gnd	■73	;X	Tr_45	PTRANS	■76	Vdd	■46	;X	Tr_86	NTRANS	■44	Gnd	■4	;X	Tr_127
PTRANS	■74	Vdd	■26	;X	Tr_46	NTRANS	■53	Gnd	■80	;X	Tr_87	PTRANS	■44	Vdd	■43	;X	Tr_128
PTRANS	■54	■26	■73	;X	Tr_47	NTRANS	■63	■15	■16	;X	Tr_88	NTRANS	■59	■3	■51	;X	Tr_129
NTRANS	■74	Gnd	■28	;X	Tr_48	PTRANS	■63	Vdd	■46	;X	Tr_89	PTRANS	■59	Vdd	■51	;X	Tr_130
NTRANS	■47	■73	■28	;X	Tr_49	PTRANS	■74	Vdd	■13	;X	Tr_90	PTRANS	■80	Vdd	■44	;X	Tr_131
PTRANS	■76	Vdd	■25	;X	Tr_50	NTRANS	■60	Gnd	■15	;X	Tr_91	NTRANS	■80	Gnd	■2	;X	Tr_132
NTRANS	■70	Gnd	■27	;X	Tr_51	PTRANS	■53	■80	■13	;X	Tr_92	PTRANS	■58	Vdd	■44	;X	Tr_133
NTRANS	■76	■27	■72	;X	Tr_52	NTRANS	■65	■14	■80	;X	Tr_93	NTRANS	■58	■49	■2	;X	Tr_134
PTRANS	■47	Vdd	■26	;X	Tr_53	NTRANS	■74	Gnd	■14	;X	Tr_94	NTRANS	■59	■49	■44	;X	Tr_135
PTRANS	■71	■72	■25	;X	Tr_54	PTRANS	■60	Vdd	■46	;X	Tr_95	PTRANS	■59	Vdd	■44	;X	Tr_136
NTRANS	■71	Gnd	■72	;X	Tr_55	PTRANS	■65	Vdd	■13	;X	Tr_96	PTRANS	■43	Vdd	■56	;X	Tr_137
PTRANS	■70	Vdd	■25	;X	Tr_56	PTRANS	■61	Vdd	■11	;X	Tr_97	NTRANS	■43	■50	■56	;X	Tr_138
PTRANS	■66	Vdd	■23	;X	Tr_57	NTRANS	■61	■12	■65	;X	Tr_98	NTRANS	■57	Gnd	■50	;X	Tr_139
NTRANS	■66	■47	■24	;X	Tr_58	NTRANS	■70	Gnd	■12	;X	Tr_99	PTRANS	■57	Vdd	■56	;X	Tr_140
NTRANS	■70	Gnd	■24	;X	Tr_59	NTRANS	■75	■52	■10	;X	Tr_100	PTRANS	■56	Vdd	■66	;X	Tr_141
PTRANS	■69	■23	■47	;X	Tr_60	PTRANS	■75	Vdd	■52	;X	Tr_101	NTRANS	■56	Gnd	■66	;X	Tr_142
NTRANS	■69	Gnd	■47	;X	Tr_61	PTRANS	■64	■65	■11	;X	Tr_102						END

La correspondencia entre la numeración de los nodos en el layout del Plano 7(c) y su nominación en el circuito lógico del Plano 5, para los nodos a graficarse en los diagramas de simulación es la siguiente:

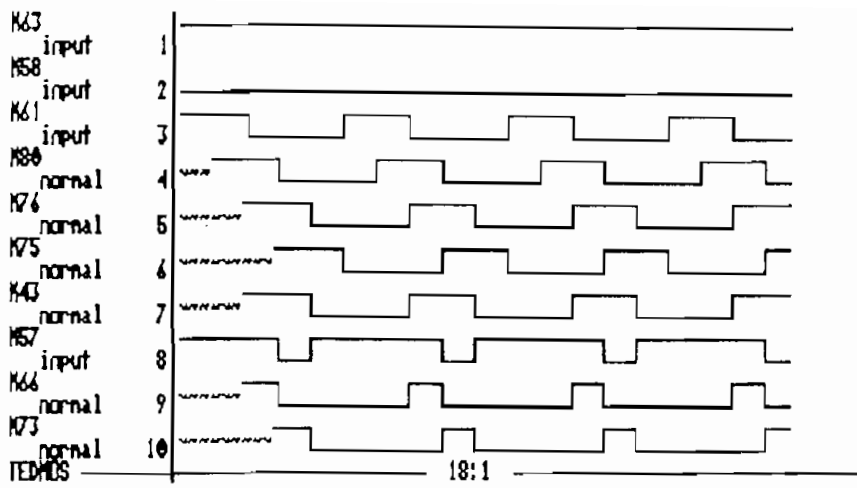
NODOS DEL CIRCUITO LOGICO		NODOS DEL LAYOUT
HDBn1_D	(entrada)	m63
HDBn0_D	(entrada)	m58
D21	(entrada)	m61
D20	(entrada)	m57
F1	(nodo interno)	m80
F2	(nodo interno)	m76
F3	(nodo interno)	m75
I5	(nodo interno)	m43
I6	(nodo interno)	m66
OUT	(salida)	m73

En la simulación de este circuito se han aplicado a los nodos m61(D21) y m57(D20) las mismas secuencias que se aplicaron en la simulación del circuito lógico, y se han considerado los cuatro casos de selección de señales: los nodos m63(HDBn1\_D) y m58 (HDBn0\_D), los resultados obtenidos se indican en la Fig.10.23.

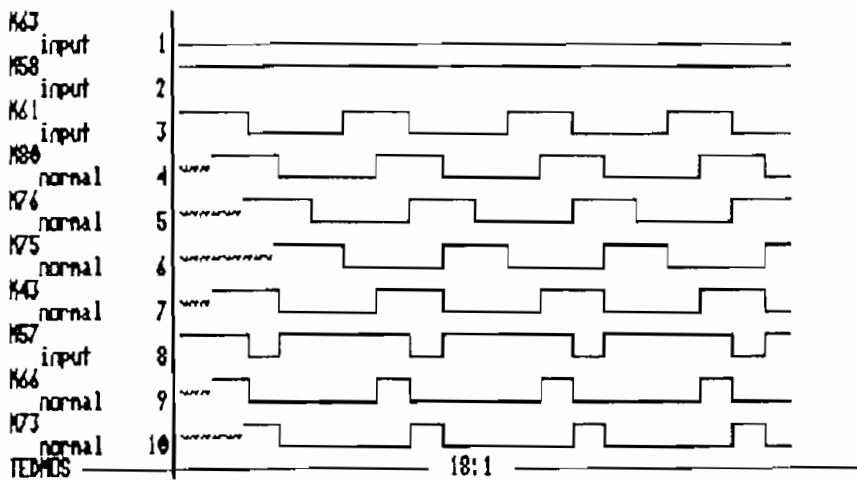


(a)

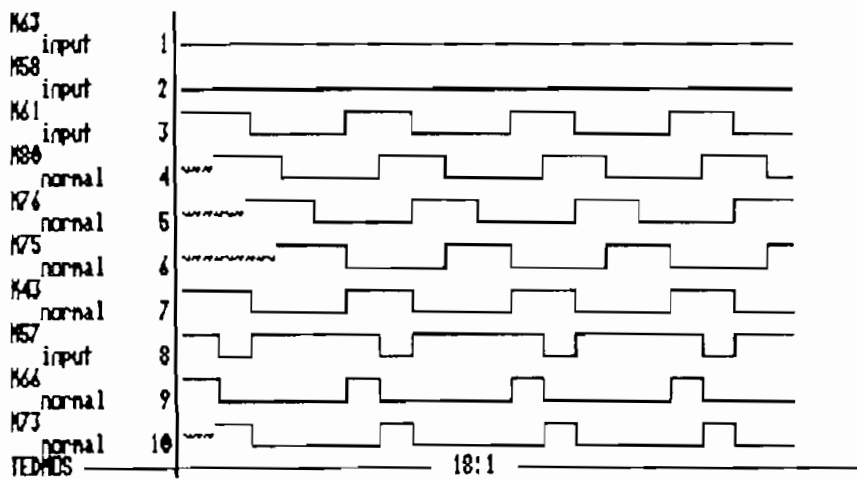
FIGURA 10.23 Simulación de la etapa de salida (a) Caso m63=1 m58=1



(b)



(c)



(d)

FIGURA 10.23 Cont.. (b) Caso  $m63=1$   $m58=0$  (c) Caso  $m63=0$   $m58=1$   
(d) Caso  $m63=0$   $m58=0$ .

En la Fig.10.23 se observa que los resultados obtenidos en la simulación del circuito equivalente del layout coinciden con aquellos de la simulación del circuito lógico que se indican en la Fig.10.16 y en la Tabla 10.8 para todos los casos, por tanto los diseños a los niveles físico y lógico son equivalentes y cumplen con los requerimientos funcionales para la etapa de salida propuestos en el Capítulo 9.

e) Generación del layout del Decodificador

Una vez el layout de cada uno de los bloques del plano de base de la Fig. 10.17 ha sido definido y verificado, el layout del circuito Decodificador HDBn se construye de manera similar a aquel del Codificador siguiendo los siguientes pasos:

- i) Los layouts parciales son integrados en un solo layout en conformidad con la distribución que les fue asignada en el plano de base.
  
- ii) Los terminales de los módulos son interconectados de acuerdo a las estrategias de enrutamiento definidas en el plano de base, las líneas de interconexión horizontales se construyen en Metal 1 y las verticales en Polysilicon, a excepción de las líneas de polarización que también se construyen en Metal 1.

El layout del circuito Decodificador HDBn así obtenido se indica en el Plano 8 del Anexo H.

### 10.3 DISEÑO DEL ANILLO DE PADS E INSERCIÓN DEL DISEÑO GLOBAL.

Como se indicó en la formulación de los planos de base de los circuitos Codificador y Decodificador HDBn de las Figuras 10.5 y 10.17 respectivamente, estos planos corresponden únicamente a secciones del núcleo del layout, en el plano de base total los circuitos Codificador y Decodificador pasan a ser secciones del núcleo, además deberán considerarse los PADS que interconectan el núcleo con los pines externos del chip.

#### 10.3.1 Plano de base del núcleo

El plano de base del núcleo del circuito Codificador/Decodificador HDBn se indica en la Fig.10.24.

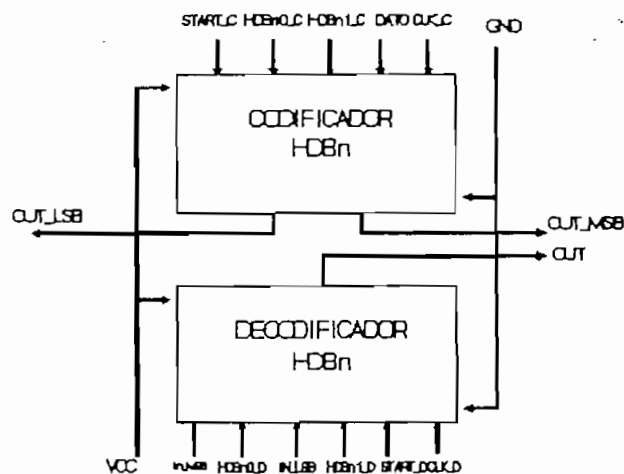


FIGURA 10.24 Plano de Base del núcleo del circuito Codificador/Decodificador HDBn

En el plano de base de la Fig.10.24 se tiene que:

- a) la orientación de los terminales de entrada/salida, así como la posición de las líneas de polarización de cada sección ya fue definida en sus planos de base particulares y obtenida en sus layouts respectivos.
- b) El núcleo está formado por dos layouts correspondientes al Codificador y Decodificador que son física y funcionalmente independientes ya que salvo las líneas de polarización, no existen caminos de enrutamiento que los interconecten.

### 10.3.2 Selección e interconexión de PADS

Los bonding PADS seleccionados para los terminales de entrada/salida del plano de base de la Fig.10.24 corresponden a los mismos que se emplearon en el diseño del circuito Medio Sumador del Capítulo 9.

- i) PAD de entrada: PAD IPS8G (TTL Input Buffer).
- ii) PAD de salida: PAD OPS1U (4mA Output Buffer).

En el núcleo del circuito (Fig.10.24) se tienen 11 terminales de entrada y 3 de salida, es decir 14 terminales de entrada/salida, considerando que debe existir al menos un par de PADS de polarización-tierra por cada 64 terminales de entrada/salida, se ha empleado para la alimentación del núcleo y del anillo externo los PADS:

- i) FWRCO (5V pad for core).
- ii) GNDCO (0V pad for core).
- iii) FWRPY (5V pad for periphery)
- iv) GNDPY (0V pad for periphery)

Nótese que han sido colocados los PADS FWRCO y GNDCO que en el circuito "Medio Sumador" no fueron considerados por limitaciones de espacio y porque por tratarse de un layout tan pequeño los PADS FWRPY y GNDPY resultaban suficientes para polarizar al núcleo y la periferia del layout simultáneamente.

Las características eléctricas de los PADS empleados se describen en el Anexo F.2.

El anillo que une a estos PADS y que encierra al núcleo del layout del circuito ha sido construido con las mismas celdas de enlace lateral y de esquinas que se diseñaron en el Capítulo 9 para el circuito "Medio Sumador".

### 10.3.3 Plano de base global del diseño

Considerando el plano de base del núcleo del circuito (Fig de 10.24) y los PADS de entrada/salida y alimentación seleccionados, el plano de base global del circuito Codificador/Decodificador HDBn es el que se indica en la Fig.10.25 en esta figura se definen las estrategias de interconexión PAD-Núcleo y PAD-PAD que forman el anillo de PADS, nótese que

se han colocado simétricamente los PADS de polarización y tierra a fin de distribuir equitativamente las señales de alimentación tanto a las dos regiones del núcleo como al anillo de PADS.

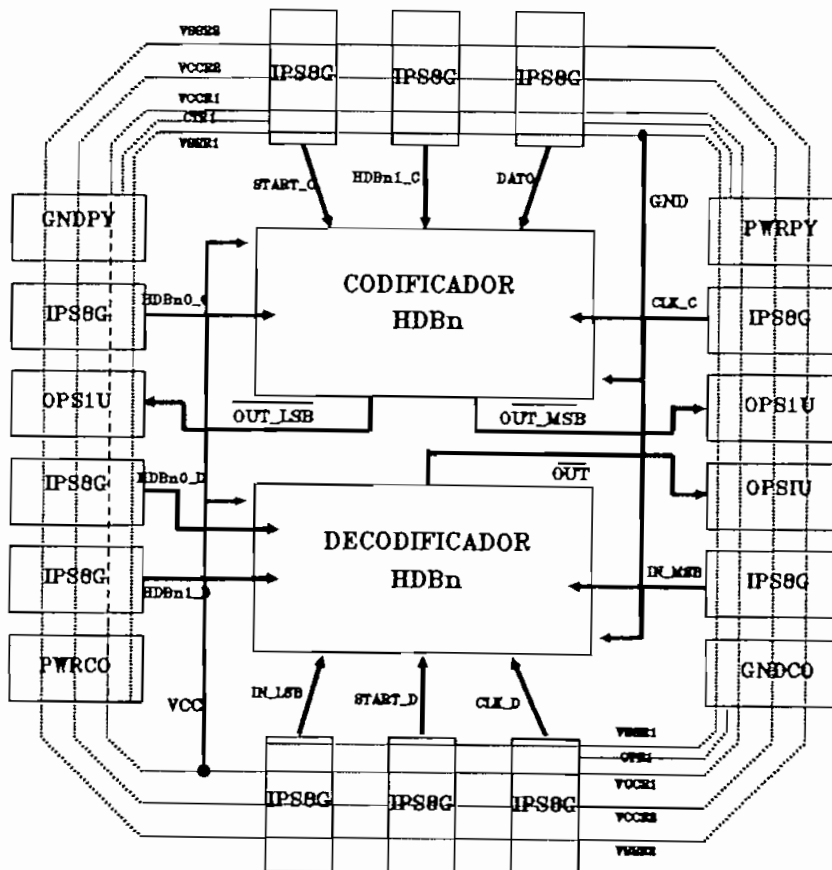


FIGURA 10.25 Plano de base global del diseño



NOTA:

En el plano de base de la Fig.10.25 se han tomado las salidas complementarias:  $\sim$ OUT\_LSB,  $\sim$ OUT\_MSB y  $\sim$ OUT desde los terminales ( $\sim$ Q) de las celdas FFD\_MS que generan las señales de salida, puesto que como se observa en el Anexo F.2, los PADs OPS1U empleados invierten el valor lógico de la señal de salida.

#### 10.3.4 Layout global del diseño

Finalmente, el layout global del circuito Codificador/Decodificador HDBn ha sido construido de acuerdo al plano de base de la Fig.10.25 como se indica en el Plano 9 del Anexo H.

En este plano se tiene que:

- a) Los terminales de acceso a los PADs son construidos en METAL 2 y los terminales de salida del núcleo son de POLYSILICON por lo que han sido cambiados a METAL 1 a través de CONTACTOS y de éste a METAL 2 a través de VIAS.
- b) Los terminales laterales de los PADs son construidos en METAL 2, por lo que el anillo que rodea al núcleo está formado íntegramente por este material.

- c) Los terminales de alimentación del núcleo son construídos en METAL 1 y para enlazarlas al anillo de PADS ha sido necesario cambiarlos a METAL 2 a través de VIAS.
  
- e) La estructura detallada de cada PAD no se indica puesto que el diseñador únicamente tiene acceso a los terminales de interconexión de los PADS, la integración completa de los PADS al diseño deberá ser realizada por el Centro Nacional de Microelectrónica de Barcelona - España (CNM) en las posiciones indicadas en el layout.

Todo el proceso de generación del layout global, esto es: posicionamiento e interconexión de módulos, interconexión de terminales con PADS y construcción del anillo de PADS ha sido realizado manualmente en el Editor de Máscaras EMA2 a manera de un diseño FULL-CUSTOM, el layout así obtenido se indica en el Plano 9 del Anexo H, y corresponde al layout final del diseño al nivel físico del circuito Codificador/Decodificador HDEn.

## CAPITULO 11

### DISEÑO DEL CODIFICADOR/DECODIFICADOR HDB<sub>n</sub> BASADO EN CELDA ESTANDAR - METODO ESTRUCTURADO PPL

Este capítulo tiene como objetivo demostrar la facilidad del proceso de diseño con PPL, para un circuito más complejo que el de la simple compuerta EXOR presentado en el Cap.8. En el Cap. 9 se presentó el diseño lógico del codificador/decodificador HDB<sub>n</sub>, para realizar el diseño con PPL se parte de las ecuaciones lógicas planteadas y en base a ellas se planifica la distribución de celdas necesaria para cumplir con cada ecuación, para luego proceder a utilizar las herramientas PPL.

Se demuestra una vez más la flexibilidad de diseño que ofrece PPL, la distribución de las celdas de un circuito ya diseñado puede ser alterada de innumerables maneras y aún continuar realizando la misma tarea.

Se presenta de manera detallada el proceso de diseño de la máquina de estados finitos MEF(1) del codificador. Para las tres máquinas restantes, el diseño se presenta de manera más simplificada.

Se detalla también el diseño de las etapas de salida utilizadas para el codificador y decodificador y finalmente la inserción del codificador y decodificador completos en el *PAD-FRAME tiny34*.

## 11.1 DISEÑO DEL CODIFICADOR PROGRAMABLE HDBn

La Fig. 11.1 presenta el esquema general del codificador HDBn, con la nomenclatura a utilizarse.

### 11.1.1 MEF(1)

Las ecuaciones planteadas para la MEF(1) son:

$$D_2 = A_4 \bar{A}_2 \bar{A}_1 \bar{A}_0 + A_3 A_2 \bar{A}_1 \bar{A}_0$$

$$D_1 = A_3 \bar{A}_1 \bar{A}_0 + A_4 \bar{A}_2 \bar{A}_1 \bar{A}_0$$

$$D_0 = \bar{A}_3 A_1 \bar{A}_0 + A_3 A_2 \bar{A}_1 \bar{A}_0 + \bar{A}_4 A_3 \bar{A}_1 \bar{A}_0$$

Las ecuaciones están planteadas como sumas de productos. Como se explicó en el Cap.8, con PFL estas ecuaciones se pueden realizar fácilmente con las celdas básicas: "I", "O", "1" y "+". Primeramente se plantea la solución de la MEF(1) sin considerar ninguno de los *flip-flops* de realimentación, se simula el circuito obtenido y luego simplemente se añaden los *flip flops*.

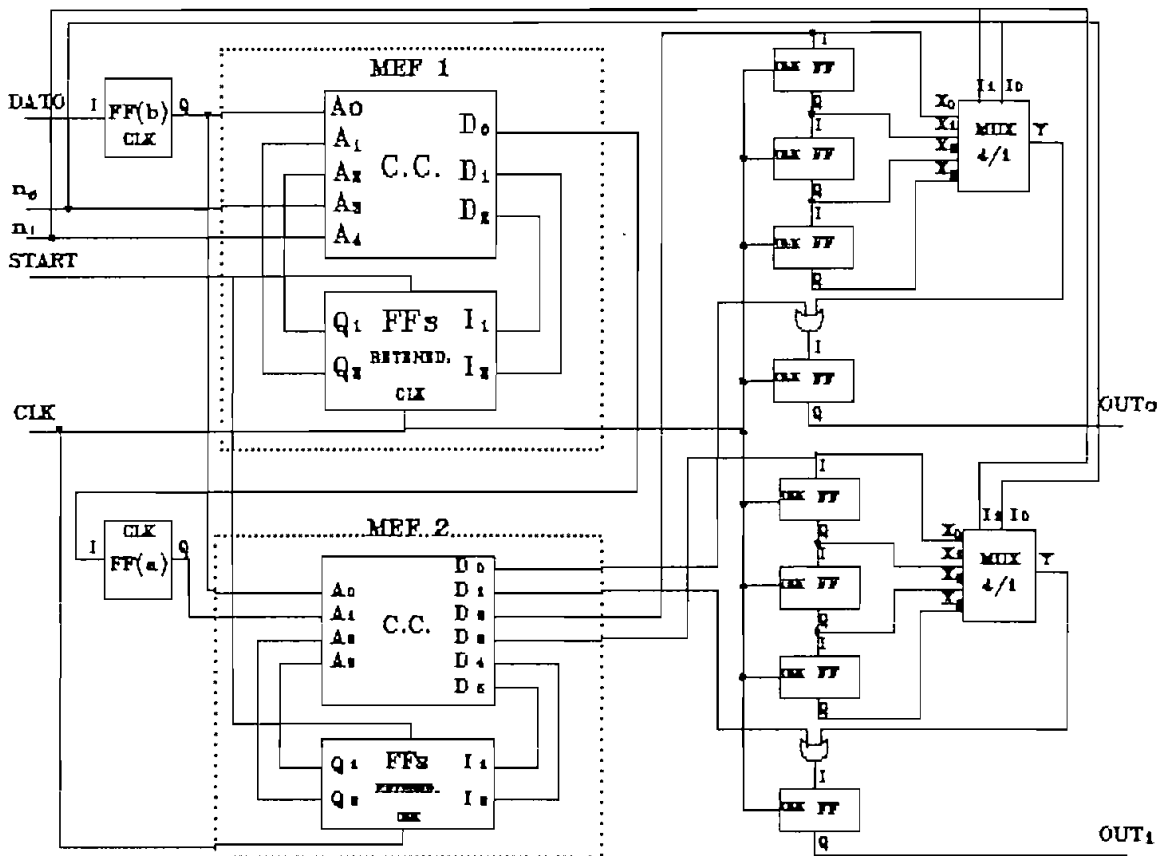


FIGURA 11.1 Esquema general del codificador HDBn.

$$a) D_2 = A_4 \bar{A}_2 A_1 \bar{A}_0 + A_3 A_2 \bar{A}_1 \bar{A}_0$$

Para esta ecuación se tienen dos términos AND y una suma; esto indica la necesidad de utilizar una fila para cada término, para luego realizar la suma. Como se requiere de algunas de las señales de entrada complementadas, se utilizan para estos casos celdas "I" (R3). Cada producto se resuelve con celdas "0" y "1" colocadas en fila, para la suma se ubican celdas "+" en una misma columna. En la Fig. 11.2 se presenta la distribución de las celdas necesarias para cumplir con  $D_2$ .

	$A_4$	$A_3$	$A_2$	$A_1$	$A_0$	$D_2$		
			I	I	I		R3	
$A_4 \bar{A}_2 A_1 \bar{A}_0$	1		0	1	0	+	R2	
$A_3 A_2 \bar{A}_1 \bar{A}_0$		1	1	0	0	+	R1	
							R0	
	C0	C1	C2	C3	C4	C5	C6	C7

FIGURA 11.2 Distribución de celdas para  $D_2$ .

Como puede observarse en la Fig. 11.2, por cada variable que aparezca en un producto, se ubica una celda "0" o "1" en la columna correspondiente a la variable, considerando si se requiere la variable o su complemento. Para realizar la suma lógica se utiliza por cada fila una celda "+". En las Fig. 11.3 y 11.4 se presenta la distribución de celdas para cumplir con  $D_1$  y  $D_0$ , respectivamente.

$$b) D_1 = A_3 \bar{A}_1 \bar{A}_0 + A_4 \bar{A}_2 \bar{A}_1 \bar{A}_0$$

	A <sub>4</sub>	A <sub>3</sub>	A <sub>2</sub>	A <sub>1</sub>	A <sub>0</sub>	D <sub>1</sub>		
			I	I	I		R3	
A <sub>3</sub> $\bar{A}_1$ $\bar{A}_0$		1		0	0	+	R2	
A <sub>4</sub> $\bar{A}_2$ $\bar{A}_1$ $\bar{A}_0$	1		0	0	0	+	R1	
							R0	
	C0	C1	C2	C3	C4	C5	C6	C7

FIGURA 11.3 Distribución de celdas para D<sub>1</sub>.

$$c) D_0 = \bar{A}_3 A_1 \bar{A}_0 + A_3 A_2 \bar{A}_1 \bar{A}_0 + \bar{A}_4 A_3 \bar{A}_1 \bar{A}_0$$

	A <sub>4</sub>	A <sub>3</sub>	A <sub>2</sub>	A <sub>1</sub>	A <sub>0</sub>	D <sub>0</sub>		
	I	I		I	I		R3	
$\bar{A}_3 A_1 \bar{A}_0$		0		1	0	+	R2	
A <sub>3</sub> A <sub>2</sub> $\bar{A}_1$ $\bar{A}_0$		1	1	0	0	+	R1	
$\bar{A}_4 A_3 \bar{A}_1 \bar{A}_0$	0	1		0	0	+	R0	
	C0	C1	C2	C3	C4	C5	C6	C7

FIGURA 11.4 Distribución de celdas para D<sub>0</sub>.

La integración de las celdas utilizadas para cada una de las funciones resueltas se presenta en la Fig. 11.5. La primera acción tomada al obtener el arreglo completo es agregar columnas para permitir el paso de las señales D<sub>2</sub>, D<sub>1</sub> y D<sub>0</sub> en caminos separados, evitando que todas se combinen en un solo camino. Para esto se decide que cada señal de salida disponga de una columna.

	$A_4$	$A_3$	$A_2$	$A_1$	$A_0$	$D_2$	$D_1$	$D_0$		
			I	I	I				R9	
$A_4\bar{A}_2A_1\bar{A}_0$	1		0	1	0	+			R8	
$A_3A_2\bar{A}_1\bar{A}_0$		1	1	0	0	+			R7	
			I	I	I				R6	
$A_3\bar{A}_1\bar{A}_0$		1		0	0		+		R5	
$A_4\bar{A}_2\bar{A}_1\bar{A}_0$	1		0	0	0		+		R4	
	I	I		I	I				R3	
$\bar{A}_3A_1\bar{A}_0$		0		1	0			+	R2	
$A_3A_2\bar{A}_1\bar{A}_0$		1	1	0	0			+	R1	
$\bar{A}_4A_3\bar{A}_1\bar{A}_0$	0	1		0	0			+	R0	
	C0	C1	C2	C3	C4	C5	C6	C7	C8	C9

FIGURA 11.5 Distribución de celdas para  $D_2$ ,  $D_1$  y  $D_0$ .

Como para todas las señales, alguna de las filas requiere de sus complementos, se puede tener una sola fila de inversores. Además, la distribución de celdas "0s" y "1s" para las filas R1 y R7 es la misma, se puede también integrarlas en una única fila, tan solo haciendo que la señal resultante, de la única fila, vaya hacia las señales  $D_2$  y  $D_0$ . Los cambios mencionados se presentan en la Fig. 11.6.

Todas las ecuaciones de la MEF(1) requieren una grilla de 7 filas x 8 columnas. Para completar el diseño de la MEF(1) se deben agregar los *flip-flops* de los caminos de realimentación.



	$A_4$	$A_3$	$A_2$	$A_1$	$A_0$	$D_2$	$D_1$	$D_0$		
	I	I	I	I	I				R6	
$A_4\bar{A}_2A_1\bar{A}_0$	1		0	1	0	+			R5	
$A_3A_2\bar{A}_1\bar{A}_0$		1	1	0	0	+		+	R4	
$A_3\bar{A}_1\bar{A}_0$		1		0	0		+		R3	
$A_4\bar{A}_2\bar{A}_1\bar{A}_0$	1		0	0	0		+		R2	
$\bar{A}_3A_1\bar{A}_0$		0		1	0			+	R1	
$\bar{A}_4A_3\bar{A}_1\bar{A}_0$	0	1		0	0			+	R0	
	C0	C1	C2	C3	C4	C5	C6	C7	C8	C9

FIGURA 11.6 Distribución de celdas simplificada para  $D_2$ ,  $D_1$  y  $D_0$ .

Para demostrar que la distribución de las celdas de la Fig. 11.6 cumple con las funciones  $D_2$ ,  $D_1$  y  $D_0$ , se procede a describir el circuito utilizando TILER, hasta conseguir su simulación. En todos los archivos que se generen con TILER, para evitar mensajes de error por nombres de nodos duplicados, cuando se integran los diferentes módulos en el codificador total, se utiliza en lugar de  $A_1$ ,  $A_2$ , etc, la notación  $A_{11}$ ,  $A_{12}$ , etc, indicando el primer dígito del subíndice que es parte de la MEF(1).

A continuación se presentan el archivo de comandos para la simulación (.src) y los resultados obtenidos (.cpy). Se han definido los vectores: *IN*, que incluye todas las señales de entrada del circuito combinacional (CC.) y *OUT*, que incluye todas las señales de salida. Para obtener los resultados en forma de tablas se agrega en el archivo (.src) la línea:

## OPTIONS TABLES -COPYINPUTS

Esta línea evita que los nombres especificados con el comando *WATCH (IN, OUT)*, se copien al archivo (.cpy) y que además los resultados se presenten en forma de tabla.

ARCHIVO (.SRC)

```
;SIMULACION DEL CIRCUITO COMBINACIONAL DE LA MEF(1) DEL CODIFICADOR
COPY
VECTOR IN A14 A13 A12 A11 A10
VECTOR OUT D12 D11 D10
OPTIONS TABLES -COPYINPUTS
WATCH IN OUT
```

```
;*** AMI ***
SET IN=00000
CYCLE
SET IN=00001
CYCLE
;*** HDB1 ***
SET IN=01000
CYCLE
SET IN=01001
CYCLE
SET IN=01010
CYCLE
SET IN=01011
CYCLE
;*** HDB2 ***
SET IN=10000
CYCLE
SET IN=10001
CYCLE
SET IN=10010
CYCLE
SET IN=10011
CYCLE
SET IN=10100
CYCLE
SET IN=10101
CYCLE
;*** HDB3 ***
SET IN=11000
CYCLE
SET IN=11001
CYCLE
SET IN=11010
CYCLE
SET IN=11011
CYCLE
SET IN=11100
CYCLE
SET IN=11101
CYCLE
SET IN=11110
CYCLE
SET IN=11111
CYCLE
```

ARCHIVO (.CPY)

```
1> 00000 000
2> 00001 000
3> 01000 011
4> 01001 000
5> 01010 000
6> 01011 000
7> 10000 010
8> 10001 000
9> 10010 101
10> 10011 000
11> 10100 000
12> 10101 000
13> 11000 010
14> 11001 000
15> 11010 100
16> 11011 000
17> 11100 111
18> 11101 000
19> 11110 000
20> 11111 000
```

Los valores de la tabla del archivo (.cpy) son idénticos a los planteados en la Tabla 9.9, y se concluye que el diseño del CC. está correcto.

Una vez realizado el diseño del CC., simplemente se deben incluir los *flip flops* necesarios. El conjunto de celdas PFL incluye un *flip flop* tipo "D". El nombre de esta celda es precisamente "D".

d) Celda "D"

Esta celda ocupa 3 filas X 2 columnas de la grilla PFL y realiza la función de un *flip flop* tipo "D". En la Fig. 11.7 se muestran tres representaciones para esta celda, para los modificadores de 4 y 5, que son las celdas utilizadas en los diseños del codificador y decodificador. Cada una de estas celdas utiliza en su estructura un total de 20 transistores.

PFL utiliza un esquema de temporización de dos fases no sobrelapadas. La celda "D" utiliza por tal razón dos fases ( $\phi_{in}$  y  $\phi_{out}$ ) para su operación. Las entradas se muestrean cuando  $\phi_{in}$  está en bajo y el *flip flop* cambia sus salidas cuando  $\phi_{out}$  está en bajo. Para la operación de la celda debe proporcionar las señales  $\phi_{in}$  y  $\phi_{out}$  y sus complementos. Es responsabilidad del diseñador asegurar que las fases de las señales no se sobrelapen, para lo cual se dispone en el conjunto de celdas la celda "4".

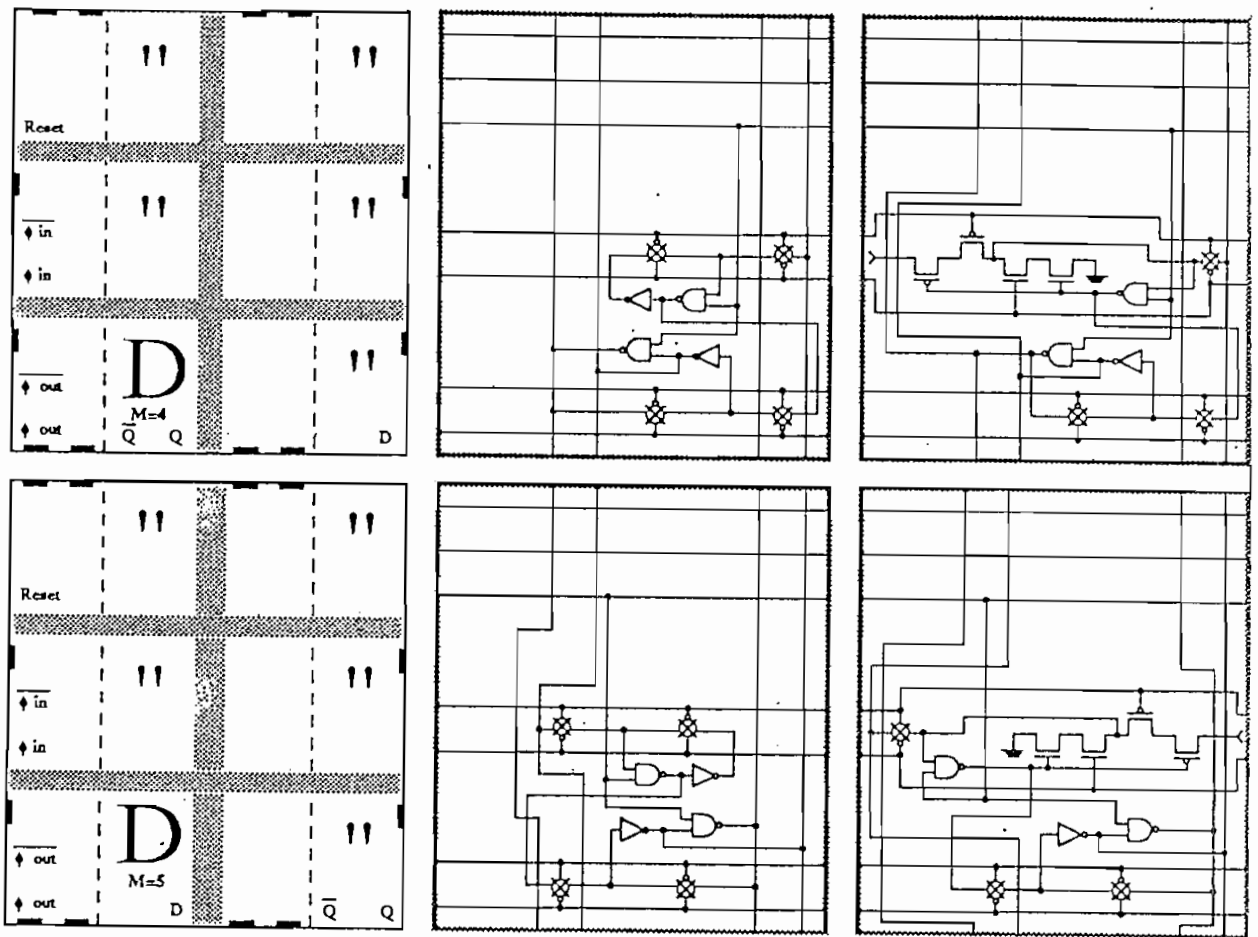


FIGURA 11.7 Celdas "D" con  $M=4$  y  $M=5$ .

Para llevar a los *flip flops* a un estado conocido, se dispone de una entrada de reset asincrónico (*Reset*), que se activa con una señal en bajo. La celda recibe la entrada de datos (*D*) y entrega la salida *Q* y su complemento ( $\bar{Q}$ ). Las señales de entrada y salida están disponibles en los caminos de las columnas, las señales de reloj y *reset* deben entregarse a la celda en caminos horizontales. La diferencia entre las celdas con  $M=4$  y  $M=5$ , es la columna en la que se tienen las señales *D*, *Q* y  $\bar{Q}$ .

#### e) Celda "4"

En la Fig. 11.8 se presenta la celda "4", con su circuito equivalente a nivel de compuertas y las señales de reloj generadas a partir de la señal externa "clk".

Las señales proporcionadas por la celda "4" deben ser conectadas directamente a las celdas "D"; si se utilizan inversores o buffers entre ellas, se corre el riesgo de producir sobrelapamientos de las señales de reloj. Cuando las celdas "4" manejan numerosas celdas "D" (u otro tipo de celdas que requieran de la celda "4"), se utilizan celdas con modificadores, que permiten manejar mayor número de celdas. Para las celdas "4", el modificador altera el tamaño de sus transistores. La celda "4" con  $M=0$  permite manejar hasta 12 celdas "D" y con  $M=1$  entre 12 y 24. Cada celda "4" tiene en su estructura un total de 18 transistores. Las 4 salidas se disponen en filas y columnas.

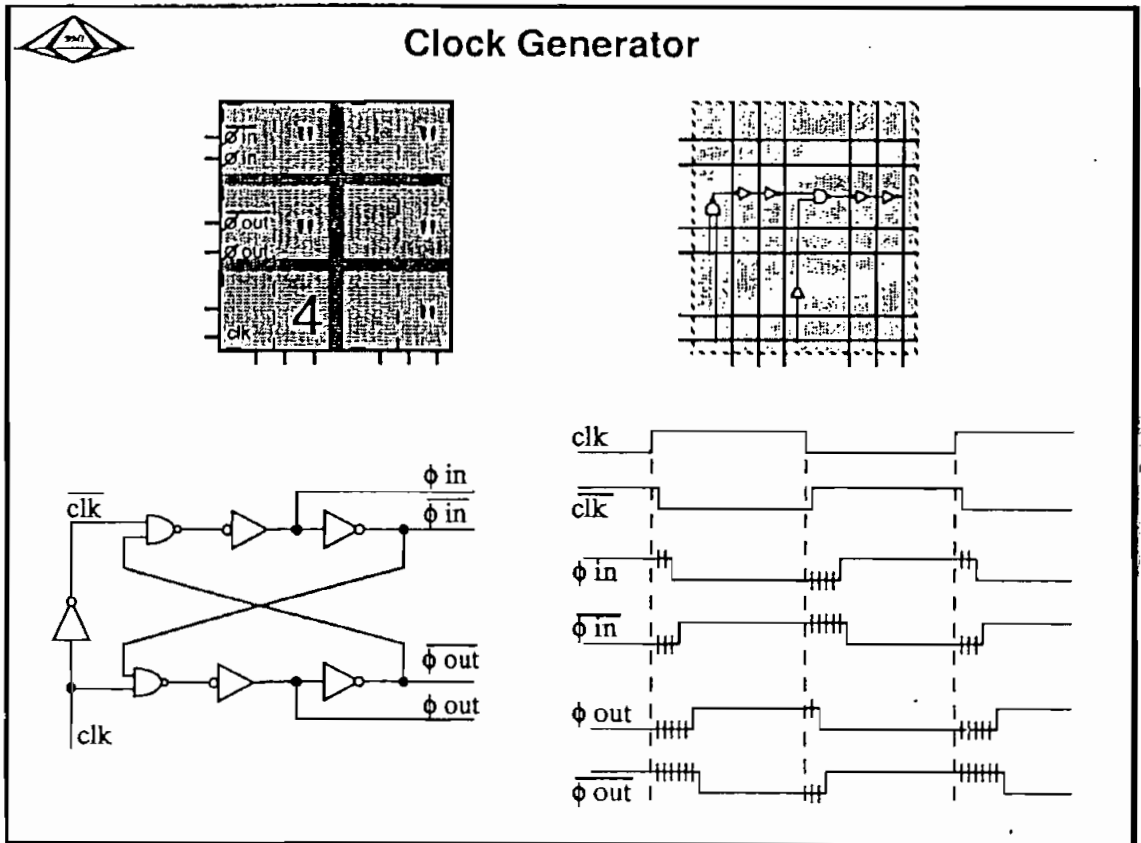


FIGURA 11.8 Celda '4'.

f) MEF(1) completa

En la Figura 11.9 se presenta el diagrama de bloques de la MEF(1), incluidos los dos *flip flops* de realimentación y el de la salida  $D_0$  (FFa de la Fig.11.1).

Para incluir los *flip flops* de realimentación, optimizando el área utilizada y eliminando la necesidad de demasiadas celdas de interconexión, se modifica el orden de las columnas de la Fig. 11.6, aprovechando las propiedades de las celdas PPL (Fig. 11.10).

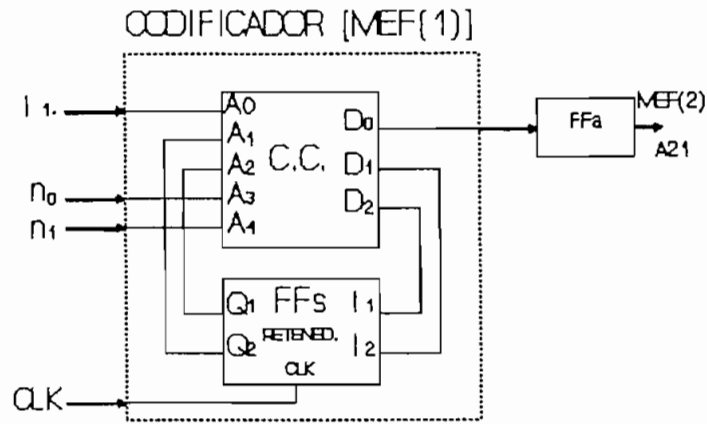


FIGURA 11.9 Esquema de la MEF(1) completa.

	$A_4$	$A_3$	$A_2$	$D_2$	$A_1$	$D_1$	$D_0$	$A_0$		
	I	I	I		I			I	R6	
$A_4 \bar{A}_2 A_1 \bar{A}_0$	1		0	+	1			0	R5	
$A_3 A_2 \bar{A}_1 \bar{A}_0$		1	1	+	0		+	0	R4	
$A_3 \bar{A}_1 \bar{A}_0$		1			0	+		0	R3	
$A_4 \bar{A}_2 \bar{A}_1 \bar{A}_0$	1		0		0	+		0	R2	
$\bar{A}_3 A_1 \bar{A}_0$		0			1		+	0	R1	
$\bar{A}_4 A_3 \bar{A}_1 \bar{A}_0$	0	1			0		+	0	R0	
	C0	C1	C2	C3	C4	C5	C6	C7	C8	C9

FIGURA 11.10 Variación de la distribución de celdas de la Fig. 11.6.

En el arreglo de la Fig. 11.10, se añaden los *flip flops* de realimentación, como se indica en la Fig. 11.11. Queda claro el objetivo de haber variado el orden de las columnas, pues con esto,  $D_2$  de C4, ingresa directamente al *flip flop* 2 y la salida del *flip flop* (en C3) es la señal  $A_2$ . Algo similar ocurre para el otro *flip flop*, la señal  $D_1$  ingresa al *flip flop* 1 en C6 y su salida en la columna C5 es  $A_1$ .

Nótese que la representación de las celdas utilizada en las grillas presentadas, no es la empleada en el manual de PPL, se la ha adoptado solamente como parte de la explicación de la planificación de la distribución de las celdas.

	$A_4$	$A_3$	$A_2$	$D_2$	$A_1$	$D_1$	$D_0$	$A_0$		
	I	I	I		I			I	R10	
	1		0	+	1			0	R9	
		1	1	+	0		+	0	R8	
		1			0	+		0	R7	
	1		0		0	+		0	R6	
		0			1		+	0	R5	
	0	1			0		+	0	R4	
				$\bar{Q}$ Q D	$\bar{Q}$ Q D				R3	
									R2	
				D2 M=	D1 M=4				R1	
									R0	
	C0	C1	C2	C3	C4	C5	C6	C7	C8	C9

FIGURA 11.11 Inclusión de los *flip flops* para la realimentación de la MEF(1).



Como la celda "D" ofrece la salida Q y su complemento, es innecesario utilizar celdas "I" para A<sub>1</sub> y A<sub>2</sub>, como se presenta en la Fig. 11.12. Las celdas "I" que quedan pueden trasladarse a otra fila, dentro de la misma columna, en una localidad que no tenga otra celda; esto se hace para A<sub>3</sub> y A<sub>4</sub>. Como se indica en el diagrama de bloques de la MEF(1) de la Fig. 11.9, la salida D<sub>0</sub> debe pasar por un *flip flop* antes de ingresar a la MEF(2). Este *flip flop* se incluye en la Fig. 11.12 con M=5, la señal D<sub>0</sub> ingresa al *flip flop* en C7 y sale en C8. Para que no exista conflicto entre las salidas del *flip flop* y las señales A<sub>0</sub> y su complemento (que corren en C8), éstas últimas deben interrumpirse en (R4:C8); la salida del *flip flop* se toma en la parte inferior en (R0:C8).

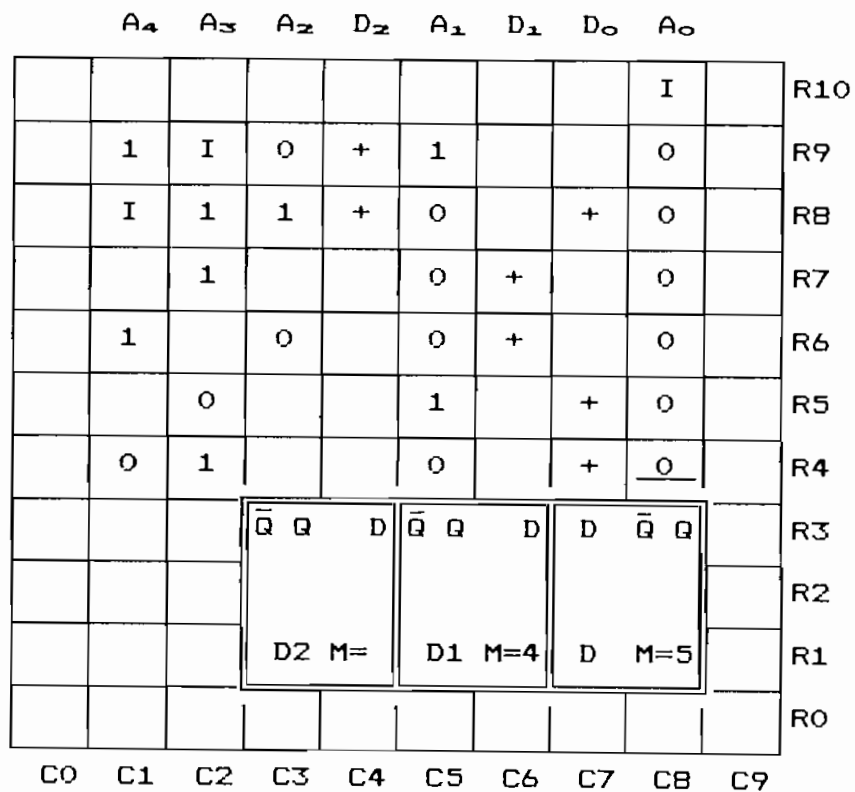


FIGURA 11.12 Inclusión del *flip flop* para D<sub>0</sub>.

La estructura de la celda "D", permite que todas las señales de reloj ingresen, por la izquierda o derecha, a cumplir las funciones necesarias en el interior de la celda y que además estén disponibles en el otro lado. Esta característica hace que al proveer una celda "4" para las señales de reloj de un *flip flop*, los otros dos colocados en fila recibían también las señales de reloj. Para el presente diseño se ha decidido colocar la celda "4" al lado izquierdo de los 3 *flip flops*. La ubicación adecuada de esta celda se muestra en la Fig. 11.13. En este caso se debe también interrumpir los caminos de las columnas C1 y C2 en (R3:C1) y (R3:C2), para que no exista conflicto con las salidas de la celda "4".

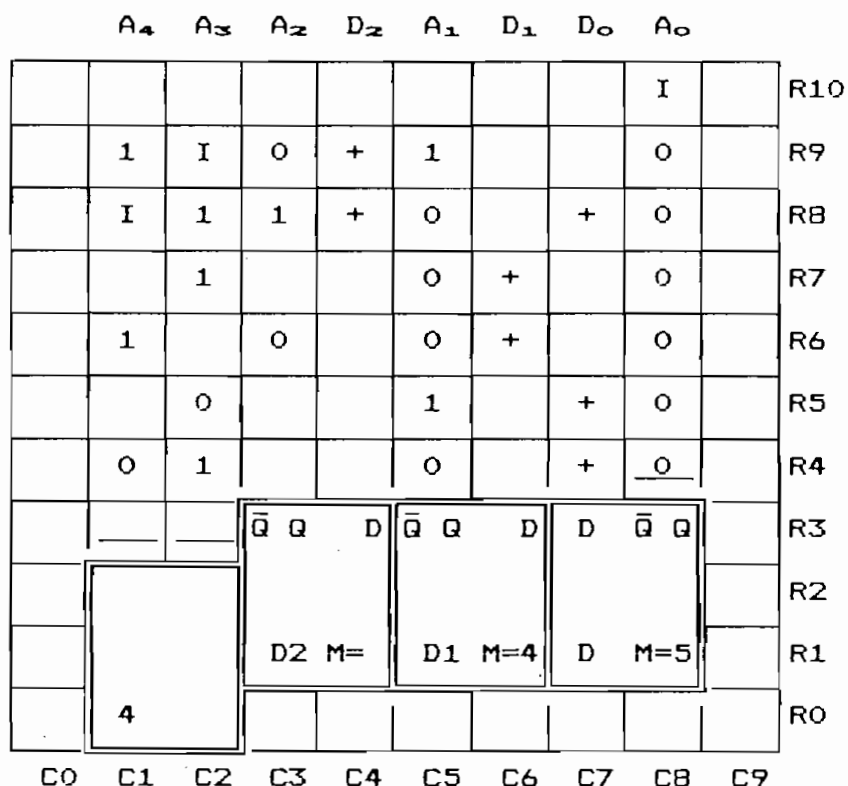


FIGURA 11.13 Inclusión de la celda "4".

El *reset* de los *flip flops*, activo en bajo, nos permite implantar fácilmente la señal de "start", indicada en el diagrama de la Fig. 11.1. La entrada de *reset* de los *flip flops* está en la fila R3. Como se indica en la Fig. 11.14, la señal de "start" ingresa al módulo por C9 y debe ser activa en "1" (alto), para esto se coloca una celda "1" en (R3:C9). La celda "1" recibe la señal "start" en C9 y trabaja como un inversor con su salida en R3, controlando de esta manera las entradas de *reset* de todos los *flip flops*. Es conveniente también tener control de cada una de las filas (de R4 a R9) con la señal de "start", para esto se incluye en cada fila una celda "0", de tal manera que solamente si la señal de *start* está en "0" el circuito opera normalmente.

	A <sub>4</sub>	A <sub>3</sub>	A <sub>2</sub>	D <sub>2</sub>	A <sub>1</sub>	D <sub>1</sub>	D <sub>0</sub>	A <sub>0</sub>	start	
								I	I	R10
	1	I	0	+	1			0	0	R9
	I	1	1	+	0		+	0	0	R8
		1			0	+		0	0	R7
	1		0		0	+		0	0	R6
		0			1		+	0	0	R5
	0	1			0		+	0	0	R4
				$\bar{Q}$ Q D	$\bar{Q}$ Q D		D $\bar{Q}$ Q		1	R3
										R2
				D2 M=	D1 M=4		D M=5			R1
	4									R0
	C0	C1	C2	C3	C4	C5	C6	C7	C8	C9

FIGURA 11.14 Inclusión de la señal de start.

Añadiendo una celda "I" en C9 para la señal de *start* y colocando interrupciones a lo largo de toda la periferia del módulo, excepto en donde se tienen las 5 señales de entrada (de selección ( $A_4$  y  $A_3$ ), de datos ( $A_0$ ), de *start* y reloj (CLK)) y la señal de salida hacia la MEF(2), el diseño está concluido. Se debe proceder ahora a describir el diseño con TILER, para luego correr SIMPPLEX, los datos que ofrece la pantalla de SIMPPLEX se presenta en la Fig. 11.15.

```
SIMPPLEX Version 4.8. Copyright 1986-91. Bonneville Microelectronics Inc.
Loading SCMOS20T.SDB database...
Loading circuit from file mef(1).ppl...
Finding circuit context...
Tracing circuits connections...
Creating extract file mef(1).ppx...

FATAL ERRORS:  0 (simulation allowed)
ERRORS:       5 (5 ignored)
WARNINGS:     0

Device statistics
PPL rows:     11
PPL columns:  10
Transistors:  138
Internal nodes: 40
Input pointers: 5/85
Output pointers: 6/85
Total pads:   0
```

FIGURA 11.15 Pantalla de SIMPPLEX al ejecutarlo sobre MEF(1).ppl

Se presentan a continuación el archivo (.prt), que describe las celdas utilizadas y el nombre de los nodos asignados con TILER. Para analizar el detalle de las interrupciones y el camino que siguen las diferentes señales, se presenta también la impresión del archivo (.prt) generado para el modo de trabajo *ZOOM-IN* (Fig. 11.17).

```

      0 1 2 3 4 5 6 7 8 9
10:  _____ I I!
 9: |1 I 0 + 1 0 0!
 8: |I 1 1 + 0 + 0 0!
 7: | 1 0 + 0 0!
 6: |1 0 0 + 0 0!
 5: | 0 1 + 0 0!
 4: |0 1 0 + 0 0!
 3: | _____ " " " " " " 1!
 2: | " " " " " " " " |
 1: | " " D " D " D " |
 0: | 4 " |

```

FIGURA 11.16 Impresión del archivo mef(1).prt.

La lista de nodos entregada por cualquiera de los archivos (.prt) es:

```

A10 [10,7,0]
A11 [9,4,0]
A12 [9,2,0]
A13 [9,1,0]
A14 [9,0,0]
A21 [1,7,0]
CLK [0,0,4]
D10 [9,6,0]
D11 [9,5,0]
D12 [9,3,0]
START [10,8,0]

```

El diseño de la MEF(1) presentado puede modificarse de acuerdo a como se estructure el diseño completo del codificador. Al unir la MEF(1) con la MEF(2), nuevamente se hará uso de las propiedades de PPL para variar los diseños, con el objeto de optimizar el área requerida. Los esquemas impresos con CAPFAST se presentan para los diseños completos, del codificador y del decodificador (incluidos los PADS).

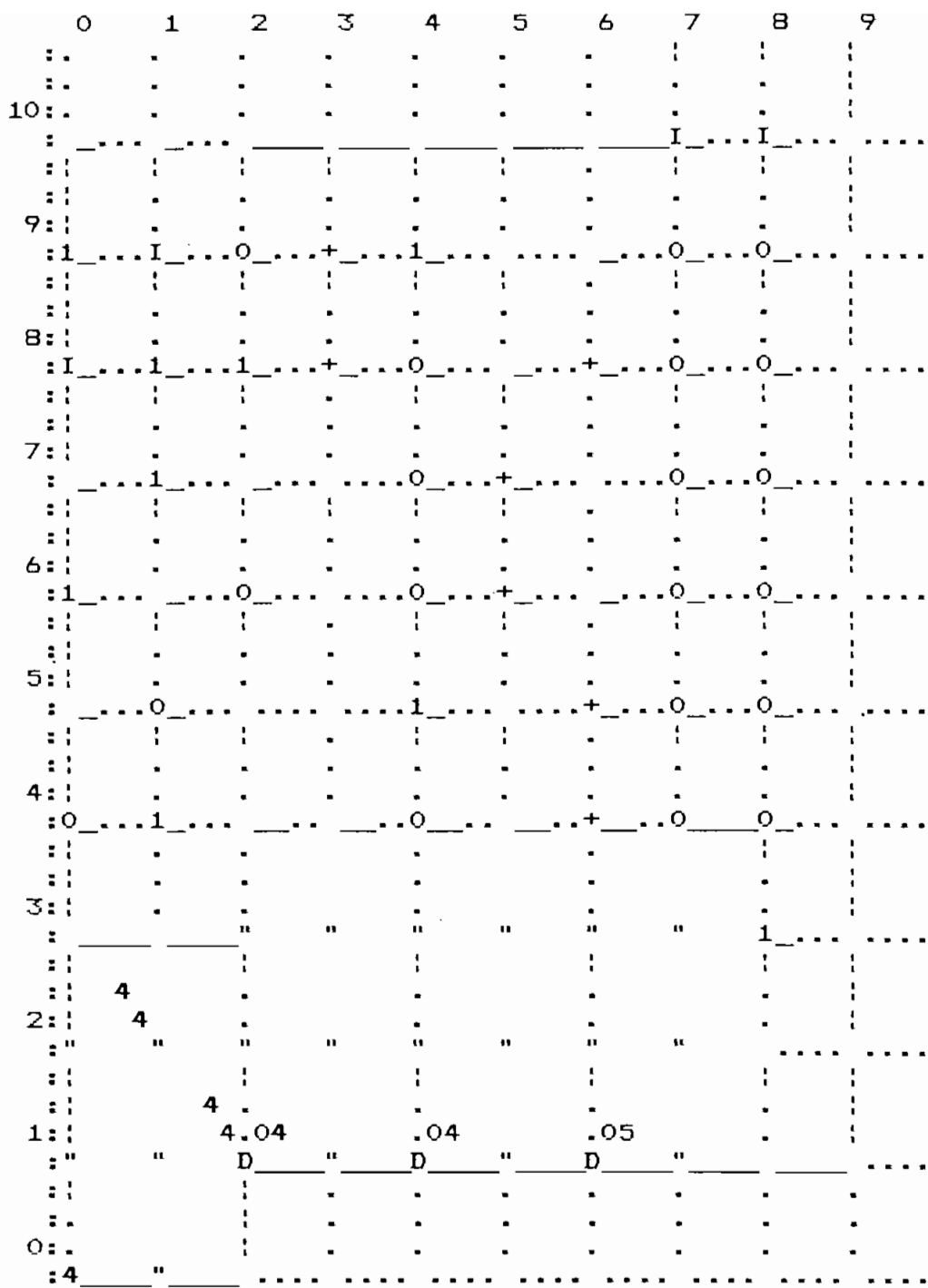


FIGURA 11.17 Impresión del archivo (.prt) para el modo ZOOM-IN.

### 11.1.2 MEF(2)

La misma secuencia de diseño seguida para la MEF(1) se sigue para la MEF(2), por lo que se presenta el diseño de manera resumida.

Las ecuaciones planteadas para la MEF(2) son:

$$D_0 = A_3 \bar{A}_1 \bar{A}_0 + \bar{A}_3 A_0$$

$$D_4 = A_2 \bar{A}_1 \bar{A}_0 + \bar{A}_3 \bar{A}_2 A_1 + A_3 A_2 \bar{A}_0 + \bar{A}_2 A_0$$

$$D_3 = A_2 A_0 + \bar{A}_3 A_2 A_1 + A_3 \bar{A}_2 A_1 \bar{A}_0$$

$$D_2 = \bar{A}_3 \bar{A}_2 A_1 + A_3 A_2 A_1 \bar{A}_0 + \bar{A}_2 A_0$$

$$D_1 = \bar{A}_3 A_2 A_1 \bar{A}_0$$

$$D_0 = \bar{A}_3 \bar{A}_2 A_1 \bar{A}_0$$

La solución planteada para cada una de las ecuaciones ( $D_0$  a  $D_3$ ) se presenta en la Fig. 11.18. Se incluye únicamente una fila de celdas "I".

A partir de la Fig. 11.18 se eliminan aquellas filas repetitivas, fusionándolas en una sola fila.

	A <sub>3</sub>	A <sub>2</sub>	A <sub>1</sub>	A <sub>0</sub>	D <sub>5</sub>	D <sub>4</sub>	D <sub>3</sub>	D <sub>2</sub>	D <sub>1</sub>	D <sub>0</sub>	
	I	I	I	I							R14
$A_3\bar{A}_1\bar{A}_0$	1		0	0	+						R13
$\bar{A}_3A_0$	0			1	+						R12
$A_2\bar{A}_1\bar{A}_0$		1	0	0		+					R11
$\bar{A}_3\bar{A}_2A_1$	0	0	1			+					R10
$A_3A_2\bar{A}_0$	1	1		0		+					R9
$\bar{A}_2A_0$		0		1		+					R8
$A_2A_0$		1		1			+				R7
$\bar{A}_3A_2A_1$	0	1	1				+				R6
$A_3\bar{A}_2A_1\bar{A}_0$	1	0	1	0			+				R5
$\bar{A}_3\bar{A}_2A_1$	0	0	1					+			R4
$A_3A_2A_1\bar{A}_0$	1	1	1	0				+			R3
$\bar{A}_2A_0$		0		1				+			R2
$\bar{A}_3A_2A_1\bar{A}_0$	0	1	1	0					+		R1
$\bar{A}_3\bar{A}_2A_1\bar{A}_0$	0	0	1	0						+	R0
	C0	C1	C2	C3	C4	C5	C6	C7	C8	C9	C10

FIGURA 11.18 Distribución de celdas para las ecuaciones de la MEF(2).

Luego de unificar las filas (R10 y R4) y (R8 y R2), se utiliza TILER para generar el archivo (.ppl) y se sigue el proceso hasta simularlo. Los archivos (.src) y (.cpy) son:

ARCHIVO (.SRC)

```

;SIMULACION DEL CIRCUITO COMBINACIONAL DE LA MEF(2) DEL CODIFICADOR
COPY
VECTOR IN A23 A22 A21 A20
VECTOR OUT D25 D24 D23 D22 D21 D20
OPTIONS TABLES -COPYINPUTS
WATCH IN OUT

```



```
***ESTADO qi=00***  
SET IN=0000  
CYCLE  
SET IN=0001  
CYCLE  
SET IN=0010  
CYCLE  
SET IN=0011  
CYCLE
```

```
***ESTADO qi=01***  
SET IN=0100  
CYCLE  
SET IN=0101  
CYCLE  
SET IN=0110  
CYCLE  
SET IN=0111  
CYCLE
```

```
***ESTADO qi=10***  
SET IN=1000  
CYCLE  
SET IN=1001  
CYCLE  
SET IN=1010  
CYCLE  
SET IN=1011  
CYCLE
```

```
***ESTADO qi=11***  
SET IN=1100  
CYCLE  
SET IN=1101  
CYCLE  
SET IN=1110  
CYCLE  
SET IN=1111  
CYCLE
```

#### ARCHIVO (.CPY)

```
>> VECTOR IN A23 A22 A21 A20  
>> VECTOR OUT D25 D24 D23 D22 D21 D20  
>> OPTIONS TABLES -COPYINPUTS  
1> 0000 000000  
2> 0001 110100  
3> 0010 010101  
4> 0011 110100  
5> 0100 010000  
6> 0101 101000  
7> 0110 001010  
8> 0111 101000  
9> 1000 100000  
10> 1001 010100  
11> 1010 001000  
12> 1011 010100  
13> 1100 110000  
14> 1101 001000  
15> 1110 010100  
16> 1111 001000
```

Los valores de la tabla del archivo (.cpy) son idénticos a los planteados en la Tabla 9.17 y se concluye que el diseño del CC. está correcto.

Una vez realizado el diseño del CC., simplemente se deben incluir los *flip flops* de realimentación necesarios. Primeramente se cambia el orden de algunas columnas, como se indica en la Fig. 11.19.

	A <sub>1</sub>	A <sub>0</sub>	A <sub>3</sub>	D <sub>5</sub>	A <sub>2</sub>	D <sub>4</sub>	D <sub>1</sub>	D <sub>0</sub>	D <sub>3</sub>	D <sub>2</sub>	
I	I	I		I							R12
0	0	1	+								R11
	1	0	+								R10
0	0			1	+						R9
1		0		0	+					+	R8
	0	1		1	+						R7
	1			0	+					+	R6
	1			1					+		R5
1		0		1					+		R4
1	0	1		0					+		R3
1	0	1		1					+		R2
1	0	0		1		+					R1
1	0	0		0				+			R0
	C0	C1	C2	C3	C4	C5	C6	C7	C8	C9	C10

FIGURA 11.19 Intercambio de columnas del CC. de la MEF(2) simplificada.

	start	A <sub>1</sub>	A <sub>0</sub>	A <sub>3</sub>	D <sub>0</sub>	A <sub>2</sub>	D <sub>4</sub>	D <sub>1</sub>	D <sub>0</sub>	D <sub>3</sub>	D <sub>2</sub>	
I												R16
0	0	0	1	+								R15
0	I	1	0	+								R14
0	0	0				1	+					R13
0	1	I	0			0	+				+	R12
0		0	1			1	+					R11
0		1				0	+				+	R10
0		1				1					+	R9
0	1		0			1					+	R8
0	1	0	1			0					+	R7
0	1	0	1			1					+	R6
0	1	0	0			1		+				R5
0	1	0	0			0			+			R4
1			Q̄ Q D		Q̄ Q D							R3
	4		D2 M=4		D1 M=4							R2
												R1
	4											R0
	C0	C1	C2	C3	C4	C5	C6	C7	C8	C9	C10	

FIGURA 11.20 MEF(2) con los flip flops, celda "4" y señal de start.

En la Fig. 11.20 se presenta el CC. de la MEF(2), con los flip flops de realimentación, la celda "4" y la señal de start incluidos.

Se debe proceder ahora a describir el diseño con TILER, para luego correr SIMPPLEX, los datos que ofrece la pantalla

de SIMPPLEX al ejecutarlo sobre el archivo MEF(2).ppl se presenta en la Fig. 11.21.

```
SIMPPLEX Version 4.8. Copyright 1986-91. Bonneville Microelectronics Inc.
Loading SCMOS207.SDB database...
Loading circuit from file mef(2).ppl...
Finding circuit context...
Tracing circuits connections...
Creating extract file mef(2).ppx...

FATAL ERRORS:  0 (simulation allowed)
ERRORS:       4 (4 ignored)
WARNINGS:    0

Device statistics
PPL rows:    17
PPL columns: 12
Transistors: 192
Internal nodes: 43
Input pointers: 5/100
Output pointers: 12/100
Total pads:  0
```

FIGURA 11.21 Pantalla de SIMPPLEX al ejecutarlo sobre MEF(2).ppl.

Se presentan a continuación los dos archivos (.prt) generados para la MEF(2):

La lista de nodos entregada por cualquiera de los archivos (.prt) es:

```
A20 [15,2,0]
A21 [15,1,0]
A22 [15,5,0]
A23 [15,3,0]
CLK [0,0,4]
D20 [15,8,0]
D21 [15,7,0]
D22 [15,10,0]
D23 [15,9,0]
D24 [15,6,0]
D25 [15,4,0]
START [16,0,0]
```

```

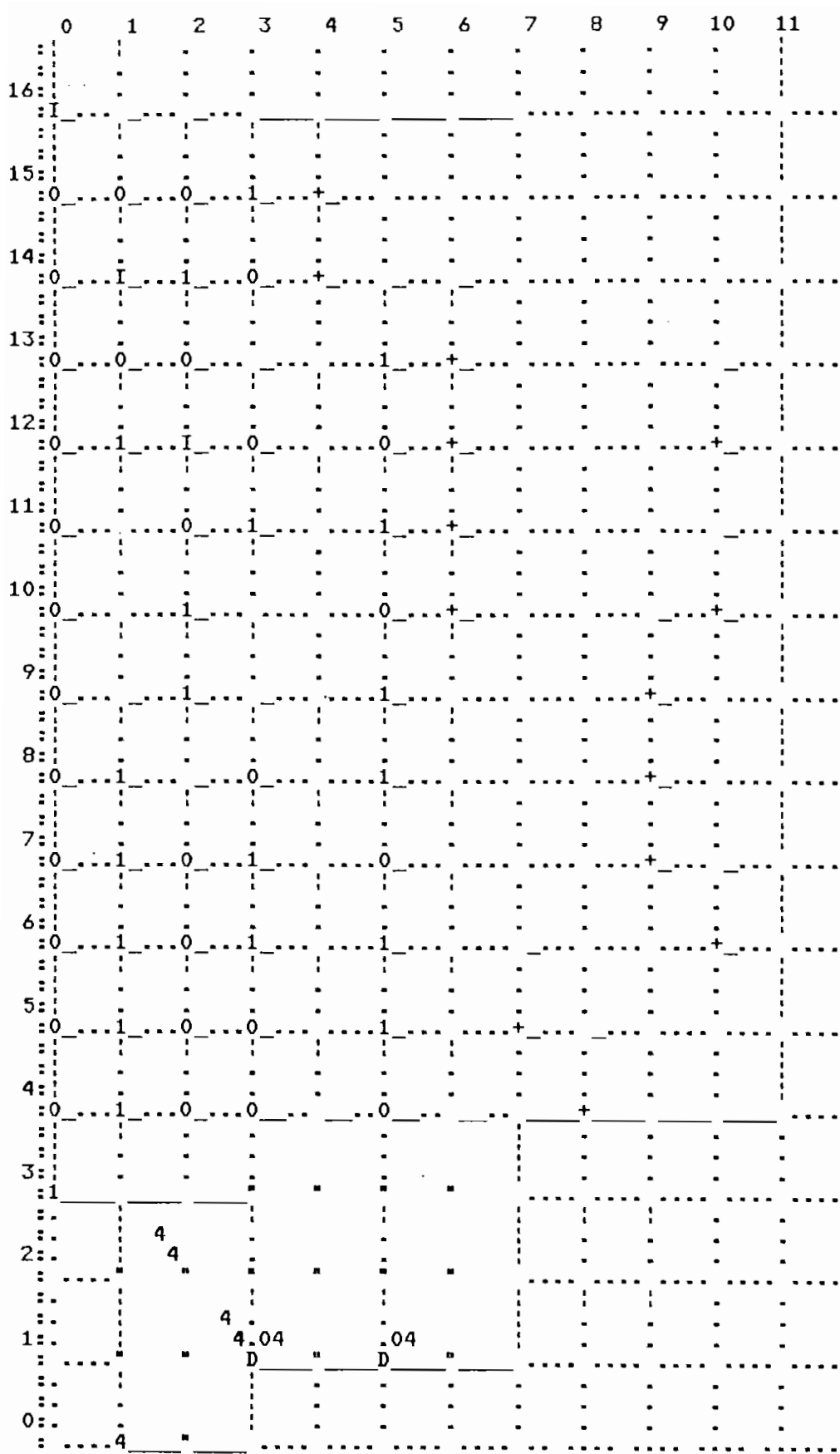
          0 1 2 3 4 5 6 7 8 9 0 1
16:|I
15:|0 0 0 1 +
14:|0 1 1 0 +
13:|0 0 0      1 +
12:|0 1 1 0 0 +
11:|0 0 1 1 +
10:|0 1 0 0 +
 9:|0 1 1      +
 8:|0 1 0 1      +
 7:|0 1 0 1 0      +
 6:|0 1 0 1 1      +
 5:|0 1 0 0 1      +
 4:|0 1 0 0 0      +
 3:|1 " " " " |
 2:| " " " " " " |
 1:| " " D " D " |
 0:| 4 " |

```

FIGURA 11.22 Impresión del archivo (.prt)

En la siguiente página se presenta la Fig. 11.23, impresión del archivo MEF(2).prt para el modo de trabajo ZOOM-IN. Por falta de espacio, no se incluye el nombre de la figura al final de la misma.

El circuito de la Fig. 11.20 puede simplificarse, utilizando dos columnas menos. Para esto, se utiliza la celda "s", en combinación con la celda "+". En la Fig. 11.24 se muestran las diferentes representaciones de las celdas "+" y "s", con modificadores M=0, que son las utilizadas en el diseño.



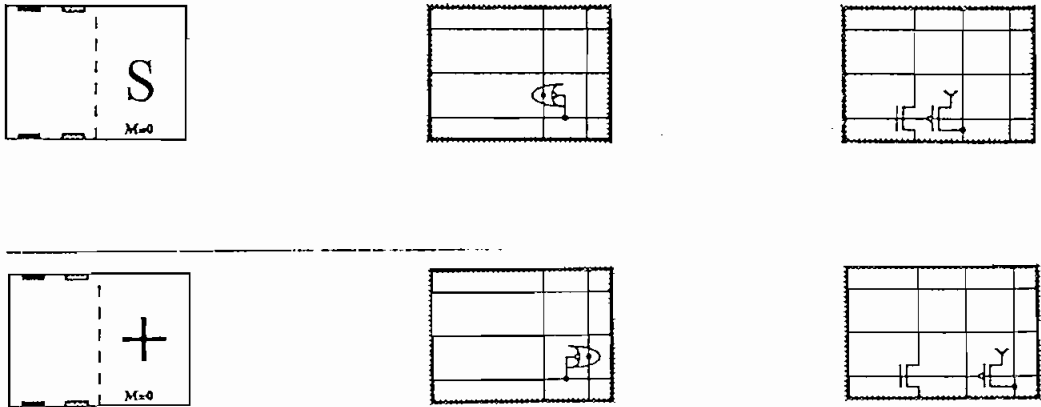
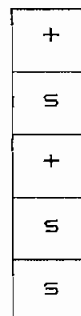


FIGURA 11.24 Celdas '+' y 's'

De la figura está claro que las dos celdas se pueden utilizar para los mismos propósitos, con la diferencia que la celda "+" entrega la señal de salida en RCOL, mientras que la "s" lo hace en LCOL. Partiendo de esta característica se pueden tener celdas "s" y "+" en una misma columna, por ejemplo realizando las funciones OR de conjuntos diferentes de señales. Esta característica se aplica a la MEF(2), integrando (D3, D1) y (D2, D0) lo que resulta en el arreglo de la Fig. 11.25. Sin embargo, es necesario mencionar, de pruebas realizadas, que no se deben intercalar celdas "+" con celdas "s"; por ejemplo no se debe tener la distribución:



Para el caso presentado, se forma una compuerta NAND entre las dos celdas "s" de la parte inferior y la celda "s" aislada forma un inversor, las dos compuertas con su salida respectiva en el camino LCOL, que electricamente no es permitido.

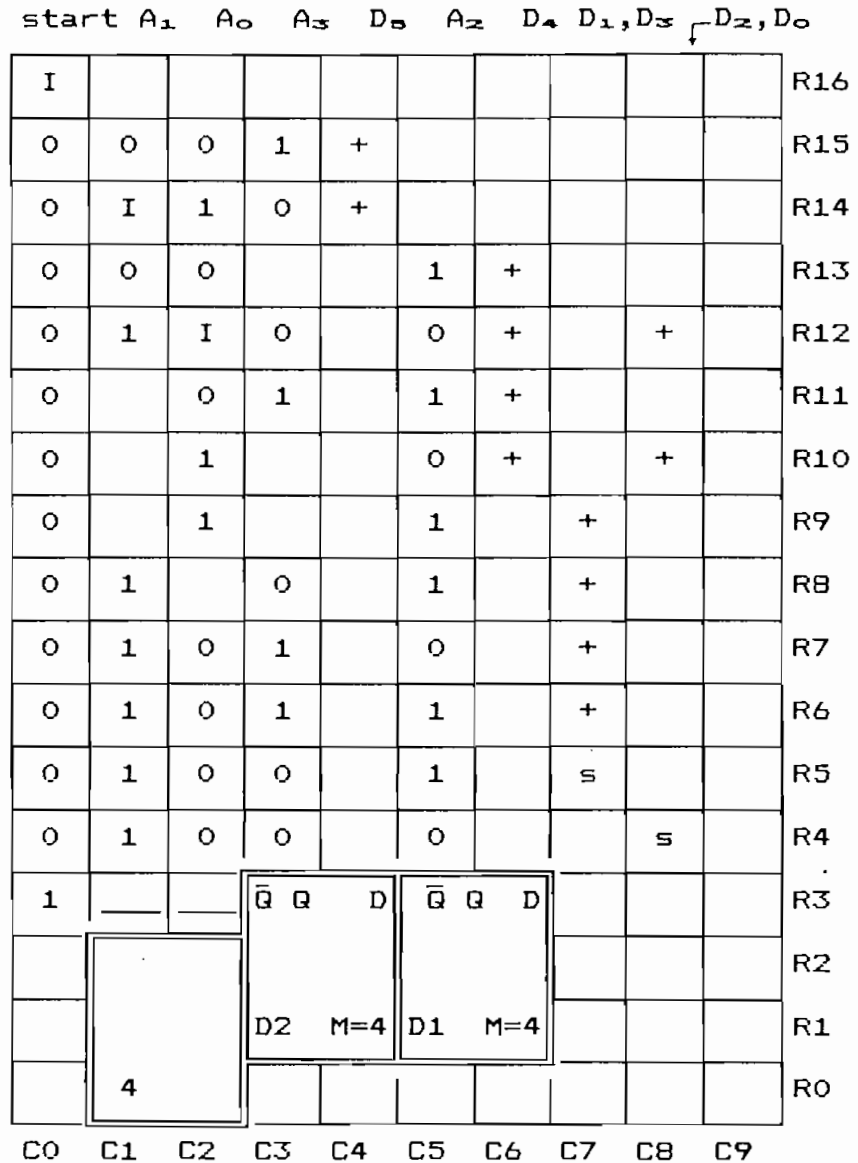


FIGURA 11.25 MEF(2) con celdas "s"



### 11.1.3 Multiplexores

Para el diseño de los multiplexores se puede utilizar únicamente las celdas básicas; sin embargo, con el objetivo de utilizar celdas diseñadas para este propósito, que a la vez llevan al ahorro de área, se utilizan las celdas "h", (multiplexores de 2 a 1), y celdas "g" (necesarias para la operación de las celdas "h").

Al integrar los diferentes módulos del codificador completo, para optimizar el área total utilizada fue necesario no solo la estructura del multiplexor con celdas "h", sino otra diferente basada en celdas "j".

#### a) Multiplexor de 4 a 1 con celdas "h"

En la Fig. 11.26 se presentan las dos celdas "h" disponibles, de las cuales se utilizó la celda con  $M=1$ , que no invierte la señal de salida respecto a las de entrada. La celda "h" es un multiplexor de 2 a 1, con "Sel" se determina cual de las señales de entrada ( $In0$  e  $In1$ ) pasa a la salida (out). La señal de salida está disponible en RCOL y LCOL, y de acuerdo a la aplicación, puede ser necesario interrumpir uno de los dos caminos. Cuando  $Sel=0$ , se selecciona la señal  $In1$  y cuando  $Sel=1$ , se selecciona  $In0$  (lógica negativa en las filas).

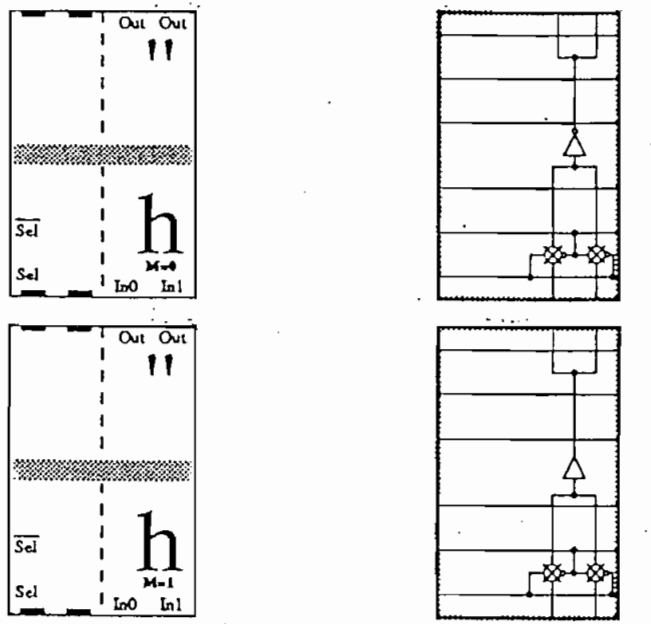


FIGURA 11.26 Celdas "h" con M=0 y M=1.

Dado que el multiplexor se basa en compuertas de transmisión, se requiere la señal "Sel" y su complemento. Para cumplir este requerimiento, se utiliza la celda "9" con M=0, que es un inversor con entrada en ROW y salida SROW, como se indica en la Fig. 11.27.

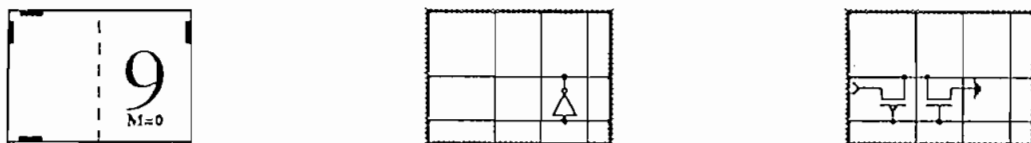


FIGURA 11.27 Celda "9" .

Utilizando las celdas "h" y "9", se puede diseñar multiplexores de "n" a 1. Para los diseños del codificador y decodificador se requieren multiplexores de 4 a 1. En la Fig. 11.28a se presenta la distribución de las celdas necesarias para obtener un multiplexor de 4 a 1. Debe notarse que el multiplexor diseñado tiene las señales de selección en las filas y las entradas y salida en las columnas (Fig. 11.28b). Las señales de entrada van de abajo hacia arriba, si se requiere que las señales pasen de arriba hacia abajo se utiliza en lugar de las celdas "h" las celdas "H".

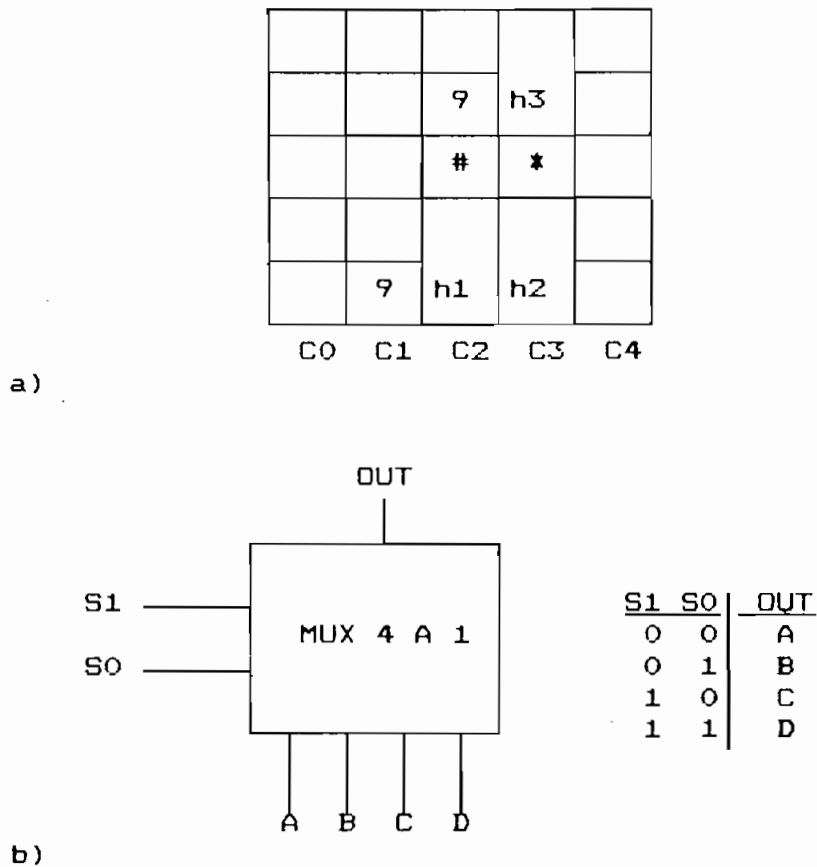


FIGURA 11.28 Multiplexor de 4 a 1 con celdas "h".

En la Fig. 11.28a se utilizan las celdas "#" y "\*" para enrutar la señal de salida de h1 hacia h3, por lo que debe interrumpirse el paso de una de las salidas de h2 hacia h3.

A continuación se presenta el archivo (.prt) generado para el modo ZOOM-IN (Fig. 11.29) y los resultados de la simulación de este módulo.

```

      0      1      2
      :      .      .
      :      .      .
      4:      .      .
      :      .      .
      :      .      .
      :      .      .
      3:      .      .01
      :      .      .9___h___
      :      .      .
      :      .      .
      :      .      .
      2:      .      .#   *
      :      .      .
      :      .      .
      :      .      .
      1:      .      .
      :      .      .
      :      .      .
      :      .      .
      0:      .01   .01
      :      .9___h___h___
  
```

**Lista de nodos:**

```

A [0,2,0]
B [0,2,1]
C [0,1,0]
D [0,1,1]
OUT [4,2,0]
S0 [0,0,4]
S1 [3,1,4]
  
```

FIGURA 11.29 Multiplexor de 4 a 1 con celdas "h" en Modo ZOOM-IN.

ARCHIVO (.CPY)

```
>> vector IN A B C D
>> vector SELECT S1 S0
>> watch IN SELECT OUT
>> set IN=1010
>> set SELECT=00
>> cycle
    1> IN=1010 SELECT=00 OUT=1
>> set SELECT=01
>> .cycle
    2> IN=1010 SELECT=01 OUT=0
>> set SELECT=10
>> cycle
    3> IN=1010 SELECT=10 OUT=1
>> set SELECT=11
>> cycle
    4> IN=1010 SELECT=11 OUT=0
End of file: MUX.src
sim> QUIT
```

b) Multiplexor de 4 a 1 con celdas "j"

En la Fig. 11.30 se muestran las tres representaciones de la celda "j".

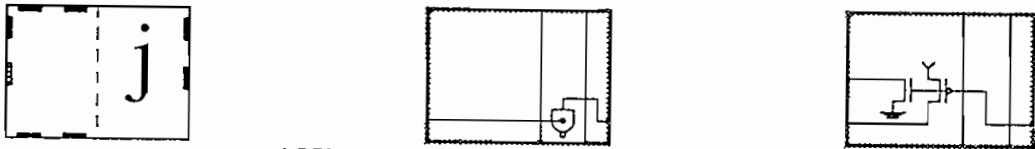
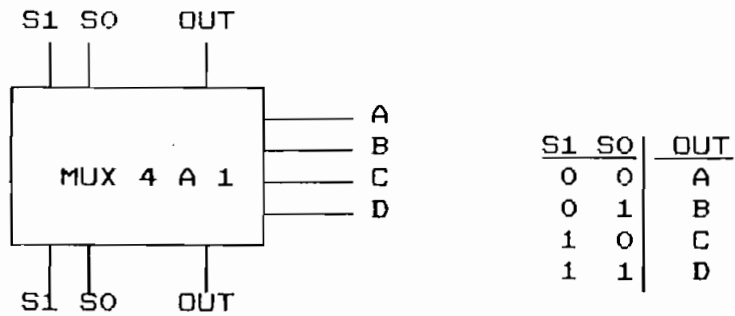


FIGURA 11.30 Celda "j".

La celda "j" en combinación con celdas "I", "0" y "1" estructuran un multiplexor con las entradas en los caminos horizontales y con las líneas de selección y salida en los caminos verticales, lo que se representa en la Fig. 11.31a.



a)

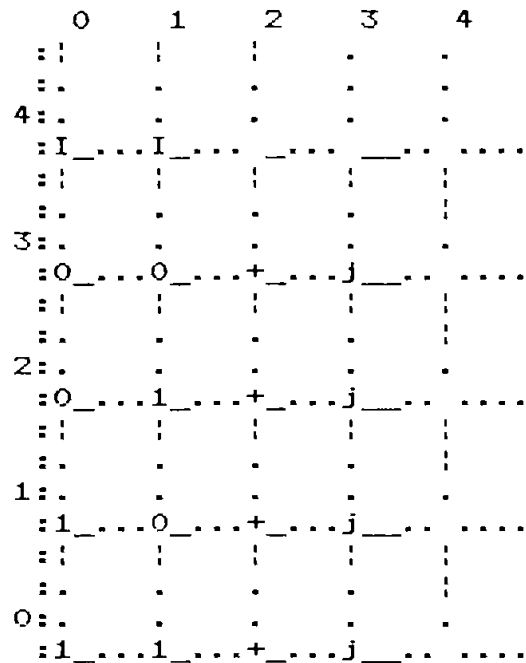
	I	I		
	0	0	+	j
	0	1	+	j
	1	0	+	j
	1	1	+	j
C0	C1	C2	C3	C4

b)

FIGURA 11.31 Multiplexor de 4 a 1 con celdas "j".

En la Fig. 11.31b se presenta la disposición de celdas necesarias para cumplir con el objetivo planteado. Si se desea que las señales de entrada ingresen al módulo por el lado izquierdo, se utilizan celdas "w" en lugar de las celdas "j", pero colocadas a la izquierda del módulo. Las columnas de celdas "0", "1" y "+" pueden ser intercambiadas a voluntad sin alterar el funcionamiento del circuito.

En la Fig. 11.32 se presenta el archivo (.prt) para el modo *ZOOM-IN* y los resultados de la simulación de este módulo.



LISTA DE NODOS:

```

A [3,4,4]
B [2,4,4]
C [1,4,4]
D [0,4,4]
OUT [4,2,0]
S0 [4,1,0]
S1 [4,0,0]

```

FIGURA 11.32 Multiplexor de 4 a 1 con celdas 'j' en Modo ZOOM-1K.

```

ARCHIVO .CPY
>> vector IN A B C D
>> vector SELECT S1 S0
>> watch IN SELECT OUT
>> set IN=1010
>> set SELECT=00
>> cycle
    1> IN=1010 SELECT=00 OUT=1
>> set SELECT=01
>> cycle
    2> IN=1010 SELECT=01 OUT=0
>> set SELECT=10
>> cycle
    3> IN=1010 SELECT=10 OUT=1
>> set SELECT=11
>> cycle
    4> IN=1010 SELECT=11 OUT=0
End of file: MUX.src

```

#### 11.1.4 Integración de los módulos del codificador

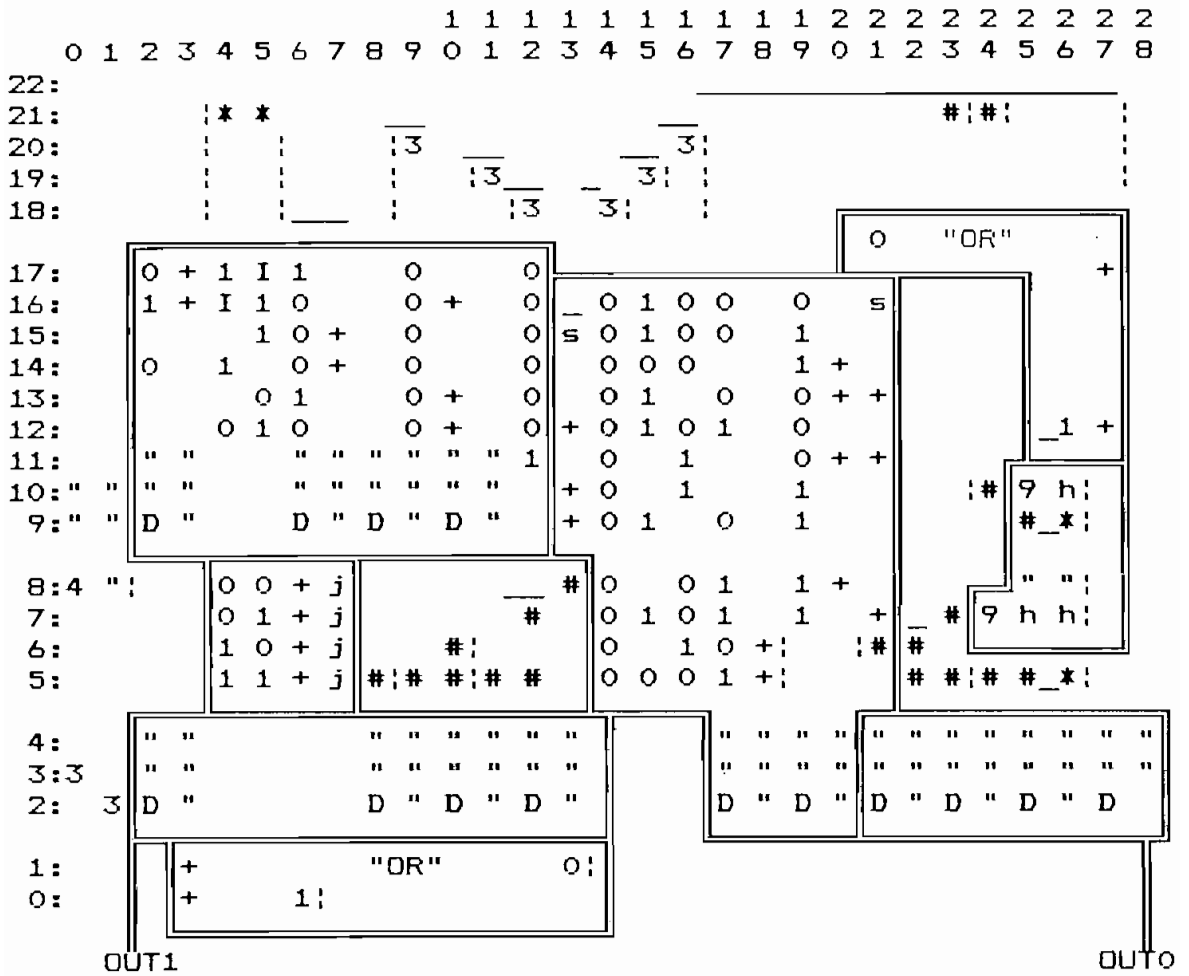
Una vez diseñados y simulados los módulos independientemente, se procede a integrarlos en un solo circuito. En la Fig. 11.33 se presenta el archivo (.prt) que se obtiene luego de haber integrado todos los módulos. La figura se ha dividido en bloques mediante líneas, para lo cual ha sido necesario eliminar algunas celdas de interconexión y símbolos de interrupciones; posteriormente se presentan otras figuras completas más detalladas.

En la Fig. 11.33 se incluye también la lista de nodos, manteniendo la notación de direcciones y datos utilizada en los otros esquemas. Se han incluido los nombres DATO para la señal de datos a codificar antes del *flip flop* de entrada (FFb de la Fig. 11.1), los nombres OUT0 y OUT1 para las señales de salida del codificador, y los nombres NO y N1 como alternativos para las señales de selección del código.

De los dos bloques más grandes, el de la izquierda corresponde a la MEF(1) y el de la derecha a la MEF(2). El bloque de la MEF(1) incluye los *flip flops* para la entrada de datos a codificar y el *flip flop* que se encuentra entre la MEF(1) y la MEF(2) (FFb y FFa de la Fig. 11.1).

Puede apreciarse claramente los multiplexores que utilizan celdas "h" y celdas "j".





Lista de nodos:

- |               |                 |
|---------------|-----------------|
| A10 [17,9,0]  | A20 [16,16,0]   |
| A11 [17,6,0]  | A21 [16,15,0]   |
| A12 [17,2,0]  | A22 [16,19,0]   |
| A13 [17,5,0]  | A23 [16,17,0]   |
| A14 [17,4,0]  |                 |
|               |                 |
| D10 [17,10,0] |                 |
| D11 [17,7,0]  |                 |
| D12 [17,3,0]  |                 |
|               |                 |
| D20 [16,21,1] | D21 [15,13,1]   |
| D22 [16,21,0] | D23 [15,13,0]   |
| D24 [16,20,0] | D25 [16,18,0]   |
|               |                 |
| CLK [8,0,4]   | START [18,14,0] |
| DATO [11,8,0] | OUT0 [2,28,0]   |
| OUT1 [2,2,0]  | NO [21,5,0]     |
| N1 [21,4,0]   |                 |

FIGURA 11.33 Distribución de los bloques que conforman el Codificador.

El multiplexor con celdas "h" se combina con los cuatro *flip flops* y el bloque "OR" próximos a el, para entregar la salida "OUTO".

El multiplexor con celdas "j" se combina con los cuatro *flip flops* y bloque OR próximos a el, para entregar la salida OUT1.

Observando la distribución de celdas de las dos MEFs, se puede apreciarse que se ha intercambiado algunas filas y columnas, respecto a las planteadas anteriormente.

Debe notarse también que se utiliza una sola celda "4" (M=1), con las salidas de esta celda se proporcionan las señales de reloj a todos los *flip flops* del circuito.

Las compuertas "OR" para la salida OUTO y OUT1 se estructuran utilizando compuertas NAND con las entradas negadas. Para la salida OUTO, una celda "0" toma la señal de salida D20 de la MEF(2), que se obtiene con una celda "s" (salida en LCOL), una celda "1" que toma la señal de salida del multiplexor con celdas "h" y un par de celdas "+".

Para la salida OUT1, una celda "0" toma la señal de salida D21 de la MEF(2), que se obtiene con una celda "s", una celda "1" que toma la señal del multiplexor con celdas "j" y un par de celdas "+".

En la Fig. 11.34 se presenta el archivo (.prt) del codificador incluidas las interrupciones y sin la división en bloques.

En el Anexo H se incluyen el esquema del archivo (.prt) para el modo *ZOOM-IN* del codificador, así como un diagrama generado con CAPFAST a nivel lógico (incluidos los PADS) en el que se aprecian claramente el enrutamiento de las señales.

La pantalla presentada por SIMPPLEX para el archivo *codi.ppl* se presenta en la Fig. 11.35.

```

Circuit: NOMBRE \(\SCMOS20T\)      Date: Mon Sep 07 23:17:03 1992

          1 1 1 1 1 1 1 1 1 1 2 2 2 2 2 2 2 2
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8
22:
21:          * *                               #|#|
20:          |3|                               |3|
19:          |3| |3| |3|
18:          |3| |3|
17:  |0 + 1 I 1      0      0| I| |0
16:  |1 + I 1 0      0 +  0| 0 1 0 0      0 s|
15:          | 1 0 +  0      0|s 0 1 0 0      1
14:  |0      1  0 +  0      0| 0 0 0      1 +
13:          |  0 1      0 +  0| 0 1  0  0 + +
12:          |  0 1 0      0 +  0|+ 0 1 0 1  0
11:  |" "      " " " " " " | 1| 0  1      0 + +
10:  |" " " "      " " " " " " | + 0  1      1 |# 9 h|
 9:  |" " D "      D " D " D " | + 0 1  0  1
 8:  |4 " | 0 0 + j      #| 0  0 1  1 +
 7:  |  0 1 + j      # | 0 1 0 1  1 + # 9 h h|
 6:  |  1 0 + j      #| | 0  1 0 +; |# # |3|
 5:  |  1 1 + j      #|# #|# # | 0 0 0 1 +; |# #|# #*|
 4:  |" "      " " " " " " | 1| " " " " " " " " " " " "
 3:  |3 " "      " " " " " " | " " " " " " " " " " " "
 2:  |3 D "      D " D " D " | D " D " D " D " D " D "
 1:  | +      0|
 0:  | +      1|

```

FIGURA 11.34 Distribución de celdas e interrupciones del Codificador.

```
SIMPLEX Version 4.8. Copyright 1986-91. Bonneville Microelectronics Inc.  
Loading SCMOS20T.SDB database...  
Loading circuit from file codi.ppl...  
Finding circuit context...  
Tracing circuits connections...  
Creating extract file codi.ppx...
```

```
FATAL ERRORS: 0 (simulation allowed)  
ERRORS:      5 (5 ignored)  
WARNINGS:    0
```

```
Device statistics  
PPL rows:    23  
PPL columns: 30  
Transistors: 568  
Internal nodes: 118  
Input pointers: 8/330  
Output pointers: 20/318  
Total pads: 0
```

FIGURA 11.35 Pantalla obtenida al ejecutar SIMPLEX sobre el archivo codi.ppl.

Para simular el circuito total del codificador se introduce una secuencia de datos, y se observa los códigos generados en las salidas OUT0 y OUT1, de acuerdo a lo especificado en las líneas de selección.

Puesto que ahora se incluye la celda "4" para generar las señales de reloj, se debe especificar en el archivo de simulación la señal de reloj "CLK" nombrada en el archivo (.ppl). Para esto se utiliza el comando *CLOCK* del simulador.

#### ARCHIVO .SRC

```
COPY  
VECTOR SELECT N1 NO  
VECTOR OUT OUT1 OUT0  
CLOCK CLK:10  
WATCH START SELECT DATO OUT  
OPTIONS -COPYINPUTS
```

```
SET START=1
CYCLE
SET      SELECT=00 DATO=1
CYCLE  2
```

```
***** AMI *****
SET START=0 SELECT=00
SET DATO=1
CYCLE
SET DATO=0
CYCLE
SET DATO=1
CYCLE 3
SET DATO=0
CYCLE 3
SET DATO=1
CYCLE 3
```

```
***** HDB1 *****
SET START=1
CYCLE
SET START=0 SELECT=01
CYCLE
SET      DATO=1
CYCLE
SET      DATO=0
CYCLE
SET      DATO=1
CYCLE  2
SET      DATO=0
CYCLE  2
SET      DATO=1
CYCLE  2
SET      DATO=0
CYCLE  3
SET      DATO=1
CYCLE  3
```

```
***** HDB2 *****
SET START=1
CYCLE
SET START=0 SELECT=10
CYCLE
SET      DATO=1
CYCLE
SET      DATO=0
CYCLE
SET      DATO=1
CYCLE  2
SET      DATO=0
CYCLE  2
SET      DATO=1
CYCLE  2
SET      DATO=0
CYCLE  4
```

```

SET      DATO=1
CYCLE   4

;***** HDB3 *****
SET START=1
CYCLE
SET START=0 SELECT=11
CYCLE
SET      DATO=1
CYCLE
SET      DATO=0
CYCLE
SET      DATO=1
CYCLE  2
SET      DATO=0
CYCLE  2
SET      DATO=1
CYCLE  2
SET      DATO=0
CYCLE  3
SET      DATO=1
CYCLE  2
SET      DATO=0
CYCLE  5
SET      DATO=1
CYCLE  7

```

Los resultados de la simulación son satisfactorios, se debe considerar para analizar los resultados los ciclos de reloj que se retrasan los mismos debido a la presencia de los *flip flops* de la entrada y salida.

#### ARCHIVO .CPY

```

>> VECTOR SELECT N1 NO
>> VECTOR OUT OUT1 OUTO
>> CLOCK CLK:10
>> WATCH START SELECT DATO OUT
>> OPTIONS -COPYINPUTS

```

```

1:2> START=1 SELECT=XX DATO=X      OUT=00
2:2> START=1 SELECT=00 DATO=1     OUT=00
3:2> START=1 SELECT=00 DATO=1     OUT=00

4:2> START=0 SELECT=00 DATO=1     OUT=00
5:2> START=0 SELECT=00 DATO=0     OUT=01
6:2> START=0 SELECT=00 DATO=1     OUT=00
7:2> START=0 SELECT=00 DATO=1     OUT=10

```

8:2>	START=0	SELECT=00	DATO=1	OUT=01
9:2>	START=0	SELECT=00	DATO=0	OUT=10
10:2>	START=0	SELECT=00	DATO=0	OUT=00
11:2>	START=0	SELECT=00	DATO=0	OUT=00
12:2>	START=0	SELECT=00	DATO=1	OUT=00
13:2>	START=0	SELECT=00	DATO=1	OUT=01
14:2>	START=0	SELECT=00	DATO=1	OUT=10
15:2>	START=1	SELECT=00	DATO=1	OUT=00
16:2>	START=0	SELECT=01	DATO=1	OUT=00
17:2>	START=0	SELECT=01	DATO=1	OUT=00
18:2>	START=0	SELECT=01	DATO=0	OUT=01
19:2>	START=0	SELECT=01	DATO=1	OUT=10
20:2>	START=0	SELECT=01	DATO=1	OUT=00
21:2>	START=0	SELECT=01	DATO=0	OUT=01
22:2>	START=0	SELECT=01	DATO=0	OUT=10
23:2>	START=0	SELECT=01	DATO=1	OUT=01
24:2>	START=0	SELECT=01	DATO=1	OUT=01
25:2>	START=0	SELECT=01	DATO=0	OUT=10
26:2>	START=0	SELECT=01	DATO=0	OUT=01
27:2>	START=0	SELECT=01	DATO=0	OUT=10
28:2>	START=0	SELECT=01	DATO=1	OUT=10
29:2>	START=0	SELECT=01	DATO=1	OUT=00
30:2>	START=0	SELECT=01	DATO=1	OUT=01
31:2>	START=1	SELECT=01	DATO=1	OUT=00
32:2>	START=0	SELECT=10	DATO=1	OUT=00
33:2>	START=0	SELECT=10	DATO=1	OUT=00
34:2>	START=0	SELECT=10	DATO=0	OUT=00
35:2>	START=0	SELECT=10	DATO=1	OUT=01
36:2>	START=0	SELECT=10	DATO=1	OUT=10
37:2>	START=0	SELECT=10	DATO=0	OUT=00
38:2>	START=0	SELECT=10	DATO=0	OUT=01
39:2>	START=0	SELECT=10	DATO=1	OUT=10
40:2>	START=0	SELECT=10	DATO=1	OUT=00
41:2>	START=0	SELECT=10	DATO=0	OUT=00
42:2>	START=0	SELECT=10	DATO=0	OUT=01
43:2>	START=0	SELECT=10	DATO=0	OUT=10
44:2>	START=0	SELECT=10	DATO=0	OUT=01
45:2>	START=0	SELECT=10	DATO=1	OUT=00
46:2>	START=0	SELECT=10	DATO=1	OUT=01
47:2>	START=0	SELECT=10	DATO=1	OUT=00
48:2>	START=0	SELECT=10	DATO=1	OUT=10
49:2>	START=1	SELECT=10	DATO=1	OUT=00
50:2>	START=0	SELECT=11	DATO=1	OUT=00
51:2>	START=0	SELECT=11	DATO=1	OUT=00
52:2>	START=0	SELECT=11	DATO=0	OUT=00
53:2>	START=0	SELECT=11	DATO=1	OUT=00
54:2>	START=0	SELECT=11	DATO=1	OUT=01
55:2>	START=0	SELECT=11	DATO=0	OUT=10
56:2>	START=0	SELECT=11	DATO=0	OUT=00

57:2>	START=0	SELECT=11	DATO=1	OUT=01
58:2>	START=0	SELECT=11	DATO=1	OUT=10
59:2>	START=0	SELECT=11	DATO=0	OUT=00
60:2>	START=0	SELECT=11	DATO=0	OUT=00
61:2>	START=0	SELECT=11	DATO=0	OUT=01
62:2>	START=0	SELECT=11	DATO=1	OUT=10
63:2>	START=0	SELECT=11	DATO=1	OUT=00
64:2>	START=0	SELECT=11	DATO=0	OUT=00
65:2>	START=0	SELECT=11	DATO=0	OUT=00
66:2>	START=0	SELECT=11	DATO=0	OUT=01
67:2>	START=0	SELECT=11	DATO=0	OUT=10
68:2>	START=0	SELECT=11	DATO=0	OUT=01
69:2>	START=0	SELECT=11	DATO=1	OUT=00
70:2>	START=0	SELECT=11	DATO=1	OUT=00
71:2>	START=0	SELECT=11	DATO=1	OUT=01
72:2>	START=0	SELECT=11	DATO=1	OUT=00
73:2>	START=0	SELECT=11	DATO=1	OUT=10
74:2>	START=0	SELECT=11	DATO=1	OUT=01
75:2>	START=0	SELECT=11	DATO=1	OUT=10

Para evaluar el área utilizada por el codificador, se corre el programa PPLACE, cuyos resultados se presentan en la Fig. 11.36.

```

Registered to: University STUDENTS
(C) Copyright 1990 Bonneville Micro.
All Rights Reserved
PPLACE Version 1.0e, Copyright 1986-91 Bonneville Microelectronics Inc.
Reading file "codi.ppl"...
Loading cell set SCMOS20T...
Performing DRC verification...
Removing unused cell connections...
Processing file "exorchip.ppl"...
      size: 1500 X 506 (microns)
[PPLACE] 1 CIF Error written to codi.err

```

FIGURA 11.36 Resultados obtenidos al ejecutar PPLACE sobre codi.ppl

El área utilizada por el codificador, sin PADs, es de 1500  $\mu\text{m}$  X 506  $\mu\text{m}$ . El error reportado en el archivo *codi.err* se debe a que el módulo no se ajusta a las dimensiones del PAD-FRAME (*Pad frame is wrong size at row 22 col 29*).



## 11.2 DISEÑO DEL DECODIFICADOR PROGRAMABLE HDBn

El esquema del decodificador completo se muestra en la Fig. 11.37.

### 11.2.1 MEF(1)

Las ecuaciones del Cap.9 se plantearon agrupando términos en los mapas de Karnaugh, según algunas conveniencias para el diseño con TENTOS. Para el diseño con PPL se consideraron las ecuaciones con el menor número de variables en cada término AND, y también menor número de términos AND en cada ecuación. Para el caso de la MEF(1), las ecuaciones modificadas son las correspondientes a  $D_3$  y  $D_2$ . Las ecuaciones que se utilizan en el diseño son:

$$D_3 = A_4 \bar{A}_2 \bar{A}_1 \bar{A}_0 + A_3 A_2 \bar{A}_1 \bar{A}_0$$

$$D_2 = A_3 \bar{A}_1 \bar{A}_0 + A_4 \bar{A}_2 \bar{A}_1 \bar{A}_0$$

$$D_1 = \bar{A}_4 A_3 \bar{A}_1 \bar{A}_0 + \bar{A}_3 A_1 \bar{A}_0 + A_3 A_2 \bar{A}_1 \bar{A}_0$$

$$D_0 = \bar{A}_4 A_1 + \bar{A}_4 A_3 A_0 + A_3 \bar{A}_2 A_1 \bar{A}_0 + A_4 \bar{A}_3 \bar{A}_2 \bar{A}_1 \bar{A}_0$$

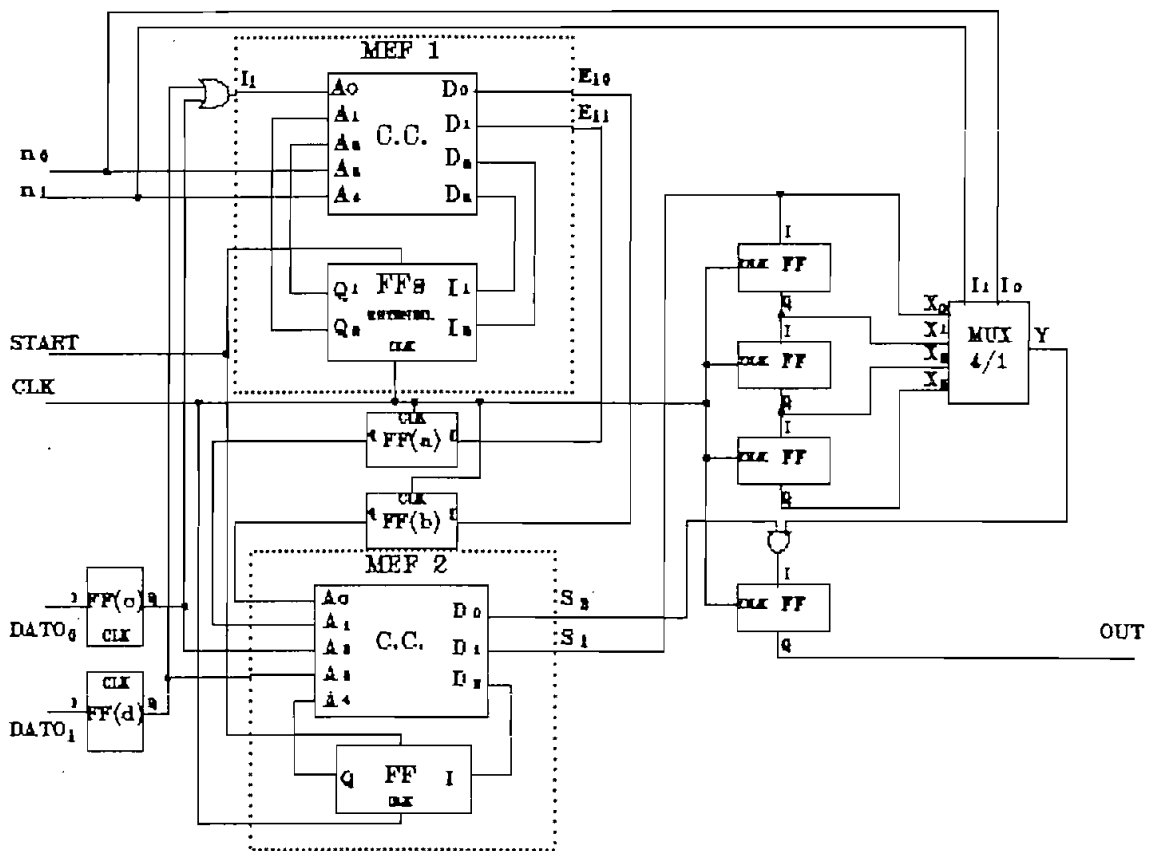


FIGURA 11.37 Esquema general del decodificador HDBn.

El resultado de integrar todas las celdas utilizadas para cada una de las funciones ( $D_0$  a  $D_3$ ) se presenta en la Fig. 11.38, se resuelve cada término AND de cada ecuación, para luego unificar aquellas filas en las que sea posible. Se incluye unicamente una fila de celdas "I".

	$A_4$	$A_3$	$A_2$	$A_1$	$A_0$	$D_3$	$D_2$	$D_1$	$D_0$		
	I	I	I	I	I						R11
$\bar{A}_4 A_1$	0			1					+		R10
$\bar{A}_4 A_3 A_0$	0	1			1				+		R9
$A_3 \bar{A}_2 A_1 \bar{A}_0$		1	0	1	0				+		R8
$A_4 \bar{A}_3 \bar{A}_2 \bar{A}_1 \bar{A}_0$	1	0	0	0	0				+		R7
$\bar{A}_4 A_3 \bar{A}_1 \bar{A}_0$	0	1		0	0			+			R6
$\bar{A}_3 A_1 \bar{A}_0$		0		1	0			+			R5
$A_3 A_2 \bar{A}_1 \bar{A}_0$		1	1	0	0			+			R4
$A_3 \bar{A}_1 \bar{A}_0$		1		0	0		+				R3
$A_4 \bar{A}_2 \bar{A}_1 \bar{A}_0$	1		0	0	0		+				R2
$A_4 \bar{A}_2 A_1 \bar{A}_0$	1		0	1	0	+					R1
$A_3 A_2 \bar{A}_1 \bar{A}_0$		1	1	0	0	+					R0
	C0	C1	C2	C3	C4	C5	C6	C7	C8	C9	C10

FIGURA 11.38 Distribución de celdas para las ecuaciones de la MEF(1).

Se pueden unificar las filas (R4 y R0), y luego proceder con la generación del archivo (.ppl) y continuar el proceso hasta simular este archivo. Los archivos (.src) y (.cpy) son:

ARCHIVO (.SRC)

;SIMULACION DEL CIRCUITO COMBINACIONAL DE LA MEF(1) DEL DECODIFICADOR  
COPY

VECTOR IN A14 A13 A12 A11 A10

VECTOR OUT D13 D12 D11 D10

OPTIONS TABLES -COPYINPUTS

WATCH IN OUT

```
;*****  
;# AMI #  
;*****  
SET IN=00000  
CYCLE  
SET IN=00001  
CYCLE
```

```
;*****  
;# HDB1 #  
;*****  
SET IN=01000  
CYCLE  
SET IN=01001  
CYCLE  
SET IN=01010  
CYCLE  
SET IN=01011  
CYCLE
```

```
;*****  
;# HDB2 #  
;*****  
SET IN=10000  
CYCLE  
SET IN=10001  
CYCLE  
SET IN=10010  
CYCLE  
SET IN=10011  
CYCLE  
SET IN=10100  
CYCLE  
SET IN=10101  
CYCLE
```

```
;*****  
;# HDB3 #  
;*****  
SET IN=11000  
CYCLE  
SET IN=11001  
CYCLE  
SET IN=11010  
CYCLE  
SET IN=11011  
CYCLE  
SET IN=11100  
CYCLE  
SET IN=11101  
CYCLE  
SET IN=11110  
CYCLE  
SET IN=11111  
CYCLE
```

ARCHIVO (.CPY)

```
>> VECTOR IN A14 A13 A12 A11 A10
>> VECTOR OUT D13 D12 D11 D10
>> OPTIONS TABLES -COPYINPUTS
  1> 00000 0000
  2> 00001 0000

  3> 01000 0110
  4> 01001 0001
  5> 01010 0001
  6> 01011 0001

  7> 10000 0101
  8> 10001 0000
  9> 10010 1010
 10> 10011 0000
 11> 10100 0000
 12> 10101 0000

 13> 11000 0100
 14> 11001 0000
 15> 11010 1001
 16> 11011 0000
 17> 11100 1110
 18> 11101 0000
 19> 11110 0000
 20> 11111 0000
```

Los valores de la tabla del archivo (.cpy) son idénticos a los planteados en la Tabla 9.28 y se concluye que el diseño del CC. está correcto.

Una vez realizado el diseño del CC., simplemente se deben incluir los *flip flops* necesarios, cambiando previamente el orden de algunas columnas. El intercambio de columnas se presenta en la Fig. 11.39 y la inclusión de los *flip flops*, señales de reloj y *start* se presenta en la Fig. 11.40.

La Fig. 11.40 se incluyen también los dos *flip flops* que se requieren entre la MEF(1) y MEF(2) (FFa y FFb de la Fig. 11.37). La señal  $D_1$  ingresa a un *flip flop* por C6 y sale por C7, para que no exista conflicto con la señal  $A_0$  y su complemento, que están presentes por C7, estas últimas se interrumpen en (R0:C7).

	A <sub>4</sub>	A <sub>3</sub>	A <sub>2</sub>	D <sub>3</sub>	A <sub>1</sub>	D <sub>2</sub>	D <sub>1</sub>	A <sub>0</sub>	D <sub>0</sub>		
I	I	I		I				I			R10
0				1					+		R9
0	1							1	+		R8
	1	0		1				0	+		R7
1	0	0		0				0	+		R6
0	1			0		+		0			R5
	0			1		+		0			R4
	1			0	+			0			R3
1		0		0	+			0			R2
1		0	+	1				0			R1
	1	1	+	0		+		0			R0
	C0	C1	C2	C3	C4	C5	C6	C7	C8	C9	C10

FIGURA 11.39 Intercambio de columnas del CC. de la MEF(1).

A <sub>4</sub>	A <sub>3</sub>	A <sub>2</sub>	D <sub>3</sub>	A <sub>1</sub>	D <sub>2</sub>	D <sub>1</sub>	A <sub>0</sub>	D <sub>0</sub>	START	
									I	R10
0	I			1			I	+	0	R9
0	1						1	+	0	R8
I	1	0		1			0	+	0	R7
1	0	0		0			0	+	0	R6
0	1			0		+	0		0	R5
	0			1		+	0		0	R4
	1			0	+		0		0	R3
1		0		0	+		0		0	R2
1		0	+	1			0		0	R1
	1	1	+	0		+	0		0	R0
		$\bar{Q}$ Q D		$\bar{Q}$ Q D		D $\bar{Q}$ Q		D $\bar{Q}$ Q	1	R2
										R1
		D3 M=4		D2 M=4		D1 M=5		D0 M=5		R0
4										
C0	C1	C2	C3	C4	C5	C6	C7	C8	C9	C10

FIGURA 11.40 NEF(1) con los flip flops, celda "4" y señal de start.

Se debe proceder ahora a describir el diseño con TILER, y ejecutar el programa SIMPPLEX, los datos que proporcionados por SIMPPLEX se presentan en la Fig. 11.41.

```
SIMPPLEX Version 4.8. Copyright 1986-91. Bonneville Microelectronics Inc.  
Loading SCMOS207.SDB database...  
Loading circuit from file mef(1).ppl...  
Finding circuit context...  
Tracing circuits connections...  
Creating extract file mef(1).ppx...
```

```
FATAL ERRORS: 0 (simulation allowed)  
ERRORS:      5 (5 ignored)  
WARNINGS:    0
```

```
Device statistics  
PPL rows:    15  
PPL columns: 12  
Transistors: 222  
Internal nodes: 49  
Input pointers: 6/119  
Output pointers: 10/119  
Total pads: 0
```

FIGURA 11.41 Pantalla de SIMPPLEX al ejecutarlo sobre MEF(1).ppl.

En la Fig. 11.42 se presentan el archivo (.prt) generado para la MEF(1). En la Fig. 11.43 se presenta el archivo (.prt) generado para el modo *ZOOM-IN*.

La lista de nodos entregada por cualquiera de los archivos (.prt) es:

```
A10 [13,7,0]  
A11 [13,4,0]  
A12 [13,2,0]  
A13 [13,1,0]  
A14 [13,0,0]  
CLK [0,0,4]  
D10 [13,8,0]  
D11 [13,6,0]  
D12 [13,5,0]  
D13 [13,3,0]  
START [14,10,0]
```



Circuit: mef(1) \ (SCMOS20T\ ) Date: Mon Sep 07 13:09:19 1992

```

          0 1 2 3 4 5 6 7 8 9 0 1
14:      _____ I |
13: |0 I      1      I + 0 |
12: |0 1      1      1 + 0 |
11: |I 1 0      1      0 + 0 |
10: |1 0 0      0      0 + 0 |
 9: |0 1      0      + 0      0 |
 8: | 0      1      + 0      0 |
 7: | 1      0 + 0      0      0 |
 6: |1 0      0 + 0      0      0 |
 5: |1 0 + 1      0      0      0 |
 4: | 1 1 + 0      + 0      0      0 |
 3: |_____ " " " " " " " " 1 |
 2: |" " " " " " " " " " " " |
 1: |" " D " D " D " D " " |
 0: | 4 " |

```

FIGURA 11.42 Impresión del archivo (.prt) de la MEF(1).

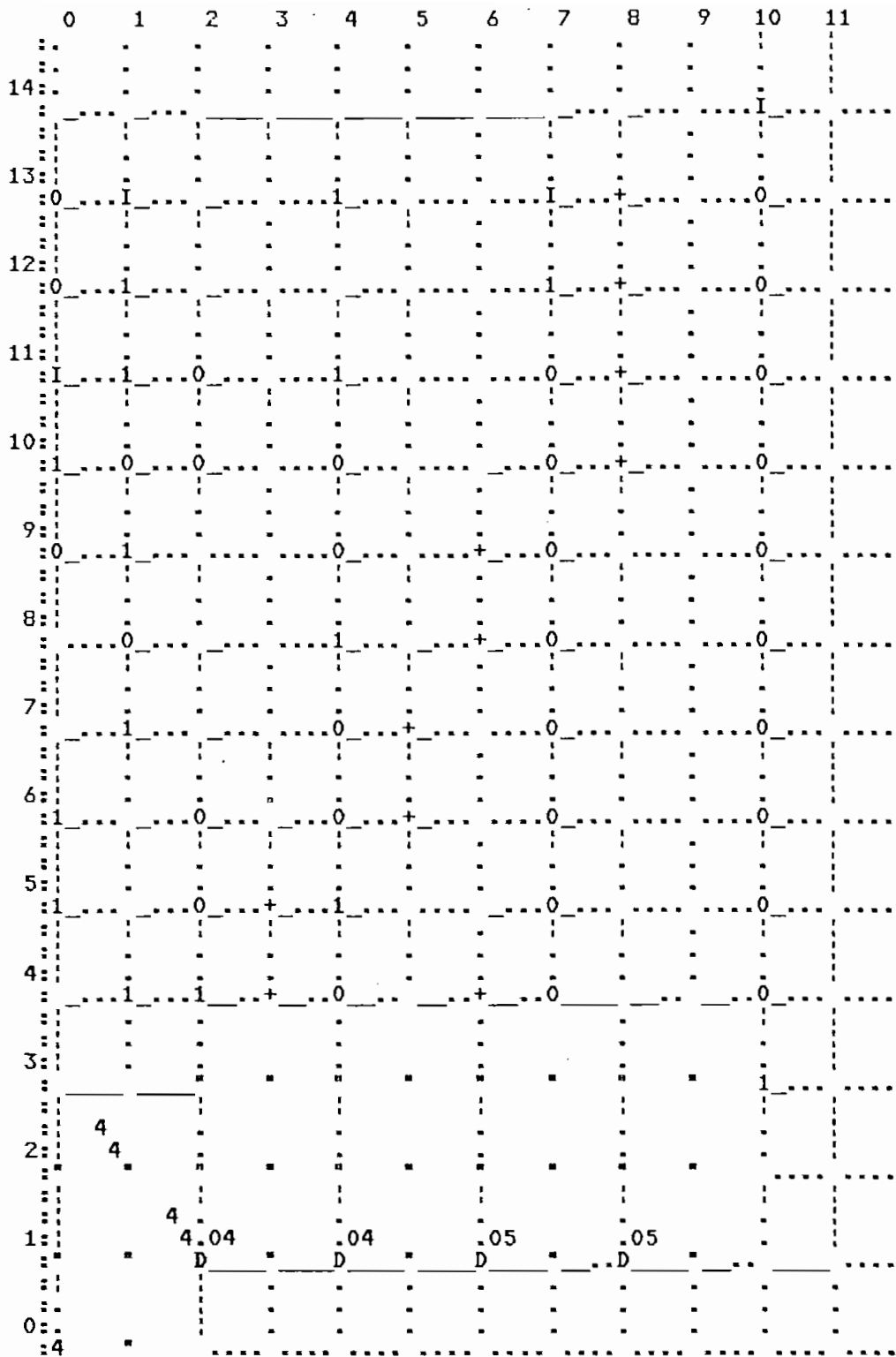


FIGURA 11.43 Impresión del archivo (.prt) para el modo ZOOM-IN.

### 11.2.2 MEF(2)

También en la MEF(2) del decodificador se ha modificado una de las ecuaciones, la correspondiente a  $D_2$ . Las ecuaciones planteadas para la MEF(2), con las que se realiza el diseño son:

$$D_2 = \bar{A}_4 A_3 + \bar{A}_4 A_2 \bar{A}_1 \bar{A}_0 + \bar{A}_4 \bar{A}_2 A_1 + A_4 \bar{A}_2 A_0 + A_3 A_1 + A_4 \bar{A}_3 \bar{A}_2 \bar{A}_1$$

$$D_1 = \bar{A}_4 A_3 + A_4 A_2 + \bar{A}_3 \bar{A}_2 A_1 + A_2 \bar{A}_1 \bar{A}_0 + A_3 \bar{A}_1 \bar{A}_0$$

$$D_0 = \bar{A}_0 + \bar{A}_4 \bar{A}_2 + A_4 \bar{A}_3$$

El resultado de integrar todas las celdas utilizadas para cada una de las funciones ( $D_0$  a  $D_2$ ) se presenta en la Fig. 11.44, se resuelve cada término de cada ecuación. Se incluye únicamente una fila de celdas "I".

	A <sub>4</sub>	A <sub>3</sub>	A <sub>2</sub>	A <sub>1</sub>	A <sub>0</sub>	D <sub>2</sub>	D <sub>1</sub>	D <sub>0</sub>			
	I	I	I	I	I						R14
$\bar{A}_0$					0			+			R13
$\bar{A}_4\bar{A}_2$	0		0					+			R12
A <sub>4</sub> $\bar{A}_3$	1	0						+			R11
$\bar{A}_4A_3$	0	1					+				R10
A <sub>4</sub> A <sub>2</sub>	1		1				+				R9
$\bar{A}_3\bar{A}_2A_1$			0	0	1		+				R8
A <sub>2</sub> $\bar{A}_1\bar{A}_0$				1	0	0	+				R7
A <sub>3</sub> $\bar{A}_1\bar{A}_0$			1		0	0	+				R6
$\bar{A}_4A_3$	0	1					+				R5
$\bar{A}_4A_2\bar{A}_1\bar{A}_0$	0			1	0	0	+				R4
$\bar{A}_4\bar{A}_2A_1$	0			0	1		+				R3
A <sub>4</sub> $\bar{A}_2A_0$	1			0		1	+				R2
A <sub>3</sub> A <sub>1</sub>			1		1		+				R1
A <sub>4</sub> $\bar{A}_3\bar{A}_2\bar{A}_1$	1	0	0	0			+				R0
	C0	C1	C2	C3	C4	C5	C6	C7	C8	C9	C10

FIGURA 11.44 Distribución de celdas para las ecuaciones de la MEF(2).

Una vez simplificadas las filas (R10 y R5), se utiliza TILER para generar el archivo (.pp1) y se sigue el proceso hasta simular este archivo. Los archivos (.src) y (.cpy) son:

ARCHIVO (.SRC)

```

;SIMULACION DEL CIRCUITO COMBINACIONAL DE LA MEF(2) DEL DECODIFICADOR
COPY
VECTOR IN A24 A23 A22 A21 A20
VECTOR OUT D22 D21 D20
OPTIONS TABLES -COPYINPUTS
WATCH IN OUT

```

```
;###ESTADO qi=0###
SET IN=00000
CYCLE
SET IN=00001
CYCLE
SET IN=00010
CYCLE
SET IN=00100
CYCLE
SET IN=00101
CYCLE
SET IN=00110
CYCLE
SET IN=01000
CYCLE
SET IN=01001
CYCLE
SET IN=01010
CYCLE
```

```
;###ESTADO qi=1###
SET IN=10000
CYCLE
SET IN=10001
CYCLE
SET IN=10010
CYCLE
SET IN=10100
CYCLE
SET IN=10101
CYCLE
SET IN=10110
CYCLE
SET IN=11000
CYCLE
SET IN=11001
CYCLE
SET IN=11010
CYCLE
```

ARCHIVO (.CPY)

```
>> VECTOR IN A24 A23 A22 A21 A20
>> VECTOR OUT D22 D21 D20
>> OPTIONS TABLES -COPYINPUTS
```

- 1> 00000 001
- 2> 00001 001
- 3> 00010 111
- 4> 00100 111
- 5> 00101 000
- 6> 00110 001
  
- 7> 01000 111
- 8> 01001 111
- 9> 01010 111
  
- 10> 10000 101
- 11> 10001 101
- 12> 10010 011
- 13> 10100 011
- 14> 10101 011
- 15> 10110 011
  
- 16> 11000 011
- 17> 11001 100
- 18> 11010 101

Los valores de la tabla del archivo (.cpy) son idénticos a los planteados en la Tabla 9.41 y se concluye que el diseño del CC. está correcto.

Una vez realizado el diseño del CC., para concluir el diseño de la MEF(2) simplemente se deben incluir los *flip flops* necesarios, cambiando previamente el orden de algunas columnas. como se indica en la Fig.11.45

	A <sub>3</sub>	A <sub>2</sub>	A <sub>1</sub>	A <sub>0</sub>	A <sub>4</sub>	D <sub>2</sub>	D <sub>1</sub>	D <sub>0</sub>			
	I	I	I	I	I					R13	
				0				+		R12	
		0			0			+		R11	
	0				1			+		R10	
	1				0	+	+			R9	
		1			1		+			R8	
	0	0	1				+			R7	
		1	0	0			+			R6	
	1		0	0			+			R5	
		1	0	0	0	+				R4	
		0	1		0	+				R3	
		0		1	1	+				R2	
	1		1			+				R1	
	0	0	0		1	+				R0	
	C0	C1	C2	C3	C4	C5	C6	C7	C8	C9	C10

FIGURA 11.45 Intercambio de columnas del CC. de la MEF(2) simplificada.

Al arreglo de la Fig. 11.45, se le añaden el único *flip flop* de realimentación, la celda "4" y la señal de *start*, como se indica en la Fig. 11.46.

	START	A <sub>3</sub>	A <sub>2</sub>	A <sub>1</sub>	A <sub>0</sub>	A <sub>4</sub>	D <sub>2</sub>	D <sub>1</sub>	D <sub>0</sub>		
I										R17	
0	I	I	I	0					+	R16	
0		0		I	0				+	R15	
0	0				1				+	R14	
0	1				0	+	+			R13	
0		1			1		+			R12	
0	0	0	1				+			R11	
0		1	0	0			+			R10	
0	1		0	0			+			R9	
0		1	0	0	0	+				R8	
0		0	1		0	+				R7	
0		0		1	1	+				R6	
0	1		1			+				R5	
0	0	0	0		1	+				R4	
1	—	—								R3	
	4									R2	
											R1
											R0
	C0	C1	C2	C3	C4	C5	C6	C7	C8	C9	C10

FIGURA 11.46 MEF(2) con el *flip flop* de realimentación, celda "4" y señal de *start*.

Se debe proceder a describir el diseño con TILER, para luego correr SIMPPLEX, cuyos resultados se presentan en la Fig. 11.47.

```
SIMPPLEX Version 4.8. Copyright 1986-91. Bonneville Microelectronics Inc.  
Loading SCMOS20T.SDB database...  
Loading circuit from file mef(2).ppl...  
Finding circuit context...  
Tracing circuits connections...  
Creating extract file mef(2).ppx...
```

```
FATAL ERRORS: 0 (simulation allowed)  
ERRORS:      6 (6 ignored)  
WARNINGS:    0
```

```
Device statistics  
PPL rows:    17  
PPL columns: 10  
Transistors: 170  
Internal nodes: 41  
Input pointers: 6/87  
Output pointers: 12/87  
Total pads: 0
```

FIGURA 11.47 Pantalla de SIMPPLEX al ejecutarlo sobre MEF(2).ppl.

En la Fig. 11.48 se presenta el archivo (.prt) generado para la MEF(2).

La lista de nodos entregada es:

```
A20 [16,4,0]  
A21 [16,3,0]  
A22 [16,2,0]  
A23 [16,1,0]  
A24 [16,5,0]  
CLK [0,0,4]  
D20 [16,8,0]  
D21 [16,7,0]  
D22 [16,6,0]  
START [16,0,0]
```

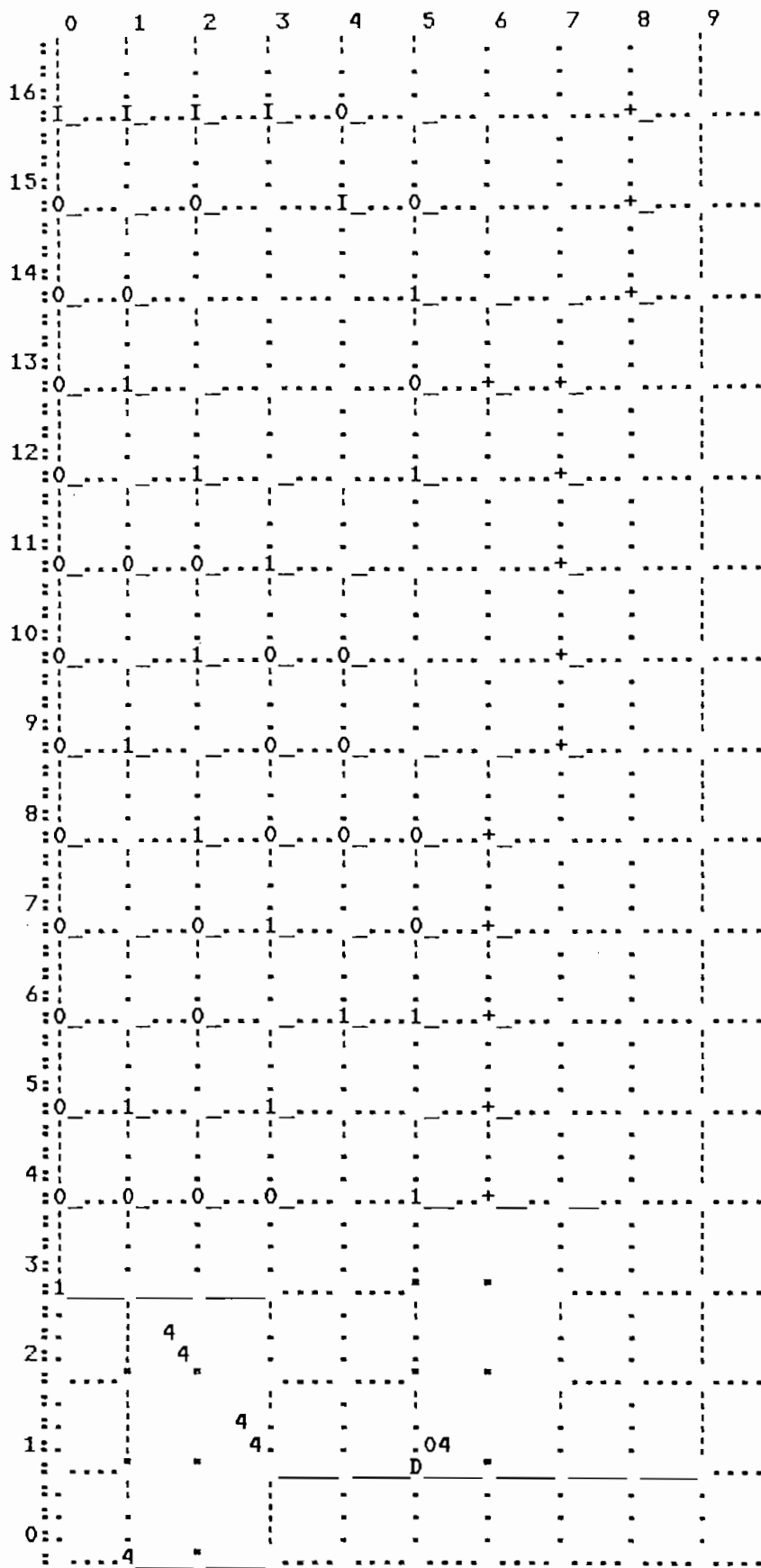


Circuit: MEF(2) \ (SCMDS20T\ ) Date: Mon Sep 07 15:28:49 1992

	0	1	2	3	4	5	6	7	8	9
16:	I	I	I	I	0					+
15:	0		0		I	0				+
14:	0	0				1				+
13:	0	1				0	+	+		
12:	0		1			1				+
11:	0	0	0	1						+
10:	0		1	0	0					+
9:	0	1		0	0					+
8:	0		1	0	0	0	0			+
7:	0		0	1		0				+
6:	0		0		1	1				+
5:	0	1		1						+
4:	0	0	0	0		1				+
3:	1									" "
2:	"	"								" "
1:	"	"				D				" "
0:	4									" "

FIGURA 11.48 Impresión del archivo (.prt)

En la siguiente página se presenta la Fig. 11.49, impresión del archivo (.prt) de la MEF(2) para el modo *ZOOM-IN*. Por falta de espacio, no se incluye el nombre de la figura al final de la misma.



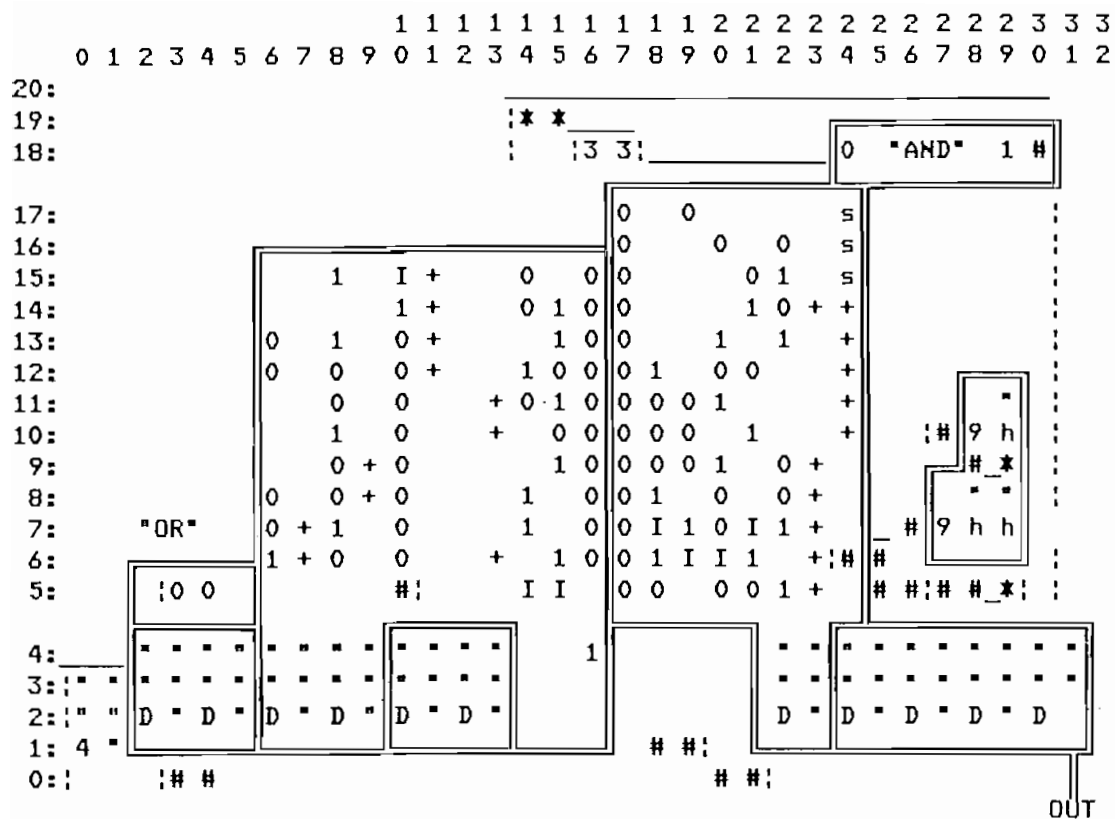
### 11.2.3 Integración de los módulos del decodificador

Una vez diseñadas y simuladas las dos MEFs del decodificador y con la estructura del multiplexor con celdas "h", el siguiente paso es integrarlos en un solo diseño. En la Fig. 11.50 se presenta el archivo (.prt) que se obtiene luego de haber integrado todos los módulos. Esta figura se ha dividido en bloques para indicar cada uno de los módulos constituyentes. Se presenta también la lista de nodos.

Para las señales de datos a decodificar, antes de los *flip flops* de entrada, se ha incluido los nombres: DAT00 y DAT01, para la salida de datos decodificados se utiliza el nombre OUT.

De los dos bloques más grandes, el de la izquierda corresponde a la MEF(1) y el de la derecha a la MEF(2). Del grupo de seis *flip flops* de la izquierda, dos son para las señales de entrada (FFc, FFd de la Fig. 11.37), dos de la MEF(1) y dos son los *flip flop* que se colocan a la salida de la MEF(1) (FFa y FFb de la Fig. 11.37), de izquierda a derecha. La salidas de FFa y FFb son entradas para la MEF(2).

Puede apreciarse claramente el multiplexor que utiliza celdas "h". Este multiplexor se combina con los cuatro *flip flops* y el bloque "AND" próximos a el, para entregar la salida "OUT".



Nombres de los nodos:

- |                 |               |
|-----------------|---------------|
| A10 [15,10,0]   | A20 [17,19,0] |
| A11 [15,6,0]    | A21 [17,18,0] |
| A12 [15,8,0]    | A22 [17,20,0] |
| A13 [15,15,0]   | A23 [17,21,0] |
| A14 [15,14,0]   | A24 [17,22,0] |
|                 |               |
| D10 [15,11,0]   |               |
| D11 [15,13,0]   |               |
| D12 [15,7,0]    |               |
| D13 [15,9,0]    |               |
|                 |               |
| D20 [18,24,1]   |               |
| D21 [18,24,0]   |               |
| D22 [17,23,0]   |               |
|                 |               |
| START [17,16,0] | CLK [1,0,4]   |
| NO [5,15,0]     | N1 [5,14,0]   |
| DAT00 [2,5,0]   | DAT01 [2,2,0] |
| OUT [2,31,1]    |               |

FIGURA 11.50 Distribución de los bloques que conforman el Decodificador.

Como para el caso del codificador, la distribución de celdas de las dos MEFs han sido alteradas pues se ha intercambiado algunas filas y columnas, respecto a las planteadas originalmente.

Así mismo, se utiliza una sola celda "4" (M=1) para proporcionar las señales de reloj a todos los *flip flops* del circuito.

Para realizar la función "OR" con las entradas de datos se utiliza el principio de la compuerta NAND con las entradas invertidas, por lo que se toman las señales negadas de los *flip flops* con celdas "0".

La compuerta AND de la etapa de salida se estructura con una celda "0" para tomar la salida  $D_{20}$ , que se obtiene con una celda "s" (salida en LCOL), y una celda "1" para tomar la salida del multiplexor. Como las dos celdas están en la misma fila se estructura una compuerta NAND. Para conseguir la función AND se debe negar esta señal; sin embargo, como la señal resultante de la función AND debe ingresar al último *flip flop* de la etapa de salida, se decide enrutar directamente la señal resultante de la compuerta NAND hacia el *flip flop* y tomar la señal negada del mismo.

En la Fig. 11.51 se presenta el archivo (.prt) para el decodificador, incluidas las interrupciones y sin la división en bloques.

En el Anexo H se presenta el archivo (.prt) para el modo ZOOM-IN y un diagrama generado con CAPFAST (incluidos los PADs) para el decodificador, en el que se aprecian claramente el enrutamiento de las señales.

Antes de realizar la simulación del decodificador se presentan los resultados obtenidos al ejecutar SIMPFLEX sobre el archivo *codi.ppl*. La pantalla presentada por SIMPFLEX para este archivo se presenta en la Fig. 11.52.

Circuit: DECO \((SCMOS20T\) Date: Tue Sep 08 04:06:45 1992

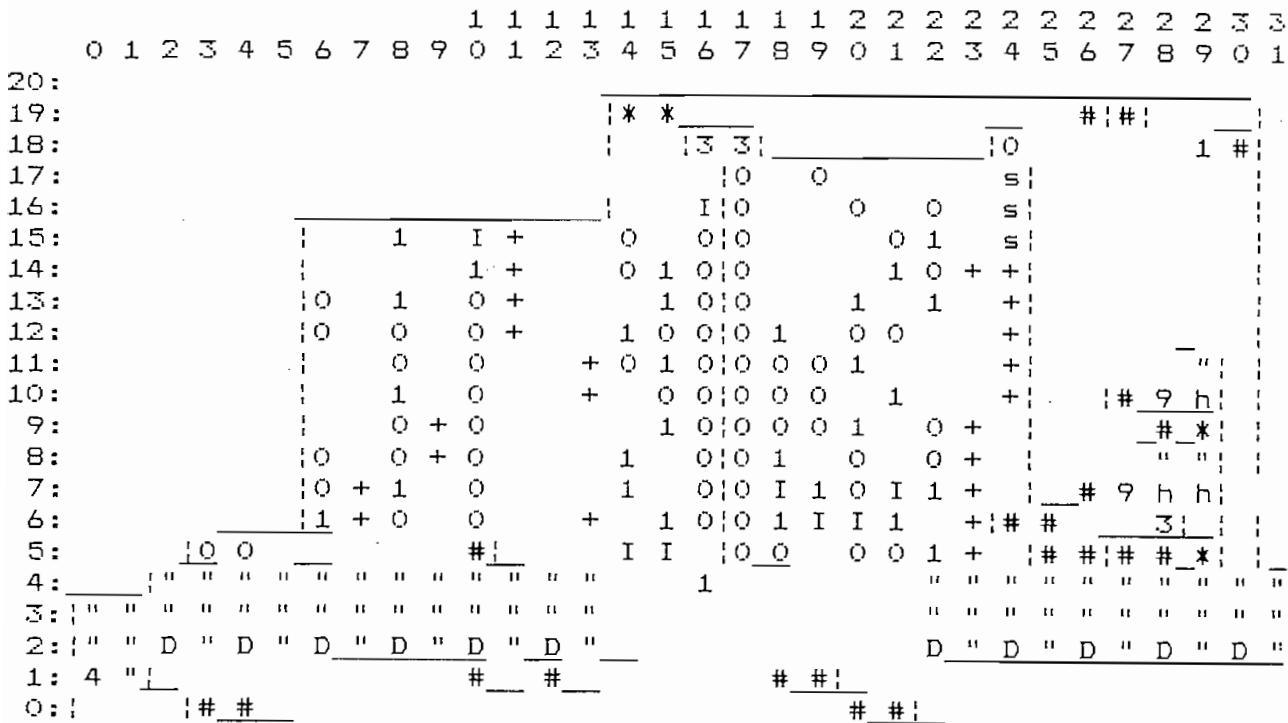


FIGURA 11.51 Distribución de celdas e interrupciones del Decodificador.

```
SIMPPLEX Version 4.8. Copyright 1986-91. Bonneville Microelectronics Inc.  
Loading SCMOS20T.SDB database...  
Loading circuit from file deco.ppl...  
Finding circuit context...  
Tracing circuits connections...  
Creating extract file deco.ppx...
```

```
FATAL ERRORS: 0 (simulation allowed)  
ERRORS: 6 (6 ignored)  
WARNINGS: 0
```

```
Device statistics  
PPL rows: 21  
PPL columns: 33  
Transistors: 516  
Internal nodes: 105  
Input pointers: 8/298  
Output pointers: 23/286  
Total pads: 0
```

FIGURA 11.52 Pantalla obtenida al ejecutar SIMPPLEX sobre deco.ppl.

Para simular el circuito total del decodificador, se considera que la forma más ilustrativa de comprobar que el decodificador está diseñado correctamente, es insertar en un solo módulo tanto el codificador como el decodificador e introducir secuencias de datos a codificar, que el decodificador debe entregar decodificadas. Para esto, se deben enrutar hacia el decodificador las señales de salida del codificador; además, se utilizan las mismas señales de selección, *start* y reloj. El archivo (.prt) obtenido para el circuito en que se interconectan codificador y decodificador, se presenta en la Fig. 11.53.

Se presentan a continuación el archivo de comandos para la simulación y los resultados obtenidos.

ARCHIVO .SRC

COPY  
VECTOR SELECT N1 NO  
CLOCK CLK:10  
WATCH START SELECT DATO OUT  
OPTIONS -COPYINPUTS

SET START=1 SELECT=00 DATO=1  
CYCLE 2

;\*\*\*\*\* AMI \*\*\*\*\*  
SET START=0 SELECT=00  
SET DATO=1  
CYCLE 3  
SET DATO=0  
CYCLE 5  
SET DATO=1  
CYCLE 2  
SET DATO=0  
CYCLE 4  
SET DATO=1

;\*\*\*\*\* HDB1 \*\*\*\*\*  
SET START=1  
CYCLE  
SET START=0 SELECT=01  
CYCLE  
SET DATO=1  
CYCLE 2  
SET DATO=0  
CYCLE 3  
SET DATO=1  
CYCLE 4  
SET DATO=0  
CYCLE 5  
SET DATO=1  
CYCLE 6  
SET DATO=0  
CYCLE 7  
SET DATO=1

;\*\*\*\*\* HDB2 \*\*\*\*\*  
SET START=1  
CYCLE  
SET START=0 SELECT=10  
CYCLE  
SET DATO=1  
CYCLE 3  
SET DATO=0  
CYCLE 4  
SET DATO=1  
CYCLE 5  
SET DATO=0  
CYCLE 6



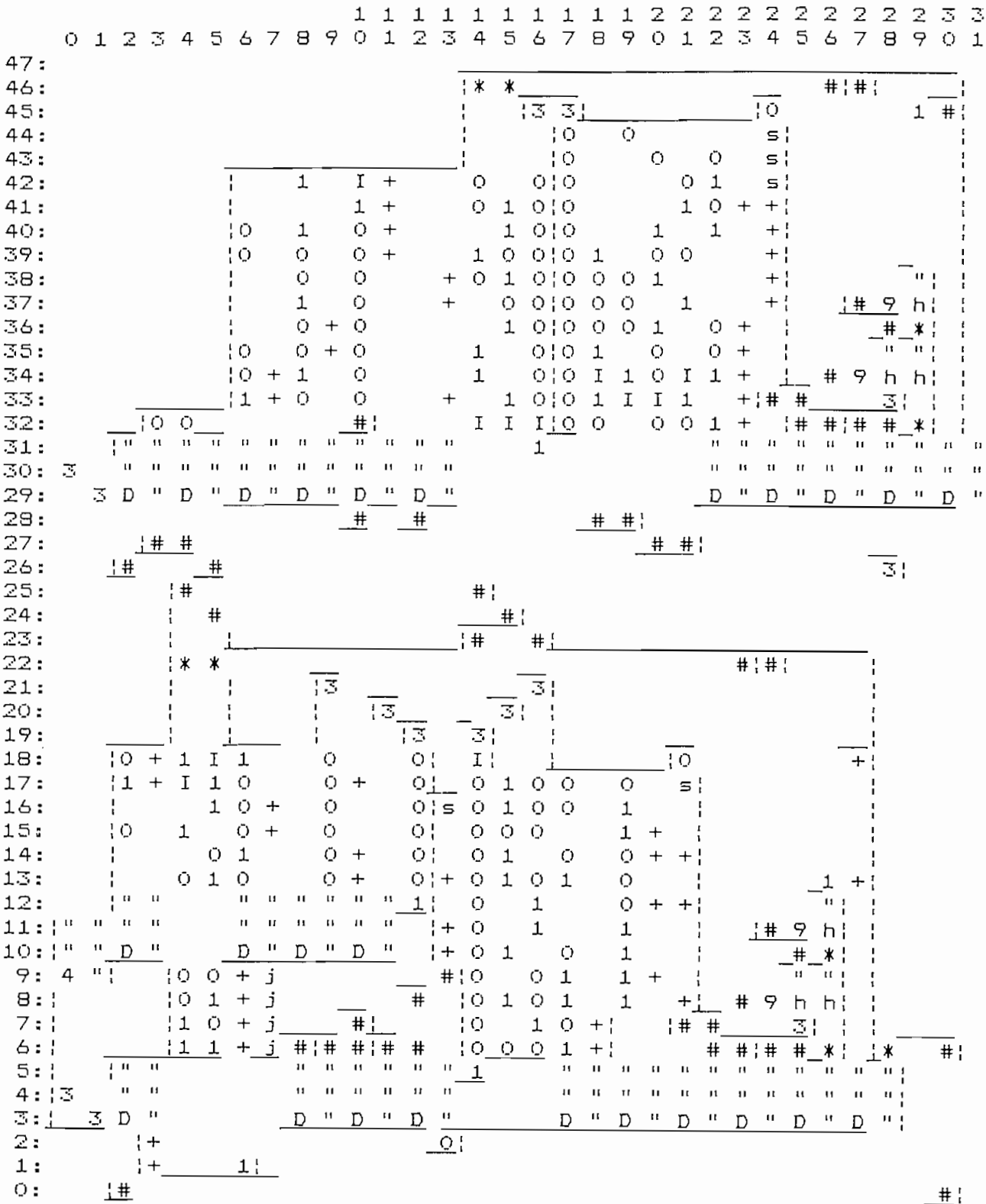


FIGURA 11.53 Archivo (.prt) del circuito en que se interconecta codificador con decodificador.

```

SET      DATO=1
CYCLE 7
SET      DATO=0
CYCLE 8
SET      DATO=1

;***** HDB3 *****
SET START=1
CYCLE
SET START=0 SELECT=11
CYCLE
SET      DATO=1
CYCLE 3
SET      DATO=0
CYCLE 4
SET      DATO=1
CYCLE 2
SET      DATO=0
CYCLE 1
SET      DATO=1
CYCLE 5
SET      DATO=0
CYCLE 7
SET      DATO=1
CYCLE 2
SET      DATO=0
CYCLE 10

```

```

ARCHIVO .CPY
>> VECTOR SELECT N1 NO
>> CLOCK CLK:10
>> WATCH START SELECT DATO OUT
>> OPTIONS -COPYINPUTS

```

1:2>	START=1	SELECT=XX	DATO=X	OUT=1
2:2>	START=1	SELECT=00	DATO=1	OUT=1
3:2>	START=1	SELECT=00	DATO=1	OUT=1
4:2>	START=0	SELECT=00	DATO=1	OUT=0
5:2>	START=0	SELECT=00	DATO=1	OUT=0
6:2>	START=0	SELECT=00	DATO=1	OUT=0
7:2>	START=0	SELECT=00	DATO=0	OUT=1
8:2>	START=0	SELECT=00	DATO=0	OUT=1
9:2>	START=0	SELECT=00	DATO=0	OUT=1
10:2>	START=0	SELECT=00	DATO=0	OUT=0
11:2>	START=0	SELECT=00	DATO=0	OUT=0
12:2>	START=0	SELECT=00	DATO=1	OUT=0
13:2>	START=0	SELECT=00	DATO=1	OUT=0
14:2>	START=0	SELECT=00	DATO=0	OUT=0
15:2>	START=0	SELECT=00	DATO=0	OUT=1
16:2>	START=0	SELECT=00	DATO=0	OUT=1
17:2>	START=0	SELECT=00	DATO=0	OUT=0
18:2>	START=1	SELECT=00	DATO=0	OUT=1

19:2>	START=0	SELECT=01	DATO=0	OUT=0
20:2>	START=0	SELECT=01	DATO=1	OUT=0
21:2>	START=0	SELECT=01	DATO=1	OUT=1
22:2>	START=0	SELECT=01	DATO=0	OUT=0
23:2>	START=0	SELECT=01	DATO=0	OUT=0
24:2>	START=0	SELECT=01	DATO=0	OUT=0
25:2>	START=0	SELECT=01	DATO=1	OUT=1
26:2>	START=0	SELECT=01	DATO=1	OUT=1
27:2>	START=0	SELECT=01	DATO=1	OUT=0
28:2>	START=0	SELECT=01	DATO=1	OUT=0
29:2>	START=0	SELECT=01	DATO=0	OUT=0
30:2>	START=0	SELECT=01	DATO=0	OUT=1
31:2>	START=0	SELECT=01	DATO=0	OUT=1
32:2>	START=0	SELECT=01	DATO=0	OUT=1
33:2>	START=0	SELECT=01	DATO=0	OUT=1
34:2>	START=0	SELECT=01	DATO=1	OUT=0
35:2>	START=0	SELECT=01	DATO=1	OUT=0
36:2>	START=0	SELECT=01	DATO=1	OUT=0
37:2>	START=0	SELECT=01	DATO=1	OUT=0
38:2>	START=0	SELECT=01	DATO=1	OUT=0
39:2>	START=0	SELECT=01	DATO=1	OUT=1
40:2>	START=0	SELECT=01	DATO=0	OUT=1
41:2>	START=0	SELECT=01	DATO=0	OUT=1
42:2>	START=0	SELECT=01	DATO=0	OUT=1
43:2>	START=0	SELECT=01	DATO=0	OUT=1
44:2>	START=0	SELECT=01	DATO=0	OUT=1
45:2>	START=0	SELECT=01	DATO=0	OUT=0
46:2>	START=0	SELECT=01	DATO=0	OUT=0
47:2>	START=1	SELECT=01	DATO=0	OUT=1
48:2>	START=0	SELECT=10	DATO=0	OUT=0
49:2>	START=0	SELECT=10	DATO=1	OUT=0
50:2>	START=0	SELECT=10	DATO=1	OUT=0
51:2>	START=0	SELECT=10	DATO=1	OUT=0
52:2>	START=0	SELECT=10	DATO=0	OUT=1
53:2>	START=0	SELECT=10	DATO=0	OUT=0
54:2>	START=0	SELECT=10	DATO=0	OUT=0
55:2>	START=0	SELECT=10	DATO=0	OUT=0
56:2>	START=0	SELECT=10	DATO=1	OUT=1
57:2>	START=0	SELECT=10	DATO=1	OUT=1
58:2>	START=0	SELECT=10	DATO=1	OUT=1
59:2>	START=0	SELECT=10	DATO=1	OUT=0
60:2>	START=0	SELECT=10	DATO=1	OUT=0
61:2>	START=0	SELECT=10	DATO=0	OUT=0
62:2>	START=0	SELECT=10	DATO=0	OUT=0
63:2>	START=0	SELECT=10	DATO=0	OUT=1
64:2>	START=0	SELECT=10	DATO=0	OUT=1
65:2>	START=0	SELECT=10	DATO=0	OUT=1
66:2>	START=0	SELECT=10	DATO=0	OUT=1
67:2>	START=0	SELECT=10	DATO=1	OUT=1
68:2>	START=0	SELECT=10	DATO=1	OUT=0
69:2>	START=0	SELECT=10	DATO=1	OUT=0
70:2>	START=0	SELECT=10	DATO=1	OUT=0
71:2>	START=0	SELECT=10	DATO=1	OUT=0

72:2>	START=0	SELECT=10	DATO=1	OUT=0
73:2>	START=0	SELECT=10	DATO=1	OUT=0
74:2>	START=0	SELECT=10	DATO=0	OUT=1
75:2>	START=0	SELECT=10	DATO=0	OUT=1
76:2>	START=0	SELECT=10	DATO=0	OUT=1
77:2>	START=0	SELECT=10	DATO=0	OUT=1
78:2>	START=0	SELECT=10	DATO=0	OUT=1
79:2>	START=0	SELECT=10	DATO=0	OUT=1
80:2>	START=0	SELECT=10	DATO=0	OUT=1
81:2>	START=0	SELECT=10	DATO=0	OUT=0
82:2>	START=1	SELECT=10	DATO=1	OUT=1
83:2>	START=0	SELECT=11	DATO=1	OUT=0
84:2>	START=0	SELECT=11	DATO=1	OUT=0
85:2>	START=0	SELECT=11	DATO=1	OUT=0
86:2>	START=0	SELECT=11	DATO=1	OUT=0
87:2>	START=0	SELECT=11	DATO=0	OUT=0
88:2>	START=0	SELECT=11	DATO=0	OUT=0
89:2>	START=0	SELECT=11	DATO=0	OUT=1
90:2>	START=0	SELECT=11	DATO=0	OUT=0
91:2>	START=0	SELECT=11	DATO=1	OUT=0
92:2>	START=0	SELECT=11	DATO=1	OUT=1
93:2>	START=0	SELECT=11	DATO=0	OUT=1
94:2>	START=0	SELECT=11	DATO=1	OUT=1
95:2>	START=0	SELECT=11	DATO=1	OUT=1
96:2>	START=0	SELECT=11	DATO=1	OUT=0
97:2>	START=0	SELECT=11	DATO=1	OUT=0
98:2>	START=0	SELECT=11	DATO=1	OUT=0
99:2>	START=0	SELECT=11	DATO=0	OUT=0
100:2>	START=0	SELECT=11	DATO=0	OUT=1
101:2>	START=0	SELECT=11	DATO=0	OUT=1
102:2>	START=0	SELECT=11	DATO=0	OUT=0
103:2>	START=0	SELECT=11	DATO=0	OUT=1
104:2>	START=0	SELECT=11	DATO=0	OUT=1
105:2>	START=0	SELECT=11	DATO=0	OUT=1
106:2>	START=0	SELECT=11	DATO=1	OUT=1
107:2>	START=0	SELECT=11	DATO=1	OUT=1
108:2>	START=0	SELECT=11	DATO=0	OUT=0
109:2>	START=0	SELECT=11	DATO=0	OUT=0
110:2>	START=0	SELECT=11	DATO=0	OUT=0
111:2>	START=0	SELECT=11	DATO=0	OUT=0
112:2>	START=0	SELECT=11	DATO=0	OUT=0
113:2>	START=0	SELECT=11	DATO=0	OUT=0
114:2>	START=0	SELECT=11	DATO=0	OUT=0
115:2>	START=0	SELECT=11	DATO=0	OUT=1
116:2>	START=0	SELECT=11	DATO=0	OUT=1
117:2>	START=0	SELECT=11	DATO=0	OUT=0

Los resultados de la simulación son satisfactorios y para analizarlos, se debe considerar los retardos debido a la presencia de los *flip flops* de la entrada y salida, tanto del

codificador como del decodificador. Para evaluar el área utilizada por el decodificador, se ejecuta PPLACE. Los resultados se muestran en la Fig. 11.54.

```
Registered to: University STUDENTS
(C) Copyright 1990 Bonneville Micro.
All Rights Reserved
PPLACE Version 1.0e, Copyright 1986-91 Bonneville Microelectronics Inc.
Reading file "deco.ppl"...
Loading cell set SCMOS207...
Performing DRC verification...
Removing unused cell connections...
Processing file "deco.ppl"...
      size: 1650 X 462 (microns)
[PPLACE] 1 CIF Error written to deco.err
```

FIGURA 11.54 Pantalla y resultados obtenidos al ejecutar PPLACE

El área utilizada por el decodificador, sin PADS, es 1650  $\mu\text{m}$  X 462  $\mu\text{m}$ . El error reportado en el archivo *deco.err* es el mismo que se encontró para el codificador; es decir, se debe a que el módulo no se ajusta a las dimensiones del *PAD-FRAME*.

### 11.3 INSERCIÓN DEL DISEÑO GLOBAL EN EL *PAD-FRAME*

Una vez diseñados y simulados tanto codificador como decodificador, se procede a insertarlos en el *PAD-FRAME tiny34*. Se decide colocar el codificador en la parte superior y al decodificador en la parte inferior del *PAD-FRAME*. La asignación de los PADS, que corresponden a los pines externos, se presenta en la Fig. 11.55.

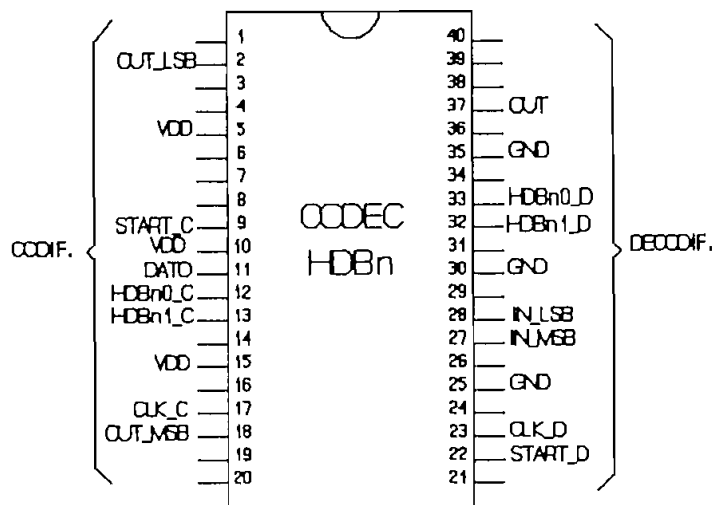


FIGURA 11.55 Distribución de pines para el codec.

Se utilizan en total 20 pines, 6 de polarización, 7 para el codificador y 7 para el decodificador. La nomenclatura escogida para nombrar los pines, es la que también se utilizó en el diseño con TENTOS.

El proceso de inserción se limita a enrutar las señales de entrada y salida del codificador y decodificador a los PADS elegidos. Una vez hecho esto se procede a simular independientemente cada uno de ellos. Los resultados obtenidos fueron similares a los obtenidos antes de incluir los PADS, y por lo tanto satisfactorios.

Dado que el codificador es totalmente independiente del decodificador, cada módulo debe incluir su celda "4", y por tanto la señal externa de reloj. Deben ser también independientes las señales de selección y *start*.

En el Anexo H, se presentan el archivo (.prt) del circuito total, al que se le ha denominado *codec.ppl*, puede notarse la presencia de una área no utilizada en la parte central del *PAD-FRAME*. Los esquemas del codificador y decodificador obtenidos con CAPFAST, a los que se ha hecho mención anteriormente, incluyen la interconexión de los módulos hacia los *PADs*, y en ellos puede observarse el enrutamiento de las señales de entrada y salida hacia los *PADs*.

Los resultados obtenidos al correr SIMPPLEX sobre el archivo *codec.ppl* se presentan en la Fig. 11.56.

```
SIMPPLEX Version 4.8. Copyright 1986-91. Bonneville Microelectronics Inc.  
Loading SCHOS20T.SDB database...  
Loading circuit from file codec.ppl...  
Finding circuit context...  
Tracing circuits connections...  
Creating extract file codec.ppx...  
FATAL ERRORS: 0  
ERRORS: 0  
WARNINGS: 0  
Device statistics  
PPL rows: 79  
PPL columns: 36  
Transistors: 1280  
Internal nodes: 235  
Input pointers: 8/642  
Output pointers: 23/618  
Signal pads: 34  
VDD pads: 3  
GND pads: 3  
Total pads: 40
```

FIGURA 11.56 Pantalla obtenida al ejecutar SIMPPLEX sobre *codec.ppl*.

El número total de transistores utilizados en el codificador y decodificador, incluidos los transistores de los FADs, es 1280.

Los resultados obtenidos con PPLACE se presentan en la Fig. 11.57.

```
Registered to: University STUDENTS
(C) Copyright 1990 Bonneville Micro.
All Rights Reserved
PPLACE Version 1.0e, Copyright 1986-91 Bonneville Microelectronics Inc.
Reading file "codec.ppl"...
Loading cell set SCMOS20T...
Performing DRC verification...
Removing unused cell connections...
Processing file "codec.ppl"...
      size: 2220 X 2250 (microns)
```

FIGURA 11.57 Pantalla y resultados obtenidos al ejecutar PPLACE

El área que se reporta es la correspondiente al *PAD-FRAME* (2220  $\mu\text{m}$  X 2250  $\mu\text{m}$ ).

Sin ningún mensaje de error, se puede proceder a la conversión a formato CIF. Se debe mencionar que también dentro de TILER se comprobó que el diseño no contenga errores CIF.

Con el módulo depurado se procede a la conversión del archivo *codec.ppl* a formato CIF. La pantalla desplegada al ejecutar el programa PPL2CIF se presenta en la Fig. 11.58.



This program will expire after 2 more Use(s).  
PPL conversion Disk, Limited To 4 Conversions  
(C) Copyright 1990 Bonneville Microelectronics Inc.

PPL2CIF Version 4.1a, Copyright 1986-91 University of Utah  
Reading file "C:\APPL\TESIS\CODEC"....  
Loading cell set SCMOS20T...  
Performing DRC verification...  
Removing unused cell connections...  
Processing file "C:\APPL\TESIS\CODEC.PPL"...  
    size: 2220 X 2250 (microns)  
    pads: 40  
Extracting cells from "SMOSfile.sif" library...  
Extracting cells from "TINY20pd.sif" library...  
Extracting cells from "CMOSbits.sif" library...

FIGURA 11.58 Pantalla y resultados obtenidos al ejecutar PPL2CIF

## CAPITULO 12

### CONCLUSIONES Y RECOMENDACIONES

#### 12.1 CONCLUSIONES GENERALES

1. Existen diversas opciones de diseño tanto en cuanto a la metodología (Full-Custom, Semi-Custom y Lógica Programable), como al proceso tecnológico de fabricación (Bipolar, Metal-Oxido-Semiconductor, Arseniuro de Galio y CMOS-Bipolar).
2. Los costos de fabricación de un prototipo son muy altos. El Proyecto Multiusuario (PMU) Iberoamericano ha permitido a la Escuela Politécnica Nacional la fabricación del primero de los diseños realizados, el "Medio Sumador" y existen perspectivas ciertas de lograr el financiamiento para la fabricación del segundo, el "Codificador/Decodificador HDBn".
3. La metodología empleada en el caso del presente trabajo ha sido seleccionada principalmente en función de las herramientas a que se tuvo acceso, del equipamiento del

que se dispuso, y de las restricciones impuestas tanto por el Centro Nacional de Microelectrónica (CNM) de Barcelona-España como por el fabricante, la empresa European Silicon Structures (ES2) de Francia.

4. Se recorrió el proceso completo para el diseño de un circuito integrado sencillo "Medio Sumador", desde su concepción a nivel funcional, hasta la caracterización de los prototipos fabricados.
5. En el diseño del circuito "Medio Sumador" se encontraron varias dificultades, propias de un trabajo en el que no se tiene experiencia previa, agravadas por el carácter experimental del paquete TENTOS, que aún no ha sido completamente depurado.
6. El circuito "Medio Sumador" obtenido presenta un error de inversión lógica en la respuesta de sus señales de salida debido a la característica inversora de los PADs (OPS1U) empleados, que no fue considerada en el momento del diseño.
7. El error cometido no tiene ninguna trascendencia en el objetivo primordial de este diseño, esto es de sentar las bases para el desarrollo de proyectos de mayor complejidad.

8. Ha sido posible concluir el diseño del circuito "Codificador/Decodificador HDBn", con el paquete TENTOS basado en la experiencia obtenida en el diseño del circuito "Medio Sumador", su fabricación será posible de lograrse el financiamiento correspondiente.
9. También se ha concluido el diseño del "Codificador/Decodificador HDBn" con el paquete PFL, el estudio de este paquete ha permitido dar un paso importante en la investigación de las metodologías de diseño no convencionales que constituyen una innovación a los conceptos tradicionales de niveles jerárquicos de diseño.
10. La realización del diseño del circuito "Codificador/Decodificador HDBn" en base a los mismos algoritmos de concepción funcional con las herramientas TENTOS y PFL ha permitido comparar de manera concreta dos filosofías distintas de diseño.
11. Las limitaciones del equipamiento de que se dispuso, tanto a nivel de hardware, como de software para el diseño de circuitos VLSI son crecientes a medida que aumenta la complejidad de los circuitos. El circuito "Codificador/Decodificador HDBn" con 1286 transistores debió ser sintetizado, construido, depurado, simulado, editado e incluso impreso (en planos) dividiéndolo en módulos.

12. Las futuras incursiones en diseño VLSI poseen en el presente trabajo una fuente de información teórico-práctico-metodológica sólida que suplirá la falta de material bibliográfico al respecto, en el medio.
  
13. Se abre la factibilidad a las instituciones públicas y privadas para hacer uso de las ventajas de la introducción de la tecnología VLSI en sus aplicaciones, logrando de esta manera consolidar industrias de mayor productividad, basadas en procesos de mayor confiabilidad, lo que conllevaría a una mayor competitividad, en momentos en que el país enfrenta el reto de integración al mercado andino.

## 12.2 CONCLUSIONES RESPECTO AL PAQUETE TENTOS

1. El sistema de herramientas que posee se halla en proceso de depuración, a ello se debe que han sido detectados varios errores que fueron reportados oportunamente al equipo de investigación que lo desarrolla, para su corrección.
  
2. La metodología convencional de diseño (TOP-DOWN), implica un trabajo en los tres niveles de diseño (funcional, lógico y físico), es necesaria una verificación rigurosa en la transición entre ellos debido a la dificultad

creciente en la corrección de los errores a medida que se desciende de nivel.

3. El diseño en el nivel físico en el paquete TENTOS presenta la ventaja de permitir el acceso a la edición e incluso creación de celdas de biblioteca. Esta ventaja implica un mayor nivel de conocimientos y lleva al diseño a la categoría de FULL-CUSTOM. Su aprovechamiento depende del dominio de la técnica de diseño VLSI que posea el usuario.
4. En la descripción de un diseño en el paquete TENTOS se pueden emplear los lenguajes NILO-TRANCA, SPICE y la descripción gráfica, siendo una de las tres descripciones suficiente para caracterizar un circuito, se estima que la descripción en lenguaje SPICE es la más versátil puesto que puede ser usada para la síntesis del circuito tanto mediante el método TRAMO como TRAGO, así como para su simulación.
5. El diseñador puede asignar restricciones de orientación y de posicionamiento a las celdas de su circuito; sin embargo la acumulación de estas restricciones limita la libertad de los algoritmos de síntesis del layout que además de emplear mayor tiempo de procesamiento, generan soluciones en que las restricciones han sido parcialmente atendidas y en casos extremos originan layouts incompletos.

6. De lo anterior se desprende la distribución de las celdas en bandas es, en última instancia, privativa de los programas de posicionamiento y no del diseñador.
7. Los algoritmos de enrutamiento del sintetizador TRAMO del TENTOS presentan dos limitantes:
  - i) La falta de transparencia vertical en las celdas de biblioteca y el no aprovechamiento de la capa de METAL2 obligan a una abertura de espacios para intercalar las celdas de interconexión vertical lo que deriva en un desperdicio de área de diseño.
  - ii) El enrutamiento horizontal a través de las filas libres de las bandas es ineficiente, puesto que de no haber filas libres para nuevas interconexiones estas últimas no se forman.
8. De un estudio detallado de los layouts generados por el sintetizador TRAMO, se concluye que los caminos de interconexión que este forma no son necesariamente los más cortos y pueden ser depurados.
9. El sintetizador de layout TRAGO propone una solución a las limitaciones del TRAMO puesto que el enrutamiento vertical se realiza en METAL2 y la altura ajustable de las bandas permite dar cabida a todas las redes de enrutamiento horizontal.

10. Las celdas de biblioteca del paquete TENTOS y por ende los layouts que estas generan presentan errores de:
  - i) Posicionamiento de Pozos N y zonas de Pasivación.
  - ii) Ausencia de Zonas Activas en la definición de transistores.
  
11. Los programas de enrutamiento del paquete TENTOS originan errores de:
  - i) Sobrelapamiento o "doble definición" de máscaras de METAL 1 debido al enrutamiento horizontal.
  - ii) Sobrelapamiento o "doble definición" de máscaras de POLYSILICON debido al enrutamiento vertical.
  
12. Los errores mencionados anteriormente han debido ser corregidos en forma "manual" mediante el editor de máscaras EMA2, ello implica trabajo de diseño al nivel físico y desvirtúa el carácter automático de generación de layouts del TENTOS.
  
13. El programa SPICE que posee el paquete TENTOS (versión de Abril de 1991) permite la simulación de circuitos con hasta alrededor de 130 transistores, debido a ello se restringió su empleo al caso de circuitos pequeños como el "Medio Sumador".
  
14. La limitación anterior debió ser solventada mediante el programa de simulación NDL perteneciente al paquete de diseño TEDMOS desarrollado en la Universidad Federal de



Rio de Janeiro "Brasil", las dificultades en el uso de este programa radican en la imposibilidad de generar las *netlists* respectivas en forma automática y en que el número máximo de transistores en los circuitos que simula fue evaluado en alrededor de 230.

15. Las dificultades indicadas anteriormente expresan la ausencia de una herramienta adecuada para la simulación de los diseños al nivel físico, lo cual es grave considerando la importancia radical que tiene el proceso de simulación y verificación circuital en el diseño VLSI.
16. Existen limitaciones en la capacidad de procesamiento y edición gráfica de circuitos grandes (miles de transistores) debido al ambiente DOS en que se desenvuelve el paquete TENTOS y fundamentalmente a la imposibilidad que tiene este de acceder a localidades superiores de memoria RAM del computador.
17. El paquete de herramientas TENTOS no cuenta con bibliotecas de PADS debido a ello se empleó la biblioteca PADLIB2 (ECPD15) provista por el fabricante ES2, la integración de estas celdas al layout debió ser realizada en forma manual mediante el editor de máscaras EMA2.
18. El diseño convencional utilizado por TENTOS, sumado a los errores propios del paquete, hicieron el diseño del

codificador sumamente rígido, al punto que cualquier modificación en el nivel funcional o lógico implicaba una reformulación total del diseño en el nivel físico.

### 12.3 CONCLUSIONES RESPECTO AL PAQUETE PPL

1. El sistema de herramientas de este paquete posee un alto grado de depuración; además, continuamente los investigadores que lo diseñaron están incluyendo mejoras y preparando los conjuntos de celdas para nuevas tecnologías.
2. Se encontraron problemas inesperados con la instalación de los paquetes PFL y CAPFAST, debido a las protecciones de software y hardware que incluyen estos paquetes. Las soluciones adoptadas quedan adecuadamente documentadas.
3. Para los diseños realizados, las herramientas del paquete PPL no presentaron limitaciones en lo referente al número máximo de transistores que pueden describirse con TILER o que pueden simularse con SIMPPL. Se logró incluso simular el codificador interconectado con el decodificador HDBn, con lo que se llegó a 1066 transistores.
4. El paquete PPL utiliza una metodología no convencional, en la que el diseñador no requiere descender al nivel físico; PPL integra diferentes niveles de las fases de

diseño, evitando iteraciones innecesarias entre ellas y operando completamente a un nivel simbólico.

5. Para realizar el diseño con PPL no se requieren diseñadores altamente especializados, un individuo con conocimientos de sistemas digitales y un conocimiento básico de los conceptos de diseño VLSI, puede emprender la tarea de aprendizaje de la metodología y herramientas PPL.
6. La metodología PPL, sumada al alto grado de depuración del sistema de herramientas disponibles, proporcionan una mejor productividad del diseñador.
7. La planificación de la distribución de las celdas PPL puede realizarse fácilmente, aún sin disponer de un computador.
8. En PPL, al realizar la planificación de la distribución de las celdas de un circuito, se determina simultáneamente la ubicación física de las mismas en el CI fabricado.
9. A diferencia de los métodos de diseño convencionales, en los que el enrutamiento se realiza automáticamente, con PPL el diseñador es quien, de acuerdo a su experiencia, realiza esta tarea.

10. El enrutamiento de señales se realiza utilizando celdas "blank", celdas de interconexión y los caminos horizontales y verticales que atraviesan cualquier celda. Este esquema de enrutamiento permite un ahorro sustancial del área requerida.
11. En PFL, el diseñador no debe preocuparse de la distribución de  $V_{DD}$  y Gnd, para él la presencia de estos caminos hacia todas las celdas es transparente.
12. El entender la lógica mixta utilizada por PFL para la implantación de funciones AND y OR permitió implantar fácilmente las ecuaciones de las MEFs del codificador y decodificador.
13. El PFL dispone de una estructura en la que se encuentran predeterminadas las posiciones de los PADs (PAD-FRAMES) y en la que el diseñador debe preocuparse simplemente de insertar su núcleo y enrutar las señales de entrada y salida hacia los PADs, ello permite:
  - i) Que el diseñador se concentre en el diseño de su aplicación específica.
  - ii) Obtener los archivos de descripción (CIF) en tiempos más cortos.
  - iii) Garantizar que las fallas de los circuitos se deben a errores en el diseño del núcleo y no de los PADs.
  - iv) Facilitar la comunicación entre el diseñador y las

fundidoras, que son las que generalmente distribuyen los *PAD-FRAMES*.

14. A pesar de que el área disponible en el *PAD-FRAME* es suficiente para insertar el codificador y decodificador, los diseños se realizaron buscando siempre optimizar el área.
15. El *PAD-FRAME* limita el tamaño de los diseños realizados o como en el presente caso puede quedar cierta área no utilizada, que podría aprovecharse para insertar otros diseños.
16. En PPL, gracias a la metodología utilizada, el diseño presenta gran flexibilidad; el realizar un cambio en cualquiera de los niveles de diseño, se resume a modificar la disposición de las celdas en el editor *TILER* y a la generación completamente automática de los archivos (.ppx) para la simulación, con lo cual se concluye el diseño.
17. La propiedad mediante la cual se pueden intercambiar filas y columnas y continuar realizando la misma función fue utilizada muchas veces durante el diseño del codificador y decodificador y permitió optimizar el enrutamiento de las señales, el número de celdas de interconexión necesaria, lo que a la vez condujo a la optimización de área requerida.

18. La metodología PPL permitió dividir el diseño del codificador y decodificador en pequeños módulos, de tal manera que se puede depurar cada uno de ellos para luego integrarlos en uno solo.
19. Los errores encontrados durante el DRC no son frecuentes, ya que las celdas están concebidas individualmente siguiendo las reglas de diseño. Además, gracias a que el paquete PPL tiene un alto grado de depuración, las celdas también consideran las reglas de diseño al posicionarlas junto a otras celdas. Generalmente, los errores del DRC son causados por el usuario y se deben a la ubicación de una celda junto a otra que no es compatible.
20. A continuación se presenta un cuadro comparativo del área y número de transistores utilizados, y un cálculo de las densidades promedio conseguidas con cada uno de los paquetes de diseño, para el codificador y decodificador. Las áreas que se consideran no incluyen los PADS y para el caso del TENTOS son aproximadas.

	CODIFICADOR		DECODIFICADOR	
	TENTOS	PPL	TENTOS	PPL
TRANSISTORES	682	568	604	516
DIMENSIONES (mm)	1.388 x 0.960	1.500 x 0.506	1,388 x 0.960	1,650 x 0.462
AREA (mm <sup>2</sup> )	1.332	0.759	1.332	0.762
DENSIDAD (TRAN/mm <sup>2</sup> )	512	748	453	677

El área total utilizada para el codificador y decodificador, incluidos los PADs, para el caso del TENTOS es de aproximadamente  $6.179 \text{ mm}^2$  ( $2.771 \text{ mm} \times 2.230 \text{ mm}$ ). Para el PFL, se debe considerar que a pesar que el área utilizada por el codificador y decodificador sin PADs es menor en un factor cercano a 2, el *PAD-FRAME* tiene dimensiones fijas, y en este caso contiene una área no utilizada. El área utilizada por el *PAD-FRAME tiny34* es  $5 \text{ mm}^2$  ( $2.220 \text{ mm} \times 2.250 \text{ mm}$ ).

#### 12.4 RECOMENDACIONES

1. El nivel actual de equipamiento y de software que ha logrado el Grupo de Microelectrónica cumple a cabalidad tan solo con la función de capacitación de recursos humanos en la concepción y diseño de circuitos integrados. Queda en evidencia que un salto cualitativo en la complejidad de los circuitos integrados de investigación únicamente será posible con la dotación de estaciones de trabajo más poderosas y herramientas computacionales más sofisticadas,
2. Los principios de simulación tanto eléctrica SPICE como lógica NDL planteados, han sido aplicados al diseño de circuitos digitales VLSI, no obstante los fundamentos son válidos para la simulación de circuitos eléctricos y electrónicos en general por lo que se recomienda su

difusión y aplicación en la enseñanza de las cátedras de Circuitos Eléctricos, Electrónica y Sistemas Digitales.

3. El programa de síntesis TRAGO del paquete TENTOS resuelve los limitantes respecto al aprovechamiento de área y a la falta de libertad en el enrutamiento horizontal que presenta la síntesis TRAMO, se sugiere realizar un estudio detallado de los layouts que genera este tipo de síntesis, partiendo del entendimiento alcanzado en el análisis de los resultados y limitaciones del programa de síntesis TRAMO.
4. El método de síntesis TRAMO del paquete TENTOS presenta una serie de errores en la generación de layouts que oportunamente fueron detallados, estos errores fueron corregidos en forma manual mediante el editor de máscaras EMA2, se sugiere proponer nuevas topologías de celdas y/o algoritmos de enrutamiento que permitan generar layouts sin errores de manera automática.
5. Para visualizar los layouts de los diseños de PPL, se puede adquirir el editor de máscaras LEDIT, que acepta archivos con formato CIF, por lo que podría utilizarse para editar cualquier layout descrito en este formato.
6. Se sugiere incursionar en el estudio de las configuraciones lógicas CMOS alternativas a la "lógica totalmente complementaria", ya que ofrecen ventajas en cuanto a



área y velocidad, aunque presentan también algunas desventajas.

7. Se recomienda realizar los contactos necesarios para conseguir la fabricación de los circuitos diseñados con el paquete PPL.

## BIBLIOGRAFIA

1. AGUILO JORDI - TERES LLUIS, ASIC's: Metodologías y herramientas de diseño, 1991.
2. BONNEVILLE MICROELECTRONICS Inc., Path Programmable Logic, System Documentation, U.S.A., 1990.
3. BONNEVILLE MICROELECTRONICS Inc., FORESIGHT and MOSIS accessed by PPL, U.S.A., 1990.
4. CARR WILLIAM - MIZE JACK, MOS/LSI Design and Application, McGraw-Hill, U.S.A., 1972.
5. CENTRO NACIONAL DE MICROELECTRONICA, Servicio MPC del CNM, Barcelona - España, 1991.
6. CTI - INSTITUTO DE MICROELECTRÓNICA, Empacotamento Eletrônico de Circuitos Integrados, Brasil, 1991.
7. DA SILVA - MEDINA A., EDSMB: Editor de Símbolos. VI Seminário Interno de Microeletrônica. UFRGS - Brasil, 1990.

8. EUROPEAN SILICON STRUCTURES, Solo 1400 Getting Started, Reino Unido, 1990.
9. LATHI B., Ingeniería Electrónica VOL.4, Ed. Interamericana, México, 1986.
10. MAMMANA Carlos - MACHADO HELENA, Manual de Usuário do Sistema Didático de Projetos, Kapelusz, 1987.
11. MEAD CARVER, Analog VLSI and Neural Systems, Addison Wesley, U.S.A., 1989.
12. MEAD CARVER - CONWAY LYNN, Introduction to VLSI Systems, Addison Wesley, U.S.A., 1980.
13. MILLMAN JACOB - HALKIAS CHRISTOS, Integrated Electronics: Analog and Digital Circuits and Systems, McGraw-Hill, U.S.A., 1972.
14. MORAES F. - REIS R., Manual do usuário do projeto TRANCA, UFGRS - Brasil, 1991.
15. MORAES F. - REIS R., TENTOS: Gerenciador de software para Microelectrônica, UFGRS - Brasil, 1991.
16. MUKHERJEE AMAR, Introduction to nMOS and CMOS VLSI Systems Design, Prentice Hall, New Jersey, 1986.

17. OWEN FRANK, FCM and Digital Transmission Systems, McGraw-Hill, U.S.A., 1982.
18. PHASE THREE LOGIC, Inc., CAFFAST Electronic Circuit Design, CAE User's Manual, U.S.A., 1991.
19. REIS RICARDO - STEMMER MARCOS, EXTRIBO: Uma versão Corrigida e Melhorada do Extrator Hierárquico de Circuitos. VI Seminário Interno de Microeletrônica. UFRGS - Brasil 1990.
20. SACHET DANIEL - PEREIRA EDUARDO, ESQUELETO: Editor de Esquemas Eléctricos. VI Seminário Interno de Microeletrônica. UFRGS - Brasil 1990.
21. SCHMITZ EBER, TEDMOS IV: Turbo Editor para circuitos integrados MOS - Manual de operación, Universidade Federal do Rio de Janeiro - Brasil, 1991.
22. SMITH K. - GU J., A Structured Approach for VLSI Circuit Design. IEEE. University of Utah, U.S.A., 1989.
23. TURRINI SILVIO, VLSI Technology, Second Course on basic VLSI Design Techniques, ICTP, Italia, 1991.
24. TUINENGA PAUL, SPICE - A guide to circuit Simulation and Analysis Using Pspice, Prentice Hall, New Jersey, 1988.

25. VAN DER ZIEL ALDER, Solid State Physical Electronics, Prentice Hall, New Jersey, 1968.
  
26. WESTE NEIL - ESHRAGHIAN KAMRAN, Principles of CMOS VLSI Design, Addison Wesley, U.S.A., 1988.

**ANEXO A**

**DISPOSITIVOS Y COMANDOS SPICE  
EMPLEADOS EN SIMULACION  
VLSI DIGITAL**

## A.1 DISPOSITIVOS CIRCUITALES DE LA NETLIST SPICE<sup>(1)</sup>

La esencia del programa SPICE radica en la gran variedad de dispositivos que posee en su biblioteca de elementos, cuyo comportamiento se simula a partir de modelos matemáticos, que permiten el análisis de la mayor parte de circuitos eléctricos. A continuación se describen los elementos fundamentales empleados en la simulación de Circuitos Integrados VLSI Digitales de tecnología CMOS según su uso en el archivo del circuito.

### OBSERVACIONES:

- a) Las estructuras encerradas entre "< >" son indispensables en su correspondiente estructura sintáctica.
- b) Las estructuras encerradas entre "[ ]" son opcionales en la estructura sintáctica a que pertenecen.

#### A.1.1 Capacitor "C"

Sintaxis: C<nombre> <nodo +> <nodo -> [nombre del modelo]  
+ <valor> [IC = <valor inicial>]

Ejemplo: Cfdbck 3 5 CMOD 10pF

---

<sup>(1)</sup> "A Guide To Circuit Simulation and Analysis Using PSpice", P. Tuinenga, Appendix B.

Parámetros del Modelo (ver comando ".MODEL")		Valor default	Unidades
C	multiplicador de capacitancia	1	
VC1	coeficiente de voltaje lineal	0	volt <sup>-1</sup>
VC2	coeficiente de voltaje cuadrático	0	volt <sup>-2</sup>
TC1	coeficiente de temperatura lineal	0	'C <sup>-1</sup>
TC2	coeficiente de temperatura cuadrático	0	'C <sup>-2</sup>

- a) C<nombre> corresponde a la identificación del capacitor.
- b) Los nodos (+) y (-) definen la polaridad del capacitor de modo que la corriente fluye del nodo (+) al (-).
- c) El valor del capacitor que puede ser positivo(+) o negativo(-) pero nunca cero (0), sus unidades serán *Faradios* de no indicarse lo contrario en [nombre del modelo].
- d) La capacitancia viene dada por el modelo matemático:  

$$\langle \text{valor} \rangle \cdot C \cdot (1 + VC1 + VC2 \cdot V^2) \cdot (1 + TC1 \cdot (T - T_{nom}) + TC2 \cdot (T - T_{nom})^2)$$
- e) <valor inicial> es el valor inicial para el capacitor durante el cálculo del punto de polarización.

#### A.1.2 Mosfet "M"

Sintaxis: M<nombre> <nodo de drenaje> <nodo de compuerta>  
+ <nodo de fuente> <nodo de sustrato>  
+ <nombre de modelo> [L = <valor>] [W = <valor>]  
+ [AD = <valor>] [AS = <valor>] [PD = <valor>]  
+ [PS = <valor>] [NRD = <valor>] [NRS = <valor>]  
+ [NRG = <valor>] [NRB = <valor>]

Ejemplo 1: MN6 2 13 0 0 NMOS L=3.0U W=4.0U



Ejemplo 2:      MP20 9 11 1 1      PMOS L=20.1U W=70.5U  
                 + AD=16000P AS=11200P PD=620U PS=460U

El MOSFET es modelado como MOSFET intrínseco con resistencias óhmicas en serie con drenaje, fuente, compuerta y sustrato. Existe también una resistencia (RDS) en paralelo con el canal drenaje-compuerta.

a) L y W son la longitud y ancho del canal.

$$L_{\text{EFFECTIVO DEL CANAL}} = L - 2 \cdot LD$$

$$W_{\text{EFFECTIVO DEL CANAL}} = W - 2 \cdot WD$$

b) L y W pueden ser especificados en el dispositivo, el modelo, o el comando .OPTIONS; teniendo ese orden de prioridad de tenerse simultáneamente varias especificaciones.

c) AD y AS son las áreas de difusión de drenaje y fuente.

d) PD y PS son los perímetros de difusión de drenaje y fuente.

e) Las corrientes de saturación drenaje-sustrato y fuente-sustrato pueden ser especificadas ya sea por JS, que es multiplicado por AD y AS respectivamente, o por IS, que es un valor absoluto.

Parámetros del Modelo (ver comando ".MODEL")		Valor default	Unidades
LEVEL	nivel de modelo (1, 2 ó 3)	1	
L	longitud del canal	DEFL	metro
W	ancho del canal	DEFW	metro
LD	longitud de la difusión lateral	0	metro
WD	ancho de la difusión lateral	0	metro
VTO	voltaje de umbral de polarización-cero	0	voltio
KP	transconductancia intrínseca	2E-5	amp/volt <sup>2</sup>
GAMMA	parámetro de umbral de sustrato	0	voltio <sup>1/2</sup>
PHI	potencial superficial	.6	voltio
LAMBDA	modulación de canal (NIVEL= 1 ó 2)	0	voltio <sup>-1</sup>
RD	resistencia óhmica de drenaje	0	ohmio
RS	resistencia óhmica de fuente	0	ohmio
RG	resistencia óhmica de compuerta	0	ohmio
RB	resistencia óhmica de sustrato	0	ohmio
RDS	resistencia en paralelo drenaje-fuente	infinito	ohmio
RSH	resistencia laminar difusión, fuente, drenaje	0	ohm/cuadrado
IS	corriente de saturación de juntura	1E-14	amp
JS	corriente/área de saturación de juntura	0	amp/m <sup>2</sup>
PB	potencial de juntura con sustrato	.8	volt
CBD	capacitancia de juntura drenaje-sustrato	0	faradio
CBS	capacitancia de juntura fuente-sustrato	0	faradio
CJ	capacitancia/área de juntura (fondo)	0	faradio/m <sup>2</sup>
CJSW	capacitancia/longitud de juntura (lateral)	0	faradio/m
MJ	gradiente de juntura (fondo)	.5	
MJSW	gradiente de juntura (lateral)	.33	
FC	coeficiente de capacitancia de polarización de juntura	.5	
CSGO	capacitancia/ancho de canal de sobreposición compuerta/fuente	0	faradio/m
CSDO	capacitancia/ancho de canal de sobreposición compuerta/drenaje	0	faradio/m
CSBO	capacitancia/ancho de canal de sobreposición sustrato/compuerta	0	faradio/m
NSUB	dopaje de sustrato	0	1/cm <sup>3</sup>
NSS	densidad de estado de superficie	0	1/cm <sup>2</sup>
NFS	densidad de estado superficial rápido	0	1/cm <sup>2</sup>
TOX	ancho de óxido	infinito	metro
TPG	tipo de compuerta	+1	
	+1 = opuesta al sustrato		
	-1 = similar al sustrato		
	0 = aluminio		
XJ	profundidad de juntura metalúrgica	0	metro
UO	movilidad superficial	600	cm <sup>2</sup> /volt·sec
UCRIT	campo crítico de degradación de movilidad (NIVEL = 2)	1E4	voltio/cm <sup>2</sup>
UEXP	exponente de degradación de movilidad (NIVEL = 2)	0	
VMAX	tendencia de velocidad máxima	0	metro/sec
NEFF	coeficiente de carga de canal (NIVEL = 2)	1	
XOC	fracción de carga de canal atribuida al drenaje	1	
DELTA	efecto de ancho en umbral	0	
THETA	modulación de movilidad (NIVEL = 3)	0	1/volt
ETA	realimentación estática (NIVEL = 3)	0	
KAPPA	factor de campo de saturación (NIVEL = 3)	.2	
KF	coeficiente de ruido de vibración	0	
AF	exponente de ruido de vibración	1	

- f) Las capacitancias de agotamiento de polarización-cero pueden ser especificadas por CJ, que es multiplicado por AD y AS, y por CJSW, que es multiplicado por PD y PS. O pueden ser establecidas por CBD y CBS, que son valores absolutos.
- g) NRD, NRS, NRG y NRB son resistencias relativas a drenaje, fuente, compuerta y sustrato en cuadrados. Estas resistencias óhmicas parásitas pueden ser especificadas ya sea por RSH, que es multiplicado por NRD, NRS, NRG y NRB respectivamente, o por RD, RS, RG, y RB, que son valores absolutos.
- h) Los valores por default de los parámetros son:

L = 100μ	W = 100μ	AD = 0
AS = 0	PD = 0	PS = 0
NRD = 1	NRS = 1	NRG = 0
NRB = 0		

L, W, AD, y AS pueden ser establecidos en el comando ".OPTIONS".

### A.1.3 Subcircuito "X"

Sintaxis: X<nombre> [nodo 1].....[nodo n]  
+ <nombre de subcircuito>

Ejemplo: X6 PRESET q3\_1 q5\_1 q6\_1 vcc NAND3

Esta estructura permite que el subcircuito referido en <nombre de subcircuito> sea insertado como un dispositivo X(nombre), similar a los otros dispositivos, en el circuito. Los nodos en la invocación X deben ser correspondientes con aquellos de la definición del subcircuito (Ver comando ".SUBCKT").

Las llamadas a subcircuitos pueden ser anilladas, es decir que pueden haber dispositivos X(nombre) en la definición de los subcircuitos.

#### A.1.4 Fuente de tensión independiente "V"

Sintaxis: V<nombre> <nodo +> <nodo -> [[DC] <valor>]  
+ [AC <valor de magnitud> [valor de fase]]  
+ [especificación de transitorio]

Ejemplo 1: Vpolariz 1 0 DC 5V

Ejemplo 2: VAC 2 3 AC .001 90

Ejemplo 3: VPULSO 1 0 PULSE (-1mv 1mv 2ns 2ns 50ns 100ns)

V<nombre> corresponde a una fuente de voltaje en que la corriente fluye desde el nodo(+) hacia el nodo(-).

- a) Ninguno, algunos o todos los valores de DC, AC y valores transitorios pueden ser especificados. En caso de omitirse alguno de ellos su valor será cero (0).
- b) El valor de fase de AC es en grados.

c) La especificación de transitorio se analiza con mayor detalle a continuación.

### Análisis de transitorios

El simulador SPICE permite, dentro de las fuentes de señal de tensión (o de corriente) independientes, la inclusión de excitaciones transitorias con las siguientes formas de onda:

EXP	forma de onda exponencial.
PULSE	forma de onda de pulso repetitivo.
PWL	formas de onda lineal en trozos discretos.
SFFM	formas de onda de frecuencia modulada.
SIN	forma de onda senoide.

Para la simulación de sistemas VLSI digitales son de interés los análisis transitorios para dos tipos de formas de onda en particular:

i) Forma de onda lineal en trozos discretos "PWL"

Sintaxis: PWL (<t1> <v1> <t2> <v2>...<ti> <vi>...<tn> <vn>)

<ti>, <vi> corresponden a los puntos de inflexión (Fig. A.1), sus unidades son segundos y voltios respectivamente, y sus valores por omisión son cero (0).

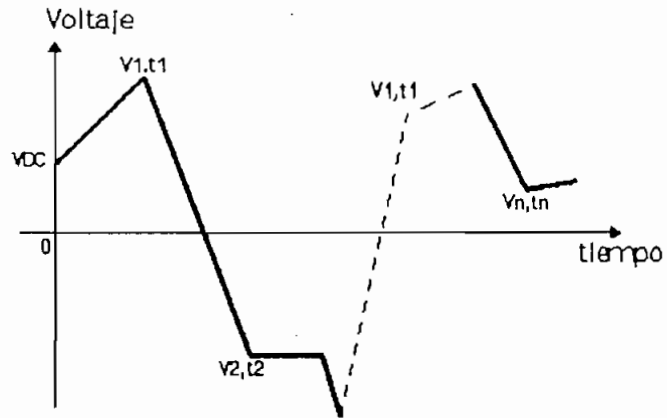


FIGURA A.1 Forma de onda lineal en trozos discretos.

ii) Forma de onda de pulsos "PULSE"

Sintaxis: PULSE (<v1> <v2> <td> <tr> <tf> <pw> <per>)

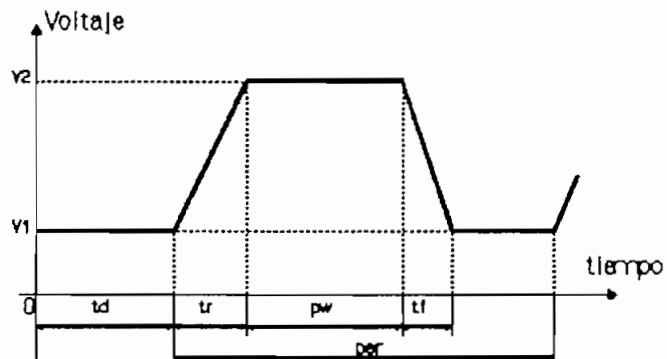


FIGURA A.2 Forma de onda de pulsos.

<v1> y <v2> se expresan en voltios, <t delay>, <t raise> <t fall>, <pulse width> y <period> se expresan en segundos, y su significado se indica en la Fig.A.2.

## A.2 COMANDOS DE SIMULACION SPICE<sup>(2)</sup>

A continuación se revisan brevemente, en orden alfabético los comandos de control disponibles en SPICE para la simulación de diseños VLSI Digitales. Cada comando es descrito por su uso en el archivo de circuito con algunos comentarios de su uso.

### A.2.1 Análisis AC ".AC"

Sintaxis: .AC [LIN][OCT][DEC] <número de puntos>  
+ <valor de frecuencia inicial>  
+ <valor de frecuencia final>

Ejemplo: .AC LIN 50 100Hz 10KHz

El comando .AC instruye el cálculo de la respuesta de frecuencia de un circuito sobre un rango de frecuencias. La forma de variación de la frecuencias puede ser:

- a) [LIN] que indica una variación lineal en que <número de puntos> corresponde al número total de puntos de variación.
- b) [OCT] que corresponde a una variación logarítmica por octavas donde <número de puntos> indica el número de puntos por cada octava.

---

<sup>(2)</sup> "A Guide To Circuit Simulation and Analysis Using PSpice", P. Tuinenga, Appendix A.

- c) [DEC] que indica una variación logarítmica por décadas donde <número de puntos> corresponde al número de puntos por cada década.

#### A.2.2 Análisis DC ".DC"

Sintaxis: .DC [forma de variación] <fuente a ser variada>  
+ <parámetros de variación>

Ejemplo: .DC Vin 0 5 1mv

El comando de análisis de variación .DC instruye el cálculo del punto de polarización del circuito sobre un rango de valores que toma una <fuente a ser variada>.

- a) Opcionalmente se indica la [forma de variación] de los valores de la fuente, que puede ser lineal [LIN], por octavas [OCT], por décadas [DEC], o a partir de una lista de valores [LIST]; los <parámetros de variación> dependerán de la forma de variación escogida.
- b) En caso de omitirse esta especificación se asumirá una variación lineal [LIN] de los valores y los parámetros serán: <valor inicial> <valor final> <valor de incremento>.

#### A.2.3 Fin de circuito ".END"

Sintaxis: .END



El comando `.END` marca el final de un circuito, todos los datos y comandos deben antecederlo. Se pueden tener varios circuitos separados por `.END` de modo que al iniciar el análisis del siguiente circuito todas las variables son inicializadas.

#### A.2.4 Fin de definición de subcircuito `".ENDS"`

Sintaxis: `.ENDS [nombre de subcircuito]`

Ejemplo: `.ENDS NAND2`

El comando `.ENDS` marca el final de la definición de un subcircuito (iniciada por el comando `.SUBCKT`).

#### A.2.5 Condiciones transitorias iniciales `".IC"`

Sintaxis: `.IC <V(<nodo 1>)=<valor>> ... <V(<nodo n>)=<valor>>`

Ejemplo: `.IC V(2)=5V V(3)=10mv`

`.IC` es usado para indicar las condiciones iniciales de los nodos especificados, para el análisis transitorio. Cada `<valor>` es un voltaje asignado a un `<nodo>` únicamente durante la duración del cálculo del punto de polarización para el análisis transitorio, luego de lo cual el nodo es liberado.

#### A.2.6 Inclusión de archivo `".INC"`

Sintaxis: `.INC [nombre de archivo]`

Ejemplo: .INC CELDA.CIR

El comando .INC instruye la inclusión de todas las líneas de otro archivo dentro del archivo del circuito en curso. De existir el comando .END en el nuevo archivo, este sólo marca el final del archivo incluido. Se pueden tener comandos .INC anidados hasta el cuarto nivel de inclusión.

A.2.7 Archivo de biblioteca ".LIB"

Sintaxis: .LIB [nombre de archivo]

Ejemplo: .LIB \GME\LIB\CELDAS.LIB

A diferencia del comando .INC, .LIB es usado para referirse únicamente del contenido asociado a los comandos .MODEL, .LIB, definiciones de subcircuitos y comentarios en el archivo indicado. Si [nombre de archivo] es omitido se asumirá "NOM.LIB".

A.2.8 Modelo de dispositivo ".MODEL"

Sintaxis: .MODEL <nombre> <nombre tipo>  
+ ([<nombre de parámetro> <valor>  
+ [especificación de tolerancia]])

Ejemplo <sup>(3)</sup> :

```

**-----**
**  MODELO PARA TRANSISTOR NMOS - EUROPEAN SILICON STRUCTURES  **
**                TYPICAL CASE  1,5 um                **
**-----**

.MODEL          NMOS          NMOS          LEVEL=2          LD=0.325U
TOX=250E-10    +NSUB=2E16    VTO=0.7          UO=510           UEXP=0.22
UCRIT=24.3K    DELTA=0.4      +XJ=0.4U        VMAX=54K        NEFF=4
RSH=55         NFS=0        JS=2U           +CJ=130U        CJSW=620P
MJ=0.53       MJSW=0.53    PB=0.68V       CGDO=320P       +CBSO=320P

**-----**
**  MODELO PARA TRANSISTOR PMOS - EUROPEAN SILICON STRUCTURES  **
**                TYPICAL CASE  1,2um                **
**-----**

.MODEL          PMOS          PMOS          LEVEL=2          LD=0.3U
TOX=250E-10    +NSUB=5E16    VTO=-1.1        UO=210           UEXP=0.33
UCRIT=51K      DELTA=0.4      XJ=0.5U        VMAX=47K        NEFF=0.88
RSH=75         NFS=0        JS=10U         CJ=490U         CJSW=590P
MJ=0.46       MJSW=0.46    PB=0.78V       CGDO=320P       +CBSO=320P

```

El comando .MODEL define un conjunto de parámetros de dispositivo que pueden ser empleados por elementos en el circuito.

- a) <nombre> es el nombre del modelo que usa el dispositivo, pudiendo existir varios modelos para un mismo elemento.
- b) <nombre tipo> es el tipo de dispositivo, en diseño VLSI digital se emplean los siguientes tipos:

CAP	Capacitancia
NMOS	MOSFET canal-N
PMOS	MOSFET canal-P

---

<sup>(3)</sup> "Reference AGI-DR04-A", European Silicon Structures (ES2), pág. 23.

c) Luego de <nombre tipo> se tiene una lista de valores de parámetros encerrados entre paréntesis, pudiendo asignarse valores a todos, algunos o ninguno de los parámetros del modelo en cuyo caso se asignarán los valores por omisión. *La lista de nombres de parámetros, significados y valores por omisión se especifica en las descripciones individuales de cada dispositivo.*

A.2.9 Establecimiento de nodos ".NODESET"

Sintaxis: .NODESET <V(<nodo 1>)=<valor>> +....  
+ <V(<nodo n>)=<valor>>

Ejemplo: .NODESET V(1)=2mV V(5)=5V

Provee una ayuda para el cálculo del punto de polarización en los nodos que involucra, a diferencia de .IC no provee un valor inicial definido al nodo, sino únicamente una aproximación para su cálculo. Se emplea mucho para romper los lazos indefinidos que se pueden formar en circuitos bistables o flip-flops.

A.2.10 Opciones ".OPTIONS"

Sintaxis: .OPTIONS [nombre de la opción]  
+ [<nombre de la opción>=<valor>]

Ejemplo: .OPTIONS ACCT RELTOL=1E-3

El comando `.OPTIONS` es usado para establecer las opciones, límites y parámetros de control para los análisis que realiza el simulador. Existen dos clases de opciones: con valores, y sin ellos. Las opciones sin valores son banderas de diferentes tipos que se activan solo con nombrarlas (se asume que todas las banderas se hallan por default desactivadas).

#### A.2.11 Esquemmatización de resultados ".PLOT"

Sintaxis: `.PLOT [DC] [AC] [NOISE] [TRAN]`  
+ [variables de salida] ([<valor límite inferior>,<valor límite superior>])

Ejemplo 1: `.PLOT DC VGD(3) I(R2) I(VIN) VGS(5)`

Ejemplo 2: `.PLOT TRAN V(20) V(21) V(4) (-5V,5V)`

Permite que los resultados de los análisis DC, AC, ruido, y transitorios sean exhibidos, para el intervalo indicado, en forma de diagramas de "impresora en línea" realizados en base a caracteres de texto.

#### A.2.12 Impresión de resultados ".PRINT"

Sintaxis: `.PRINT [DC] [AC] [NOISE] [TRAN]`  
+ [variables de salida]

Ejemplo: `.PRINT TRAN V(20) V(21) V(4)`

Permite que los resultados del análisis DC, AC, ruido, y transitorios sean indicados en forma de tablas de valores numéricos.

#### A.2.13 Graficación de resultados ".PROBE"

Sintaxis: .PROBE [variables de salida]

Ejemplo: .PROBE Vpolariz(3) VBC(5) VCE(2) IB(R23)

Traslada los resultados de los análisis DC, AC y transitorios a un archivo de datos denominado PROBE.DAT usado por el programa post-procesador de gráficos PROBE.

- a) Si no se especifican los nombres de las variables de salida, se almacena la información de todos los voltajes de nodo y corrientes de red del circuito.
- b) A diferencia de .TRAN y .PRINT no se especifica tipo de análisis antes de las variables de salida.

#### A.2.14 Definición de subcircuitos ".SUBCKT"

Sintaxis: .SUBCKT <nombre> [nodo 1].....[nodo n]

Ejemplo: .SUBCKT NAND3 I1 I2 I3 OUT vcc

El comando .SUBCKT inicia la definición de un subcircuito que es finalizada con el comando .ENDS.

- a) El subcircuito es integrado a la red principal siempre que es invocado desde esta con una X.
- b) <Nombre> es la denominación del subcircuito y es usada en la estructura de la invocación X para identificarlo.
- c) Los subcircuitos pueden ser anidados, es decir que cada subcircuito puede a su vez contener invocaciones a otros subcircuitos.
- d) La estructura [nodo 1]...[nodo n] debe ser correspondiente a aquella de la invocación X. Cuando el subcircuito es llamado, los nodos reales contenidos en X reemplazan a los nodos del argumento del comando .SUBCKT de la definición.
- e) Las definiciones de subcircuitos no pueden ser anidadas, es decir, no pueden haber comandos .SUBCKT entre los comandos .SUBCKT y .ENDS originales.
- f) Solo pueden haber descripciones de dispositivos y comandos .MODEL dentro de la definición de un subcircuito.
- g) Los nombres de nodos, dispositivos y modelos son locales al subcircuito en que son definidos, de existir nombres iguales en otros subcircuitos, no existe posibilidad de conflicto.

#### A.2.15 Análisis transitorio ".TRAN"

Sintaxis: .TRAN [/OP] <valor de paso de impresión>  
+<valor de tiempo final> [<valor de no-impresión>  
+[valor máximo de paso]] [UIC]

Ejemplo: .TRAN/OP 1ns 100ns 20ns UIC

El comando .TRAN instruye la realización del análisis transitorio en el circuito que se realiza desde TIEMPO=0 hasta <valor de tiempo final>.

- a) El análisis transitorio usa un intervalo de tiempo que es autoajustado internamente de modo que es incrementado en intervalos de baja actividad, y decrementado en intervalos de gran variación.
- b) Los resultados de este análisis son enviados al archivo de salida, para los tiempos requeridos, en caso de no coincidir los tiempos de cálculo con los de impresión los resultados son extrapolados.
- c) Aún cuando el análisis transitorio siempre se inicia en TIEMPO=0, se puede suprimir la exhibición de los resultados desde  $t=0$  hasta <valor de no-impresión>.
- d) [valor máximo de paso] indica el máximo intervalo de análisis para intervalos de baja actividad, y es por omisión igual a <valor de tiempo final>/50.



- e) /DP indica la exhibición de los resultados detallados del análisis del punto de polarización.
  
- f) La clave UIC (Use Initial Conditions), es factible de emplearse cuando se han indicado condiciones iniciales mediante el comando .IC, y de ser usada instruye la omisión del cálculo del punto de polarización.

**ANEXO B**

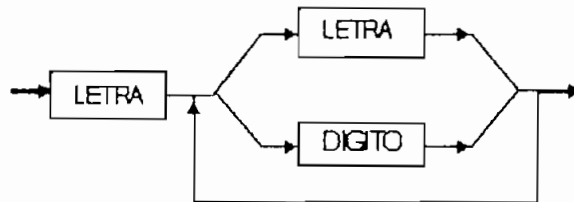
**DISPOSITIVOS Y COMANDOS  
EMPLEADOS EN SIMULACION  
LOGICA HDL**

## B.1 ELEMENTOS CIRCUITALES DE LA NETLIST NDL<sup><4></sup>

### OBSERVACIONES:

En las descripciones sintácticas se han adoptado las siguientes convenciones:

- a) Las estructuras encerradas entre "< >" son indispensables en la estructura sintáctica, por tanto no pueden ser obviadas.
- b) Las estructuras encerradas entre "[ ]" son opcionales en las estructuras sintácticas, y pueden ser omitidas.
- d) nd \*\*\* : nombre de un nodo definido en el circuito;  
Cabe recordar su estructura estudiada anteriormente:



de donde se desprende que el primer caracter es siempre una letra, además, la secuencia de caracteres puede ser tan larga cuanto el usuario desee.

- e) El programa de simulación NDL ha sido desarrollado en la UFRJ - Brasil, a ello se debe que la sintaxis de las directivas de simulación emplea terminología en idiomas Inglés y Portugués.

---

<4> \*TEDMOS IV: TURBO EDITOR PARA CIRCUITOS INTEGRADOS MOS\*, E. Schmitz - J.Assis - ... - R. Merino. Universidade Federal do Rio de Janeiro - Brasil.

### B.1.1 Compuertas

#### a) Inversor "INVERT"

Sintaxis: INVERT <nd\_salida> <nd\_entrada>;

Ejemplo: INVERT out in;

#### b) NAND de $n$ entradas "NAND"

Sintaxis: NAND <nd\_salida> <nd\_ent1> ... <nd\_entn>;

Ejemplo: NAND out in1 in2 in3;

#### c) NOR de $n$ entradas "NOR"

Sintaxis: NOR <nd\_salida> <nd\_ent1> ... <nd\_entn>;

Ejemplo: NOR salida ent1 ent2 ent3;

### B.1.2 Transistores

#### a) Transistor de carga "TRANS"

Sintaxis: TRANS <nd\_compuerta> <nd\_fuente> <nd\_drenaje>;

Ejemplo: TRANS nod\_gate nod\_source nod\_drain;

#### b) Transistor NMOS "NTRANS"

Sintaxis: NTRANS <nd\_compuerta> <nd\_fuente> <nd\_drenaje>;

Ejemplo: NTRANS In Gnd Out;

#### c) Transistor PMOS "PTRANS"

Sintaxis: PTRANS <nd\_compuerta> <nd\_fuente> <nd\_drenaje>;

Ejemplo: PTRANS In Vdd Out;

### B.1.3 Tipos de nodos

#### a) Nodos de pull-up "PULLUP"

Sintaxis: PULLUP <nd\_1> \*\*\*\*\* <nd\_n >;

Ejemplo: PULLUP nodo1 nodo2 nodo3;

Los nodos corresponden a aquellos conectados a un transistor de carga.

#### b) Nodo de entrada "INPUT"

Sintaxis: INPUT <nd\_1> \*\*\*\*\* <nd\_n>;

Ejemplo: INPUT In\_mux1 In\_mux2 In\_mux3;

Los nodos de entrada corresponden a aquellos a los que se puede asociar una fuente de tensión con capacidad de carga infinita.

#### c) Nodo normal

No existe directiva alguna que permita caracterizar a un nodo como normal. El programa internamente asume que todos los nodos son "*normales*", salvo aquellos que específicamente hayan sido declarados como de *entrada* o de *pull-up*.

### B.1.4 Llamada a subcircuitos

Sintaxis: <NOMBRE\_SUBCKT> <nd\_interf1> \*\*\* <nd\_interfn>;

Ejemplo: MUX4a1 A1 A2 A3 A4 ctrl1 ctrl2 B;

El subcircuito es tratado por el compilador NDL como un elemento circuital similar a los restantes, con sus mismas características y sintaxis. En este caso se tiene que:

- a) <NOMBRE\_SUBCKT> corresponde al nombre asignado al subcircuito en su definición en el comando "DEFINE".
  
- b) <nd\_interf1> ... <nd\_interfn> son los nodos de interfaz con los que el subcircuito se interconecta con el circuito principal, estos nodos deben hallarse en el mismo orden que sus correspondientes en la definición del subcircuito (ver comando DEFINE), no obstante, sus nombres no necesariamente deben ser los mismos.
  
- d) Las llamadas a subcircuitos pueden ser recursivas, es decir que pueden haber llamadas a subcircuitos dentro de la definición de un subcircuito.

## B.2 COMANDOS DE SIMULACION NDL

Con excepción de los comandos de "definición circuital", los comandos en el simulador NDL pueden ser ejecutados de manera interactiva conforme la simulación progresa, o pueden ser especificados en un archivo de comandos, en ambos casos los resultados son los mismos.

Siempre que el simulador se halla listo para recibir un comando indica un cursor con el formato:

```
#ciclo.#fase>
```

como por ejemplo: 0:0>

## OBSERVACIONES

Además de las convenciones adoptadas anteriormente, en la descripción de los elementos circuitales, se han adoptado las convenciones siguientes para la descripción de los comandos de simulación:

- a) valor : 0, 1 ó X
- b) lista\_valor : una secuencia de 0s, 1s y/o Xs.
- c) n\_veces : un número positivo
- d) canal : un número entre 1 y 10
- e) Los comandos pueden ser dados en letras mayúsculas o minúsculas, en la sintaxis se indican en mayúsculas los nombres abreviados que pueden ser usados para su invocación.

### B.2.1 Comandos de Definición circuital

#### a) Definición de tecnología "FAMILY"

Sintaxis: FAMILY <[CMOS] [NMOS]>;

Ejemplo: FAMILY CMOS;

Indica la tecnología de diseño del circuito descrito, misma que puede ser CMOS (MOS Complementario) ó NMOS, lo que influye directamente en el tipo de nodos admitidos en las estructuras circuitales de la NETLIST.

**b) Definición de subcircuito**

Sintaxis: DEFINE <NOMBRE\_SUBCKT> <nd\_interf1>...<nd\_interfn>;

Ejemplo: DEFINE MUX4a1 IN1 IN2 IN3 IN4 Ctrl1 Ctrl2 OUT;

- i) <NOMBRE\_SUBCKT> corresponde al nombre de identificación del subcicruito, este nombre es usado por el circuito principal cuando el subcircuito es invocado.
- ii) <nd\_interf1>...<nd\_interfn>; corresponden a los nodos de interfaz que definen los terminales de interconexión del subcircuito con el circuito principal cuando el subcircuito es llamado.
- iii) Aun cuando la llamada a subcircuitos es recursiva, su definición no lo es, es decir que dentro de un subcircuito pueden haber llamadas a otros subcircuitos, pero no pueden haber definiciones de subcircuitos internos.

**c) Declaración de variables locales "LOCAL"**

Sintaxis: LOCAL <nd\_1> <nd\_2> +... <nd\_n>;

Ejemplo: LOCAL x1 x2 x3 Gate1 Gate2;

<nd\_1> <nd\_2> <nd\_n> son los nodos internos (locales) de un subcircuito, de modo que se pueden usar los mismos nombres en los nodos de otros subcircuitos o del circuito principal. Los nodos que no sean declarados como locales se asume que son nodos globales comunes al circuito principal y a todos los subcircuitos definidos.



d) Fin de circuito "END"

Sintaxis: END

e) Fin de subcircuito "END;"

Sintaxis: END;

### B.2.2 Comandos de atribución de valores

a) "INICIE"

Sintaxis: INicie <valor>

Ejemplo: IN 0

Todos los nodos del circuito reciben este valor.

b) "LIGUE"

Sintaxis: LIgüe <nd\_1> <valor> ... <nd\_n> <valor>

Ejemplo: LIGUE Vdd 1 Gnd 0

El nodo (sea o no del tipo INPUT) es conectado a una fuente de tensión con el valor indicado.

c) "ENTRADA"

Sintaxis: ENtrada <nd\_1> <lista\_valor>...<nd\_n> <lista\_valor>

Ejemplo: EN IN1 00001111 IN2 00110011

Se provee una lista de valores que son atribuidos secuencialmente a un nodo que debe ser del tipo INPUT. En cada CICLO de simulación el próximo valor será atribuido.

d) "RELOGIO"

Sintaxis: RElogio <nd\_1> <lista\_valor>...<nd\_n> <lista\_valor>

Ejemplo: RELOGIO CK1 1100 CK2 0110

Se provee una secuencia de valores que, son atribuidos secuencialmente a un nodo que debe ser del tipo INPUT. En cada FASE de simulación el próximo valor será atribuido.

NOTA: El número de FASES de cada CICLO está dado por el número de dígitos del comando RELOGIO.

e) "PRENDA"

Sintaxis: PRenda <nd\_1> <valor> \*\*\* <nd\_n> <valor>

Ejemplo: PRENDA Nint 1

Prenda transforma permanentemente un nodo interno (sea o no del tipo INPUT) en un nodo alimentado por una fuente de señal externa que no puede ser alterada en el proceso de simulación.

f) "SOLTE"

Sintaxis: SOLte <nd\_1> \*\*\*\* <nd\_n>

Ejemplo: SO Nint

Anula la función PRENDA para los nodos especificados.

### B.2.3 Comandos de observación de puntos del circuito

#### a) "MOSTRE"

Sintaxis: MOstre <nd\_1> ..... <nd\_n>

Ejemplo: MOSTRE OUT1 OUT2 OUT3

Los estados de los nodos indicados son mostrados en la pantalla del terminal.

#### b) "OBSERVE"

Sintaxis: OBserve <nd\_1> ..... <nd\_n>

Ejemplo: OB OUT1 OUT2 OUT3

A lo largo del proceso de simulación los estados de los nodos indicados son mostrados automáticamente en forma de una lista asociados al intervalo de tiempo fase/ciclo en que se produjeron.

#### c) "ESQUECA"

Sintaxis: ESqueca <nd\_1> ..... <nd\_n>

Ejemplo: ES OUT2

Anula la función OBSERVE para los nodos indicados.

### B.2.4 Comandos de activación del proceso de simulación

#### a) "FASE"

Sintaxis: FAse <n\_veces>

Ejemplo: FASE 8

Indica el inicio de la simulación que será ejecutada tantas cuantas FASES sean especificadas, los resultados son

exhibidos luego de cada fase.

b) "CICLO"

Sintaxis: CICLO <n\_veces>

Ejemplo: CICLO 8

Indica el inicio de la simulación que será ejecutada tantos cuantos CICLOS sean indicados, el estado de los nodos es exhibido luego de cada ciclo. En este caso no se exhiben los resultados intermedios de la simulación en cada FASE.

B.2.5 Comandos del modo gráfico

a) "GRAFICO"

Sintaxis: GRafico

Este comando permite que los resultados de la simulación sean presentados en la ventana interactiva gráfica del interfaz del simulador descrita en la Fig.5.10. y que se reproduce en la Fig.B.1.

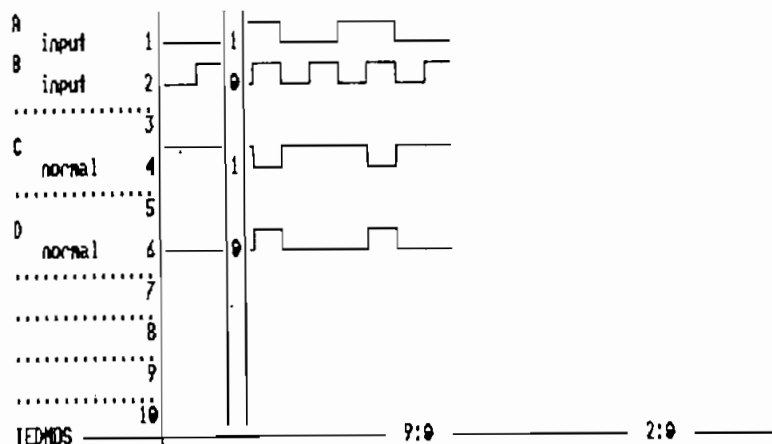


FIGURA B.1 Interfaz gráfico del simulador.

b) "ALFA"

Sintaxis: ALfa

Permite al usuario el retorno al modo alfanumérico de trabajo del simulador.

c) "PLUGUE"

Sintaxis: PLugue <nd> <canal>

Ejemplo: PL B 2

Permite mostrar el nodo indicado en el canal del "osciloscopio" especificado en la ventana del interfaz (Fig. B.1).

d) "DESPLUGUE"

Sintaxis: DEsplugue <canal>

Ejemplo: DE 4

Desactiva la señal adscrita al canal indicado.

e) "TROQUE"

Sintaxis: TROque <canal> <canal>

Ejemplo: TR 4 6

Intercambia las conexiones y las líneas indicativas de las señales involucradas en cada canal.

f) "REGUA"

Sintaxis: REGua

Genera sobre el gráfico la regla móvil vertical del interfaz gráfico (Fig.B.1) en que se indica el estado lógico de cada columna (0, 1 ó X). Junto con la regla aparece en la línea inferior del interfaz (a la derecha del reloj indicador del último ciclo/fase simulado) un nuevo reloj que indica el tiempo de simulación sobre el que se halla posicionada la regla móvil. La regla y su reloj indicativo se retiran con el comando <ENTER>.

g) "REDESENHE"

Sintaxis: REDesenhe

Bajo algunas circunstancias el contenido del interfaz gráfico puede ser alterado por algún mensaje de error, este comando permite rediseñar el contenido del interfaz.

h) "LIMPE"

Sintaxis: LIMpe

Permite retirar todo el contenido lógico del interfaz sin retirar las señales activadas a cada canal.

i) "IMPRIMA"

Sintaxis: IMprima

Permite reproducir el contenido del osciloscopio de la Fig.B.1 en una impresora.

## B.2.6 Comandos generales

### a) "LISTA"

Sintaxis: LISTa

Lista el nombre de todos los nodos del circuito.

### b) "EXECUTE"

Sintaxis: EXecute <arch\_comandos>

Ejemplo: EX comand

Este comando permite ejecutar los comandos contenidos en un archivo <arch\_comandos>.SIM. En este archivo los comandos son descritos de manera igual a como se digitan en el simulador, luego de esta ejecución el simulador retorna a su modo normal.

### c) "EDITE"

Sintaxis: EDite <archivo>

Ejemplo: ED arch\_sim

Activa un editor de textos para editar el archivo indicado, de ser este omitido se crea un archivo *SEMNONE.TXT* o se asume el último archivo editado.

### d) "MENU"

Sintaxis: ? (socorro, menú)

Permite generar en pantalla los comandos del simulador.

e) "FIM"

Sintaxis: FIm

Fin de simulación.



**ANEXO C**

**DIRECTIVAS DE DESCRIPCION CIF**

## C.1 PRIMITIVAS GEOMETRICAS <sup>(1)</sup>

Un formato de archivo simple como el CIF puede incluir solo construcciones geométricas elementales, como rectángulos, círculos, polígonos y alambres (Fig.C.2) de modo que todas las estructuras geométricas existentes en los layouts de CIs deben expresarse a este nivel de descripción.

### C.1.1 Rectángulo "Box"

Sintaxis: B <largo> <ancho> <centro x> <centro y>  
[<dirección x> <dirección y>];

Ejemplo: B 25 60 80 40 -20 20;

Los campos que definen un rectángulo se indican gráficamente en la Fig.C.1 para el que se ha definido el ejemplo anterior;

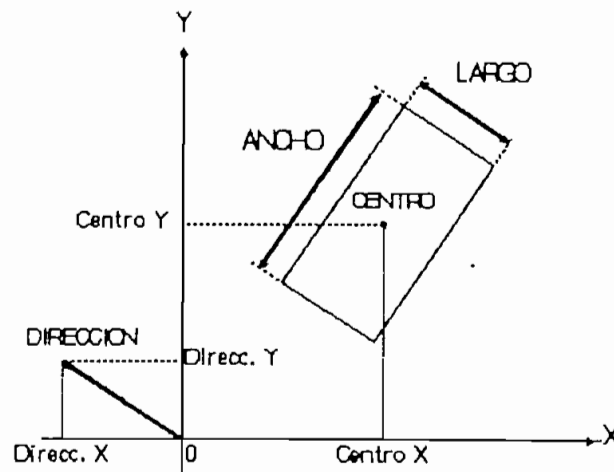


FIGURA C.1 Representación de un rectángulo (box) en el formato CIF.

<sup>(1)</sup> "Introduction to VLSI Systems", C. Mead - L. Conway, págs 117 a 119.

- a) <centro x> <centro y> definen la posición del rectángulo,
- b) <dirección x> <dirección y> definen su orientación,
- c) <largo> y <ancho> indican las dimensiones del rectángulo en dirección paralela y perpendicular, respectivamente, al vector de dirección.
- d) [<dirección x> <dirección y>] pueden o no especificarse, de omitirse, la orientación asumida será (1 0), es decir paralela al eje x.

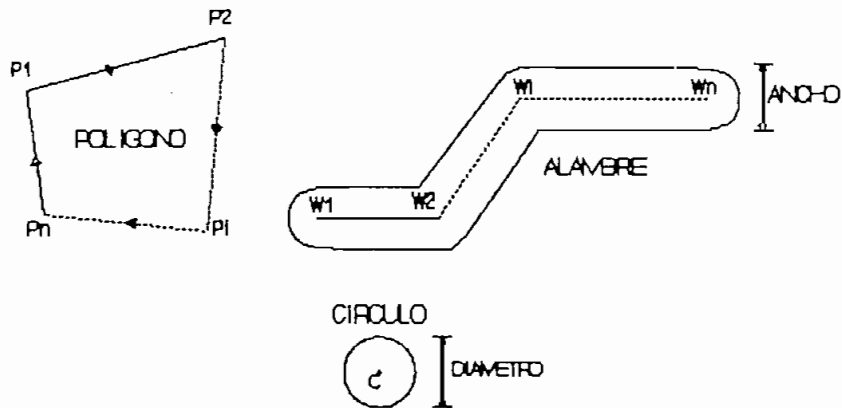


FIGURA C.2 Algunas primitivas geométricas en el formato CIF.

### C.1.2 Polígono "Polygon"

Sintaxis: P <vértice x1> <vértice y1>+<vértice xi>  
 <vértice yi>+<vértice xn> <vértice yn>;

Ejemplo: P 0 0 10 20 -30 40;

Un polígono es una región cerrada determinada por los vértices dados en el camino, en el orden que estos indican:

- a)  $\langle \text{vértice } x_i \rangle \langle \text{vértice } y_i \rangle$  corresponden a las coordenadas  $(x_i y_i)$  de los vértices del polígono,
- b) se asume que la última arista del polígono es aquella que une el punto de coordenadas  $(x_n y_n)$  con el punto inicial de coordenadas  $(x_1 y_1)$ .

#### C.1.3 Círculo "Round"

Sintaxis: R  $\langle \text{diámetro} \rangle \langle \text{centro } x \rangle \langle \text{centro } y \rangle$ ;

Ejemplo: R 200 -500 800;

La forma de el círculo es determinada por su  $\langle \text{diámetro} \rangle$ , y la posición, por las coordenadas  $(x y)$  de su centro.

#### C.1.4 Alambre "Wire"

Sintaxis: W  $\langle \text{ancho} \rangle \langle \text{vértice } x_1 \rangle \langle \text{vértice } y_1 \rangle \dots \langle \text{vértice } x_i \rangle$   
 $\langle \text{vértice } y_i \rangle \dots \langle \text{vértice } x_n \rangle \langle \text{vértice } y_n \rangle$ ;

Ejemplo: W 40 0 0 5 2 10 10 -30 40;

Esta primitiva geométrica describe un camino con  $\langle \text{ancho} \rangle$  uniforme en torno a una línea central cuyos vértices se definen por las coordenadas  $(x_1 y_1) \dots (x_i y_i) \dots (x_n y_n)$  en ese orden.

## C.2 ESPECIFICACION DE CAPA "L"

Sintaxis: L <nombre de la capa>;

Ejemplo: L CNWI;

Cada primitiva geométrica debe ser etiquetada especificando la máscara de fabricación a que pertenece. En lugar de etiquetar el nombre de cada primitiva individualmente, la capa es especificada como un "modo" que se aplica a todas las primitivas subsiguientes, hasta que una nueva capa sea indicada;

- a) <nombre de la capa> es un nombre corto de la capa con forma mnemónica, lo que provee legibilidad al archivo,
- b) las nominaciones de las capas han sido estandarizadas para cada tecnología de fabricación; generalmente, su primer carácter indica la tecnología de fabricación y el resto es un mnemónico de la capa.

Para la tecnología CMOS, las denominaciones de las capas adoptadas son:

DEFINICION DE NIVELES EN CIF (según las reglas ES2 - ECPD15 y ECPD12 -) <sup>(6)</sup>

<u>NIVEL</u>	<u>CIF</u>
Nwell	CNWI
active area	CTOX
poly	CPOL
N + implantation	CNPI
P + implantation	CPPI
contact	CCON
metal1	CME1
via	CVIA
metal2	CME2
passivation	CPAS
label	TLAB
cell boundary	TCLB
text	TEXT

### C.3 SIMBOLOS

Una descripción CIF con miles de primitivas requeriría archivos enormes, además, considerando que varios layouts VLSI incluyen ítems repetitivos es útil definirlos como símbolos. Esta facilidad, complementada con la habilidad de "invocarlos" para insertarlos en una posición epecífica reduce significativamente el tamaño de las descripciones CIF.

#### C.3.1 Definición de símbolo "DS"

Sintaxis: DS <# símbolo> <A> <B>; .....;DF;

Ejemplo: DS 23 100 1; ..... ;DF;

Un símbolo es definido precediendo su descripción geométrica con la indicación *Definition Start* DS, y marcando su final con *Definition Finish* DF.

---

<sup>(6)</sup> "Especificaciones de participación", servicio MPC del CNM, pág. 8.

- a) <# Símbolo> se emplea para asignar un número de identificación al símbolo (no guarda relación con el orden que este ocupa en el archivo de descripción).
- b) El mecanismo de definición de símbolos permite escalar las medidas de distancias. Así, cuando se define un símbolo cada medida de distancia (posición o tamaño) citada en las primitivas geométricas que la definición abarque es escalada de manera que:

$$\textit{Distancia final} = \frac{\textit{Distancia en la definición} * \langle A \rangle}{\langle B \rangle}$$

[C.1]

- Esta opción permite reducir el tamaño del archivo restringiendo el número de dígitos en los enteros que especifican dimensiones.
- c) Las definiciones no pueden ser anidadas, es decir luego de un comando DS, la terminación DF debe hallarse antes del siguiente DS. Las definiciones pueden, no obstante, contener llamadas a otros símbolos, que a su vez pueden llamar a otros símbolos, y así por el estilo.
  - d) Existe tan solo una restricción en el posicionamiento de los símbolos en el archivo: todo símbolo debe hallarse definido antes de que su invocación sea necesaria. Esta restricción puede ser satisfecha colocando las definiciones de símbolos al inicio del archivo, antes de los comandos de llamadas a los símbolos.

### C.3.2 Llamada a símbolo "C"

Sintaxis: C <# símbolo> <transformación>;

Ejemplo: C 23 MX R -1 1 T 10 20;

La directiva C especifica la invocación al símbolo identificado por <# símbolo> que le fuera asignado en su definición (DS <# símbolo> ...).

La <transformación> a ser aplicada en el símbolo es especificada por una lista de transformaciones primitivas.

Las transformaciones primitivas factibles son:

T <xt> <yt>      Traslada el origen referencial de la grilla del símbolo al punto de coordenadas (xt yt).

MX                      Refleja el símbolo en torno al eje x, es decir se multiplica su coordenada y por -1

MY                      Refleja el símbolo en torno al eje y, es decir se multiplica su coordenada x por -1.

R <xr> <yr>      Rota el eje x referencial del símbolo a la dirección dada por (xr,yr).

a) Las llamadas a símbolos pueden ser anidadas, es decir dentro de la definición del símbolo puede haber llamadas a otros.



- b) Las definiciones de capas son resguardadas a través de las llamadas y definiciones. Así, en la secuencia:

```
L CME1;  
B 10 20 400 200;  
C 23 T 10 20 MX MY;  
DS 34 10 1;  
'';  
DF;  
L CME1;  
R 5 5;
```

el segundo "L CME1" es innecesario, a pesar de las especificaciones de símbolos 23 y 34.

- c) El escalamiento en las estructuras DS, afecta a los parámetros de transformación en la llamada al símbolo, pero no a las dimensiones internas del símbolo invocado.

### C.3.3 Eliminación de definición de símbolo "DD"

Sintaxis: DD <número>;

Ejemplo: DD 100;

La señal DD indica al programa que lee el archivo que todos los símbolos con índice mayor o igual que <número> ( $\geq$  <número>) serán "suprimidos", es decir que no podrán ser invocados nuevamente.

Esta opción permite renovar las definiciones de símbolos con el consiguiente ahorro de memoria de procesamiento.

## C.4 TEXTOS DE USUARIO

### C.4.1 Expansión de usuario

Sintaxis: <número> <Texto de usuario>;

Ejemplo: 9 "CELDA INV.CEL";

Esta estructura permite la introducción de directivas (o comandos) a ser implantados por el usuario en sus programas de procesamiento de archivos CIF. Pueden ser usadas para actividades como:

- a) Invocación para la inserción de otras celdas de biblioteca en ese punto de la descripción,
- b) Instrucciones a un pre-procesador que serán ignoradas por otro programa que lea solo las estructuras CIF,
- c) Consignación de información auxiliar o de estructuras de datos.

#### C.4.2 Comentario

Sintaxis: (<Texto de usuario>);

Ejemplo: (Diseño de banda de salida);

Los comentarios facilitan la lectura de un archivo CIF.

#### C.5 DIRECTIVA DE FIN DE ARCHIVO

Sintaxis: E;

La directiva E señala el final del archivo CIF.

**ANEXO D**

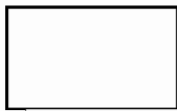
**DETALLES DEL SISTEMA  
TENTOS**

## D.1 LANGUAGE NILOTRANCA

El language NILOTRANCA se basa en el language lógico NILO al que se ha adaptado para la descripción a nivel estructural (lógico) de diseños VLSI mediante la adición de palabras reservadas que se emplean como directivas para posicionamiento de celdas y redes de enrutamiento de señales.

### D.1.1 Simbología empleada

En las descripciones estructurales subsiguientes se ha adoptado la siguiente notación:



La estructura encerrada por este símbolo es de tipo general y es especificada en otra parte de la descripción del language NILOTRANCA.



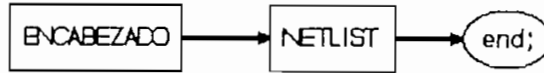
La secuencia de caracteres encerrada por este símbolo es propia del language NILOTRANCA y debe ser empleada sin variación alguna (opcionalmente se pueden cambiar los caracteres en mayúscula por sus correspondientes minúsculas y viceversa).

#### NOTA:

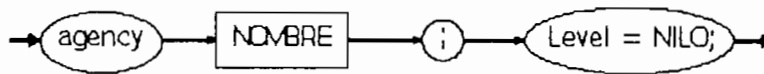
La secuencia \*/...../\* es empleada en el language NILOTRANCA para encerrar comentarios al final de cualquier línea.

### D.1.2 Estructura del archivo NILOTRANCA

El archivo NILOTRANCA tiene la siguiente estructura general:



Donde *end;* indica al compilador el final de la descripción en formato NILOTRANCA. El encabezado indica el inicio de la descripción y su estructura es la siguiente:



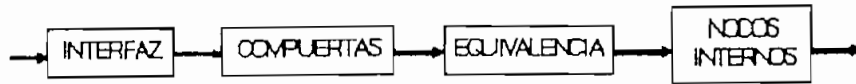
Donde las palabras reservadas encerradas por las elipses son directivas de inicialización para los algoritmos de compilación del circuito descrito.

Nombre es una secuencia cualquiera de caracteres que puede estar formada por letras o dígitos y su extensión puede ser tan larga cuanto el usuario requiera.

### D.1.3 Estructura de la NETLIST

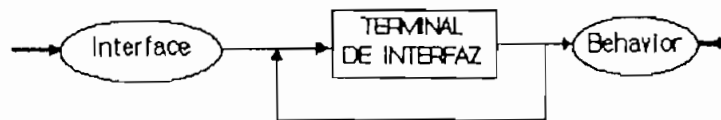
La NETLIST contiene la descripción del circuito al nivel lógico (estructural), esta descripción se basa en celdas

estándar y sus interconexiones o redes, esta descripción se divide en 4 secciones principales:

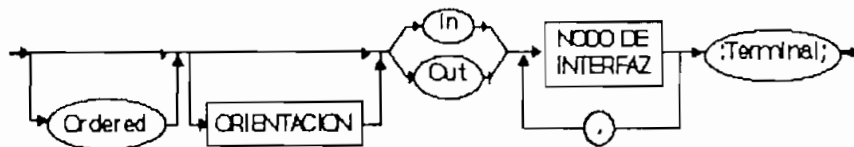


#### D.1.4 Sección de INTERFAZ (Interface)

En esta sección se especifican las señales de interfaz que enlazan el circuito con su exterior. Su estructura es:



##### a) Terminal de interfaz



Las señales de interfaz ingresan o salen del circuito a través de nodos de interfaz (c) que deben catalogarse como nodos de entrada in o nodos de salida out. Estos nodos pueden tener una orientación (b) determinada (Fig.D.1).

Dentro de cada orientación los nodos pueden ordenarse posicionalmente con el comando ordered, de modo que si las orientaciones de las señales son Norte o Sur, estas se or-

denan de izquierda a derecha, y si las orientaciones son Este u Oeste, los nodos se ordenan de arriba hacia abajo (Fig. D.1).

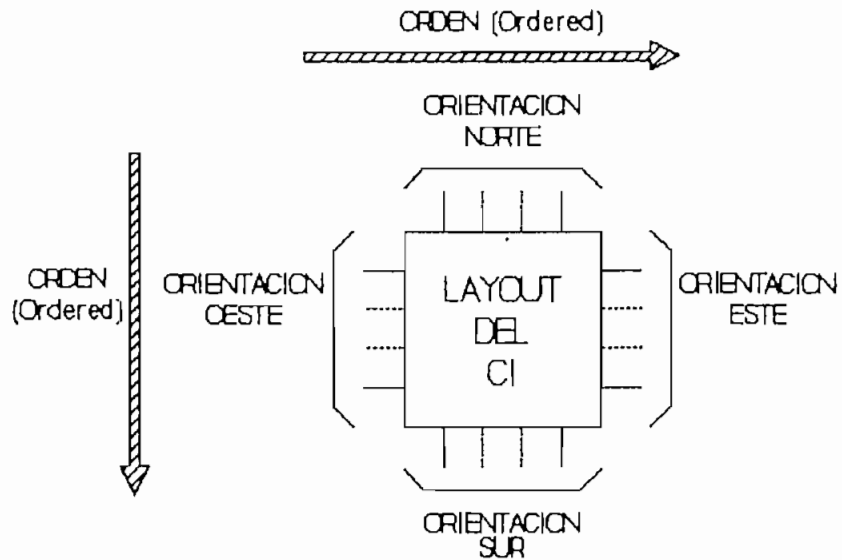
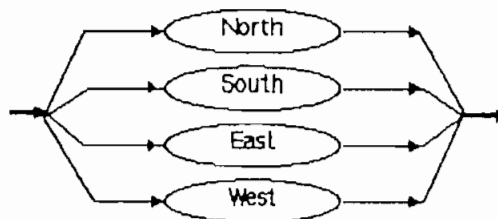


FIGURA D.1 Orientación y ordenamiento de las redes de interfaz.

b) Orientación

Las orientaciones posibles se pueden determinar mediante las palabras reservadas:



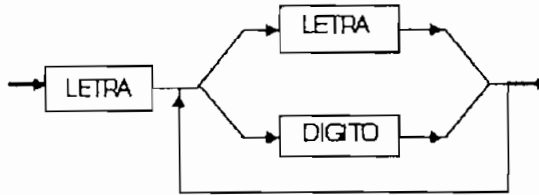
c) Nodo de interfaz

Es un nodo asignado dentro de la red a cada señal de entrada/salida. Su estructura es:





Nombre tiene el formato:



De donde se concluye que todo nombre debe comenzar con una letra. La misma letra sea mayúscula o minúscula tiene idéntico valor sintáctico, y su extensión es de tantos caracteres cuantos el usuario desee.

De la estructura de **terminal de interfaz** (a) se desprende que varios nodos pueden tener las mismas especificaciones de ordenamiento, orientación y catalogación (in/out), estos nodos pueden especificarse por ejemplo como *inA*, *inB*, *inC*, ó como *in[1]*, *in[2]*, *in[3]* que se puede abreviar de la forma *in[1:2:3]*

**d) Ejemplo**

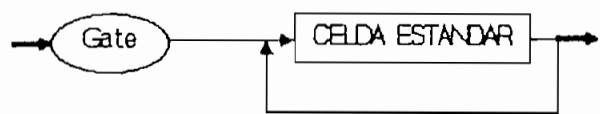
```
interface
  ordered south in ic[3:2]: terminal; /* ic[3], ic[2] */
  ordered south out jps,sts: terminal;
  ordered north in ic[1:0]: terminal; /* ic[1], ic[0] */
      north in jpk ; terminal;
      in jks ; terminal;
```

Lo que indica al sistema que en el proceso de ubicación de celdas se debe dar prioridad a la solución que coloque a las celdas ligadas a los terminales *ic[3]*, *ic[2]*, *jps* y *sts*

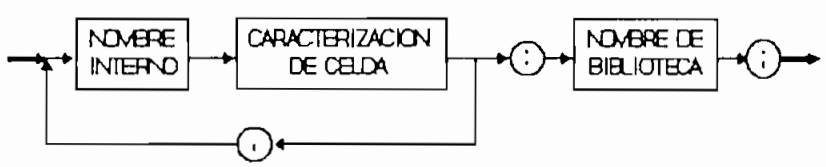
en la última banda del circuito, ordenando los terminales  $ic[3]$  e  $ic[2]$  de derecha a izquierda; de manera análoga con los terminales orientados hacia el norte ( $ic[1]$ ,  $ic[0]$  y  $ipk$ ). La red  $jks$  en cambio no tiene restricciones de orientación ni ordenamiento para su ubicación en el plano de base.

**D.1.5 Sección de COMPUERTAS (Gate)**

En esta sección se definen las celdas estándar a ser usadas en el circuito y opcionalmente se asignan restricciones al posicionamiento de estas dentro de la distribución de bandas. Su estructura es:



**a) Celda estándar**



Corresponde a la estructura en que se nominan y caracterizan las celdas de biblioteca empleadas en el circuito.

**b) Nombre interno**

Es el nombre asignado por el usuario a las celdas de su diseño, cada celda deberá tener un nombre distintivo diferen-

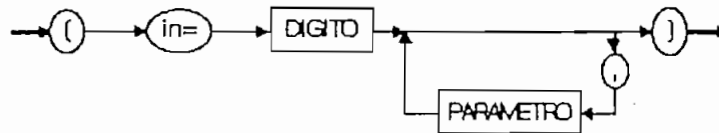
te del de las otras del circuito, aun cuando se trate de la misma celda estándar. Su estructura es:



que fue analizada anteriormente.

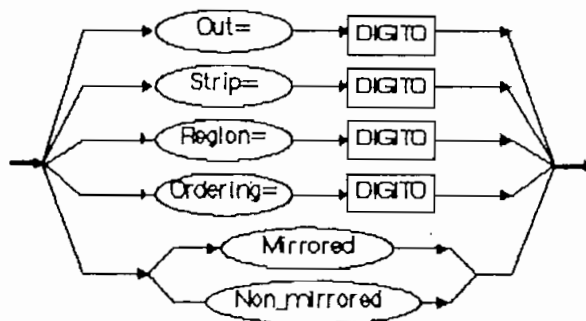
### c) Caracterización de celda

Corresponde a la definición de parámetros que permiten asignar restricciones a las celdas para su ubicación dentro de la estructura de bandas del layout global del circuito, su estructura es:



Donde: *in*=digito, es el número de entradas de la celda.

Los parámetros de caracterización de las celdas son:



Donde las palabras reservadas indican:

<i>out</i>	:Número de salidas
<i>strip</i>	:Banda de particionamiento
<i>region</i>	:Región de particionamiento
<i>ordering</i>	:Orden de posicionamiento
<i>mirrored/non_mirrored</i>	:Reflexión o no en el eje "y"

Las palabras reservadas introducen restricciones (opcionales) que deben ser usadas con mucha cautela ya que limitan la libertad de los algoritmos de ubicación de celdas, ocasionando que generen soluciones que no son óptimas, y en el peor de los casos incompletas.

d) **Nombre de biblioteca**

Es el nombre con el que se identifican las **celdas estándar** en la biblioteca del sistema TENTOS, este nombre no tiene relación con el nombre interno asignado a cada celda por el usuario.

e) **Ejemplo:**

*Gate*

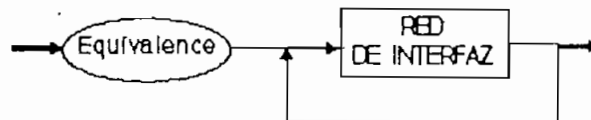
```
A1 (in=2, mirrored, ordering=1, strip=1),  
A2 (in=2, non_mirrored, ordering=4) : NOR;  
A22 (in=3, region=2, strip=2, out=2) : FFD_MS;
```

Esta definición generará una solución que posicionará a la celda A1 (de tipo NOR de dos entradas) reflejada en torno al eje y, en la primera posición (de izquierda a derecha) de la primera banda, la celda A2 (de tipo NOR de dos entradas)

se colocará sin reflexión en la cuarta posición de la banda que le asigne el algoritmo de particionamiento. La celda A22 (del tipo TRANS de tres entradas y dos salidas) se colocará en la segunda región de la segunda banda.

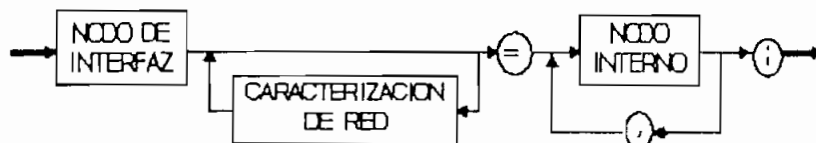
#### D.1.6 Sección de EQUIVALENCIA (Equivalence)

En esta sección se definen las redes de interconexión entre los terminales de entrada/salida indicados en la sección de INTERFAZ y las celdas internas del circuito. Su estructura es:



##### a) Red de interfaz

Define las interconexiones entre los nodos de interfaz del circuito y sus nodos internos.



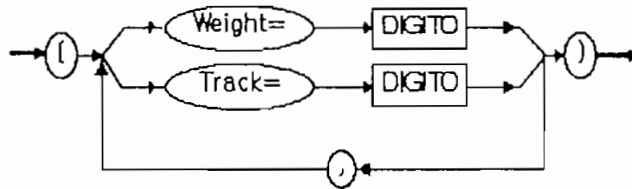
Si el nodo de interfaz es de entrada entonces se enlaza con varios nodos internos del circuito, y si es de salida se enlaza con un solo nodo interno.

b) **Nodo de interfaz**

Corresponde a cada nodo de entrada/salida del circuito, definido anteriormente en la sección INTERFAZ.

c) **Caracterización de red**

En esta sección se indican los parámetros que permiten restringir la distribución de las redes de interconexión durante el proceso de enrutamiento del circuito. Su estructura es:



Donde las palabras reservadas indican:

*weight* : peso diferenciado de la red

*track* : trilla empleada en la interconexión.

i) **Peso diferenciado de red:** es un factor empleado para asignar prioridades que permitan para mantener las celdas que la red involucra en la misma banda o en bandas adyacentes, un peso *weight=0* indica la ausencia de prioridad en el posicionamiento de las celdas de esa red. De no indicarse el peso su valor por omisión es *weight=1*.

ii) **Trillas de interconexión:** En el sistema TENTOS, las bandas están divididas en 11 trillas: VCC, P0, P1,...,P9,

GND, dispuestas en las bandas en el orden que se indica en la Fig. D.2.

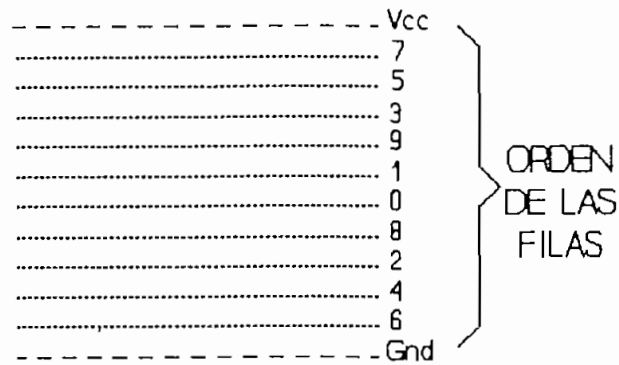
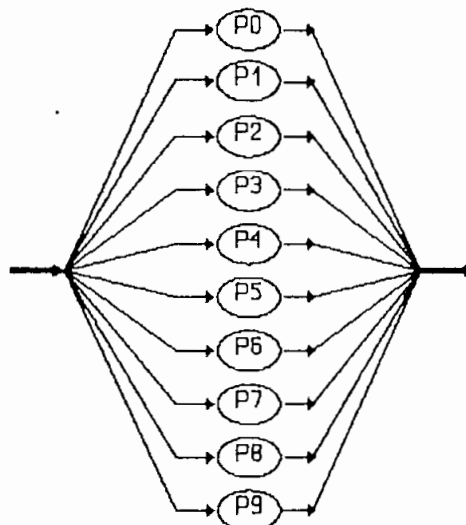


FIGURA D.2 Disposición de las Filas (Trillas) dentro de las bandas en el sistema TENTOS.

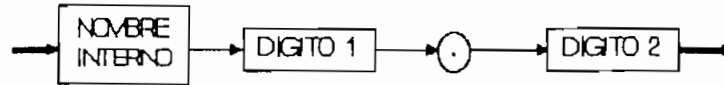
Mediante el parámetro *track* el lenguaje NILOTRANCA permite definir los caminos preferenciales para el enrutamiento de una red.

Las trillas que pueden ser empleadas para el enrutamiento de redes son:



d) Nodo interno

Los nodos internos son los nodos de las celdas estándar de la red que se enlazan directamente con los nodos de interfaz del circuito. Su estructura sintáctica es:



Donde:

Nombre interno es el nombre asignado en la sección COMPUERTAS a la celda estándar involucrada en la red.

Digito 1 = [1...ne]

Digito 2 = [ne+1...ns]

ne = número de nodos de entrada de la celda estándar

ns = número de nodos de salida de la celda estándar

Para las celdas A1 (entradas:2, salidas:1) y A22 (entradas:3, salidas:2) definidas en el ejemplo anterior, los nodos internos serían como se indica en la Tabla D.1:

Nombre interno	Entradas [1...ne]	Salidas [ne+1...ne+ns]	Utilización
A1	1,2	3	A1.1 A1.2 A1.3
A22	1,2,3	4,5	A22.1 A22.2 A22.3 A22.4 A22.5

TABLA D.1 Nominación de los nodos en la netlist NILO-TRANCA.



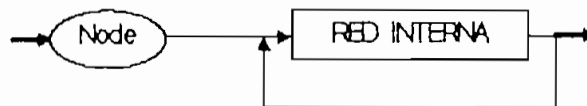
e) Ejemplo

```
equivalence
  ic[0] (Track=P1) = a1.2, a2.1; /* interfaz de entrada */
  jps   (Weight=2) = a22.4;     /* interfaz de salida */
```

En este ejemplo, el terminal de entrada *ic[0]* definido en la sección de INTERFAZ se enlaza con el nodo (2) de entrada a la celda *a1*, (*a1.2*) y con el nodo (1) de entrada a la celda *a2* (*a2.1*), esta red deberá ubicarse en la trilla *P1* de su banda correspondiente. El terminal de salida *jps* se enlaza con el nodo (4) de la celda *a22* (*a22.4*), para esta red se ha definido un peso de dos (2) que indica que la interconexión de estos nodos se debe realizar prioritariamente en una sola banda.

D.1.7 Sección de NODOS INTERNOS (Node)

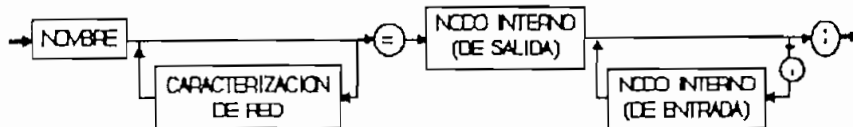
En esta sección se definen las redes internas del circuito su estructura es:



a) Red interna

Las redes interna y de interfaz son eléctrica y físicamente (a nivel de layout) iguales, su única diferencia es a

nivel sintáctico y radica en que no enlaza a los nodos de interfaz del circuito que ya fueron interconectados a las celdas internas en la sección de EQUIVALENCIA. Su estructura es:



b) Nombre

Es una secuencia alfanumérica (definida anteriormente) empleada para nominar a cada red interna del circuito

c) Caracterización de red

Fuesto que la red interna es físicamente igual a la red de interfaz, la estructura y parámetros de caracterización de red son idénticos y ya fueron definidos en la sección de EQUIVALENCIA.

d) Nodo interno (de salida)

La estructura sintáctica de los nodos internos en la NETLIST ya fue indicada en la sección de EQUIVALENCIA.

Del diagrama estructural de la red interna se desprende

que a cada red debe necesariamente asociarse únicamente un nodo interno de salida que es el primero en especificarse.

e) **Nodo interno (de entrada)**

Su estructura sintáctica es similar a la de todos los nodos internos en la NETLIST.

Del diagrama estructural de la red interna se concluye que pueden haber varios nodos de salida e incluso estos nodos pueden no existir.

f) **Ejemplo**

```
node
  n1  (Weight=0)      = a22.4, a1.1;
  n2  (Track=P3)     = a1.3, a22.1, a22.2;
```

La red *n1* enlaza los nodos *a22.4* (de salida) y *a1.1* (de entrada), existiendo para esta red un peso diferenciado de cero (*weight=0*). La red *n2* enlaza los nodos *a1.3* (de salida) y *a22.2*, *a22.1* (de entrada), será ubicada en la trilla *P3* de su banda correspondiente y se asume un peso diferenciado por omisión: *weight=1* para esta red.

## D.2 PARAMETROS DE PARTICIONAMIENTO<sup>(1)</sup>

Los parámetros de particionamiento para los procesos de síntesis TRAMO (Tranca Automatic Module Generator) y TRAGO (Tranca Automatic Generator), son los siguientes:

### D.2.1 Criterio de selección

El particionador genera varias soluciones al problema de distribución de celdas, este parámetro indica el criterio prioritario para la selección final de la respuesta, las opciones posibles son:

- a) **Solución individual**, cuando se asume como solución final, la primera que se genere.
- b) **Area**, cuando de entre las soluciones se escoge aquella que ocupa menor área.
- c) **Interfaz**, cuando se prefiere aquella solución que atiende al mayor porcentaje de restricciones de interfaz.

### D.2.2 Número de regiones

Define el número de regiones en que se divide cada banda a fin de evitar la saturación de los canales de enrutamiento. El número de regiones puede especificarse explícitamente o dejar que el programa lo evalúe mediante la opción *calculado*.

---

<sup>(1)</sup> 'Manual do Usuário do Projeto Tranca V1.0', F. Behn Moraes - M. Lubaszewski - R. Reis. Cap 3.2.2.

### D.2.3 Relación de aspecto

Determina el aspecto del circuito en términos de la relación largo/alto del layout global.

### D.2.4 Criterio de Balanceo (en %)

Define la variación máxima del tamaño de una banda en relación al promedio, en términos de porcentaje con respecto a la mayor celda del circuito.

Por ejemplo: si el largo total de las bandas es 1000, se tienen 5 bandas, la mayor celda del circuito mide 50, y el % de balanceo es 60% entonces los largos de las bandas generadas pueden variar entre  $(1000/5 - 0.6 \cdot 50)$  y  $(1000/5 + 0.6 \cdot 50)$  esto es entre 170 y 230.

### D.2.5 Factor de forma

Es un factor que reduce proporcionalmente el largo de las bandas centrales aumentando el de las bandas extremas, con el objetivo de evitar una sobrecarga del enrutamiento en las bandas centrales del circuito. Se define como:  $(\text{tamaño de la banda central}) / (\text{tamaño promedio de las bandas})$ .

Por ejemplo, si el factor de forma es 0.8 en un circuito de 5 bandas con largo promedio de 100, los largos de las bandas serán aproximadamente 120, 100, 80 100 y 120.

#### D.2.6 Crecimiento del área

Considerando que las bandas del circuito experimenta un alargamiento debido a la apertura de espacios para intercalar las celdas de interconexión, el crecimiento de área define el valor en que es aumentado el largo de cada celda a fin de lograr una mejor aproximación al largo real de las bandas.

#### D.2.7 Número de bandas

Permite al diseñador indicar el número de bandas en que se distribuyen las celdas, o puede dejarse que el programa lo evalúe en la opción *calculado*.

La fórmula para el calculo del número de bandas es:

$$Nb = \sqrt{\frac{W \cdot (1 + K2)}{K1 \cdot RA}}$$

[D.1]

Donde:

W = suma de los largos de las celdas del circuito.

K1= Altura de las celdas de circuito.

K2= Factor de crecimiento de área.

RA= Relación de aspecto

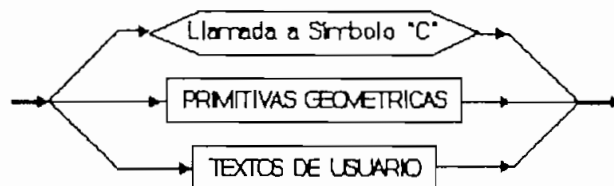
### D.3 FORMATO RS PARA DESCRIPCION GEOMETRICA DE LAYOUTS

Los programas del sistema TENTOS para la descripción de los circuitos al nivel físico, emplean el formato RS, similar al formato de CIF.

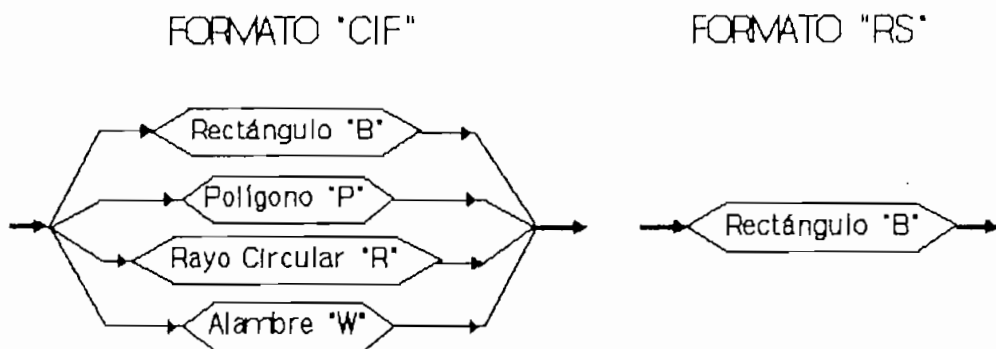
#### D.3.1 Estructura del archivo RS

La descripción en formato RS es similar a la descripción en formato CIF estudiada en detalle en el *Capítulo 5.4*. En el aspecto estructural, la única diferencia es la siguiente:

La estructura de la descripción geométrica en ambos casos es la siguiente:



Sin embargo, las primitivas geométricas difieren en su estructura en que el formato CIF permite cuatro tipos de primitivas geométricas, en cambio el formato RS tan sólo permite una, es decir:



### D.3.2 Sistema de coordenadas para las descripciones RS

El formato de RS usa el mismo sistema referencial de coordenadas cartesianas del formato CIF (Fig. D.3) Las direcciones y distancias son interpretadas siempre en términos de la vista frontal del CI terminado.

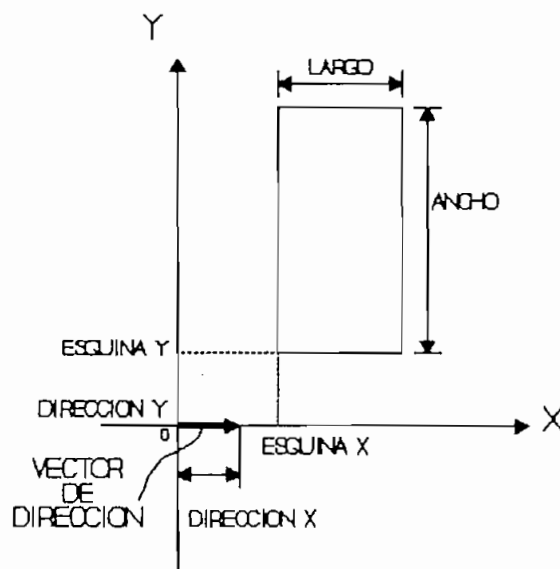


FIGURA D.3 Representación de un rectángulo en el formato RS.



a) Unidades de medida

En el formato RS las unidades de medida de distancias son unidades de micras  $\mu m$  (en el formato CIF son centésimas de micra).

b) Direcciones

Las rotaciones de las primitivas geométricas, al igual que en el formato CIF, también se expresan mediante vectores de dirección, en este caso las coordenadas del vector de dirección  $\langle \text{dirección } x \rangle \langle \text{dirección } y \rangle$  únicamente podrán valer  $(x,0)$  ó  $(0,y)$  puesto que en el formato RS únicamente se admiten rotaciones ortogonales (múltiplos de  $90^\circ$ ).

### D.3.3 Directivas para la descripción RS

En el análisis subsiguiente únicamente se enfatiza en las estructuras sintácticas que difieren del formato CIF, las restantes se asume que son iguales a sus correspondientes en este formato.

a) Primitivas geométricas

i) Rectángulo "Box"

Sintaxis: B  $\langle \text{largo} \rangle \langle \text{ancho} \rangle \langle \text{esquina } x \rangle \langle \text{esquina } y \rangle$   
[ $\langle \text{dirección } x \rangle \langle \text{dirección } y \rangle$ ];

Ejemplo: B 20 40 10 20 0 1;

Los campos que definen al un rectángulo se indican

gráficamente en la Fig. A.2 para el que se ha definido el ejemplo anterior.

<Esquina x> <esquina y> corresponden a las coordenadas de la esquina inferior izquierda del rectángulo con respecto al origen del sistema cartesiano referencial.

#### b) Especificación de capa "L"

Sintaxis: L<nombre de la capa>;

Ejemplo: LMP;

Los mnemónicos para <nombre de la capa> para las tecnologías CMOS y ECPD, adoptadas en el formato RS, y su correspondencia con las capas en formato CIF son:

<u>NIVEL</u>	<u>CIF</u>	<u>RS</u>
Nwell	CNWI	S
Area activa	CTOX	A
Polysilicon	CPOL	F
Implantación N+	CNPI	D
Implantación P+	CPPI	B
Contacto	CCON	C
Metal 1	CME1	M
Vía	CVIA	V
Metal 2	CME2	H
Pasivación	CPAS	G
Contorno de celda	TCLB	O

#### c) Símbolos

Con la misma filosofía del formato CIF, el formato RS permite la descripción de layouts jerárquicos en base a símbolos, lo que da flexibilidad al diseño y permite reducir el tamaño de los archivos de descripción de los layouts.

i) Definición de símbolos "DS"

Sintaxis: DS <# símbolo> <A> <B>; .....;DF;

Ejemplo: DS 52 1 10; ..... ;DF;

ii) Llamadas a símbolos "C"

Sintaxis: C <# símbolo> <Transformación>;

Ejemplo: C 23 MX R -1 1 T 10 20;

Las transformaciones primitivas aplicables a los símbolos son las mismas del formato CIF con la limitación, indicada anteriormente, de que únicamente se admiten rotaciones ortogonales de los símbolos invocados.

iii) Eliminación de definición de símbolo "DD"

Sintaxis: DD <número>;

Ejemplo: DD 100;

d) Textos de usuario

i) Expansión de usuario

Sintaxis: <número> <Texto de usuario>;

Ejemplo: 9 "CELDA INV.CEL";

ii) Comentario

Sintaxis: (<Texto de usuario>);

Ejemplo: (Diseño de banda de salida);

e) Directiva de fin de archivo

Sintaxis: E;

**ANEXO E**

**ARCHIVOS AUXILIARES DEL SISTEMA  
TENTOS**

## E.1 ARQUIVO DE CONFIGURACIÓN DEL SISTEMA: TENTOS.CFG

\*\*\*\* ARQUIVO DE CONFIGURACAO PARA O TRANCA / TRAMO / TRAGO \*\*\*\*

\*\*\*\* MORAES - 07/MARCO/1991

ARAUJO - 30/MARCO/91 ( Este arquivo sofreu alteracoes == USE FC == )

\*\*\*\* OBSERVACAO: Apenas as palavras que comecam por \$ sao palavras reservadas, desta forma e' possivel termos tantas regras de desenho para o trago e tramo quanto quisermos, bastando ativar as regras com a palavra RESERVADA '\$regras'.

### DIRETORIOS DO USUARIO.

\$dir_entrada	\TENTOS
\$dir_saida	\TENTOS
\$dir_tmp	\TENTOS

### DIRETORIOS QUE DEVEM SER FIXOS:

\$dir_lib	\gme\lib
\$dir_tec	\GME\TEC
\$dir_aramos	\GME\ARAMOS
\$dir_pac	\GME\PAC
\$dir_esqueleto	\GME\ESQUELETO
\$dir_tranca	\GME\TRANCA
\$dir_spice	\GME\SPICE
\$dir_texto	\ADM\TURBOC

NOME DO ARQUIVO DE REGRAS QUE DESCREVEM A TECNOLOGIA - no diretorio dir\_tec

\$arquivo_regras	CMOS15
------------------	--------

NOME DO ARQUIVO QUE DESCREVE A BIBLIOTECA - no diretorio dir\_lib

\$gerente_celula	CEL.LIB
------------------	---------

EXTENSAO DAS CELULAS DA BIBLIOTECA

\$sufixo_celula	CEL
-----------------	-----

EXTENSAO DOS ARQUIVOS SPICE

\$sufixo_spice	SIM
----------------	-----

FORMATO DA SAIDA DO EXTRIBO (EX7) valido: spice (default) ou aramos  
\$out\_extribo aramos

INDICA A NECESSIDADE OU NAO DE APERTAR UMA TECLA NO FIM DOS PROGRAMAS  
\$wait 0

\*\*\*\*\* REGRAS DE DESENHO PARA O TRAGO - 1.5u

```
regras
lambda 0.2    o lambda deve vir antes das regras
LA  2.4
DAA 3.2
DAI 2.0
DIW 2.8
DDB 4.0
DSG 9.6
LP  1.6
DFF 2.4
DAF 1.0
STF 1.6
LC  2.0
LM  2.4
DMC 1.0
DMM 2.4
LV  2.0
LH  2.4
DHV 1.0
DHH 2.4
$endregras
```

\*\*\*\*\* PARAMETROS PARA O TRAGO

```
$TEC N          tipo dos pocos: N, P, NP
$out rs        saida: RS ou CIF
$W_ALIMENTA 10.0 largura da alimentacao
$POSIC 4       grau para conectividade no placement
$OFFSET 0      distancia entre vcc,gnd e as difusoes
$REPETIR 8     numero de vezes para realizar o roteamento
$MIN_HORIZ 6   fator de justaposicao
$GND_ESQ 1     se 1 o terra e a esquerda
```

\*\*\*\*\* NOME DAS CAMADAS DE DESENHO, EM RS E CIF

```
$camadas 12
poli      LMP CFOL
metall1   LMM CME1
metall2   LMH CME2
difn      LMD CNFI
difp      LMR CFPI
pocon     LMS CNWI
pocop     LMG CFWI
cont      LMC CCON
```

```
via      LMV CVIA
impn     LMD CNFI
impp     LME CPPI
ativa    LMA CTOX
```

\*\*\*\*\* DEFINICOES UTILIZADAS PELO ROTEADOR \*\*\*\*\*

==> File RO\_LAY.C & RO\_SAIDA.C:

```
$regras_ro
YVCC  97  /* coordenada y da linha de VCC */
LALIM  6  /* largura da trilha de VCC */
LTRI   4  /* largura de trilha */
YGND   3  /* coordenada y da linha de GND */
LPOLI  2  /* largura de nodo */
HFOLI  94  /* altura de nodo */
YNOD  12  /* coordenada y de nodo */
XENT   2  /* deslocamento com relacao a pos real do nodo */
DH     12  /* delta de altura em caso de cel. int. EXT */
DMM    3
CC      6
LCH     2
LCV     2
LMH     6
LMV     6
LPH     6
LPV     6
DYC     2
LPCI   6
DFFI   3
HFQC   46
OFB    60  /* coord. inferior do pogo P */
OFD     0  /* coord. inferior do pogo N */
LNOD   4  /* largura de nodo */
HNOD  82  /* altura de nodo */
OFFSET 1  /* deslocamento com relacao a pos real do nodo */
FRONT  3  /* distancia do contato a fronteira de uma
           celula de interconexao */

$endregras
```

## E.2 ARQUIVO DE CONFIGURACIÓN DE EDITOR ESQUEMÁTICO:

*EDITOR.CFG.*

```
# ARQUIVO_DE_CONFIGURACAO #
*
* -----
* Projeto SID-MICROELETRONICA / UFRGS-CF'GCC
* -----

* Arquivos dos drivers de dispositivos
  $MOUSE=nao
  $VIDEO="\gme\esquelet\HERC"
  $IMPR ="\gme\esquelet\PRINTERD"

* Arquivo do fonte alternativo de definicao de caracteres de texto
1  $TXTFONT="\gme\esquelet\SANS"

* Tamanho da folha de impressao (Em decimos de milimetro)
  $IMPFOLHA = 2700,2000

* Sensibilidade de movimentacao do mouse (Mickey)
  $MOUSENS = 8,8

* Tempo de espera para as operacoes de edicao
  $DELAY = 100

* Tamanho da folha utilizado ( A0 = 0 ate' A8 = 8)
* >> No EDSMB e' sempre igual a A0
  $FOLHA=0

*
* Inicializacao das cores -> EGA
*
* 0 = Preto      4 = Vermelho    8 = Violeta      12 = Vermelho Escuro
* 1 = Branco     5 = Amarelo    9 = Laranja     13 = Azul Claro
* 2 = Azul       6 = Cinza      10 = Azul Escuro 14 = Verde Claro
* 3 = Verde      7 = Rosa       11 = Verde Escuro 15 = Vermelho Claro
*

  $CORES = 0,63,9,30,36,62,7,37,5,46,1,42,4,27,19,39

* Selecao de cores dos menus : Titulo, Borda, Fundo, Texto, Barra
  $CORMENU1 = 1,13,2,1,1
  $CORMENU2 = 1,13,2,1,1
  $CORMENU3 = 8,10,1,0,3

* Selecao de cores do status : Borda, Fundo, Caracter
  $CORSTAT = 1,0,6

* Selecao de cores do help : Borda, Fundo, Caracter
  $CORHELP = 10,2,1
```



```

* Selecao de cores do directorio : Borda, Fundo, Caracter
  $CORDIR = 10,2,1

* Selecao de cores do directorio : Borda, Fundo, Caracter
  $CORHIERARQ = 10,2,1

*
* Selecao de cores das primitivas (Editor de Esquemas)
*
* Ordem das cores : Conexao, Bloco Funcional, Pino de Esquema, Derivador,
*                   Linha Comentario, Simbolo e Destaque
*

  $CORPRIM = 1,7,5,12,1,1,1

* Espacamento entre os itens do menu
  $ESPITEM = 6
  $ESPTOPO = 20

* Tamanho do Campo de Gravidade
  $LIMGRAV = 25

* Selecao das bibliotecas
  $BIRUSU="\gme\lib"
  $BIBSIS="\gme\lib"

* Selecao das bibliotecas de simbolos (Editor de Esquemas)
  $PATHBIB = "\gme\lib"

* Selecao do Directorio de Trabalho (Editor de Esquemas)
  $DIRTRAB = "\projecto"

* Drivers de video disponiveis
  #VIDEO : CGA
           EGA
           VGA
           HERC

* Drivers de impresora disponiveis
  #IMPR : SEM
         PRINTERS
         PRINTERD
         PRINTERQ

```

### E.3 ARCHIVO DE TECNOLOGÍA: CMOS15.TEC

```
*-----*
* ARCHIVO DE TECNOLOGIA CMOS15.TEC *
* ULTIMA MODIFICACION: ABRIL 1992 *
* FREDY LEMUS C./IVAN BERNAL C. *
* TECNOLOGIA: ES2 DUAL LAYER METAL CMOS 1.5µm*
* (ECPD15/1 REV.A) *
*-----*
```

CMOS

15

```
*-----*
* MOLDES DE EDICION GRAFICA E IMPRESA *
*-----*
```

```
* FORMATO: <CAPA DE DISEÑO> <COLOR DEL CONTORNO>
* <patrón de edición de puntos ***
* *** matriz de 8x8 ***
* *** 0:punto apagado ***
* *** 1:punto encendido ***>
```

D 0

00000000

01000100

00000000

00010000

00000000

01000100

00000000

00000000

S 7

00000000

10000000

00000000

00000000

00000000

00001000

00000000

00000000

D 6

10000001

00000000

00000000

00011000

00011000

00000000

00000000

10000001

E 5

10000000

10000000

00000000

00001000

00001000

00001000

00000000

10000000

A 2  
00001000  
00000100  
00000010  
00000001  
10000000  
01000000  
00100000  
00010000  
F 4  
00100010  
01000100  
10001000  
00010001  
00100010  
01000100  
10001000  
00010001  
M 1  
00100100  
01000010  
01000010  
00100100  
00100100  
01000010  
01000010  
00100100  
C 7  
11111111  
10011001  
10011001  
11111111  
11111111  
10011001  
10011001  
11111111  
H 3  
11110000  
00001111  
11110000  
00001111  
11110000  
00001111  
11110000  
00001111  
V 5  
11111111  
11111111  
11111111  
11111111  
11111111  
11111111  
11111111  
11111111

G 7  
00000000  
00000000  
00010000  
00111000  
00010000  
00000000  
00000000  
00000000

E 7  
10000001  
01000010  
00100100  
00011000  
00011000  
00100100  
01000010  
10000001

\*  
\* D = Contorno (Box)  
\* S = POCO N  
\* D = Implante N+  
\* B = Implante P+  
\* A = Area Ativa  
\* P = Polissilicio  
\* M = Metal 1  
\* C = Contato  
\* H = Metal 2  
\* V = Via  
\* G = Depassivacao  
\* E = Erro DRC

\*  
\*LAMBDA 0.2

\*  
\*-----\*  
\* NOMES DOS NIVEIS CORRESPONDENTES EM CIF PARA O CONVERSOR \*  
\*-----\*

CIF O TCLB  
CIF S CNWI  
CIF D CNPI  
CIF B CPPI  
CIF A CTOX  
CIF P CPOL  
CIF M CME1  
CIF C CCOM  
CIF H CME2  
CIF V CVIA  
CIF G CPAS  
\*

ESCALA 1 10  
LAMBDA 0.1

```

*-----*
* DESCRICAO DA TECNOLOGIA PARA O EXTRATOR *
*-----*
*
* Regras de conectividade
CON A C
CON P C
CON M C
CON M V
CON H V

* Regras de formacao dos transistores:
* MOSFET <nome> <gate> <drain> <substr> <eff. len> <eff. widt>
MOSFET NMOS P A D 0.3 0.2
MOSFET PMOS P A B 0.5 0.1

* Resistencia minima a ser considerada (Ohm)
RTH 1000

* Resistencia laminar e resistencia de contato (Ohm)
RES A 65.0 75.0
RES P 25.0 50.0
RES M 0.06 0.0
RES H 0.04 0.2

* Capacitancia minima a ser considerada (Femto Farad)
XCMIN 100

* Capacitancia em uF/m2 de cada nivel
XCAP F 49.0
XCAP M 25.0
XCAP H 13.0
XCAP & 1380.0
FIM
*{*

* Modelos dos transistores
*
.MODEL NMOS NMOS LEVEL=2 LD=0.325U TOX=250E-10 NSUB=2E16 VTO=0.7
+UD=510 UEXP=0.22 UCRIT=24.3K DELTA=0.4 XJ=0.4U VMAX=54K NEFF=4
+RSH=55 NFS=0 JS=2U CJ=130U CJSW=620P MJ=0.53 MJSW=0.53 PR=0.68V
+CGDO=320P CGSO=320P
*
.MODEL PMOS PMOS LEVEL=2 LD=0.3U TOX=250E-10 NSUB=5E16 VTO=-1.1
+UD=210 UEXP=0.33 UCRIT=51K DELTA=0.4 XJ=0.5U VMAX=47K NEFF=0.88
+RSH=75 NFS=0 JS=10U CJ=490U CJSW=590P MJ=0.46 MJSW=0.46 PR=0.78V
+CGDO=320P CGSO=320P
* } { .END }

```

#### E.4 ARQUIVO DE VERIFICAÇÃO DE REGLAS: CMOS15.DRC

```
( ----- )  
( )  
( CPGCC/UFRGS )  
( DARC3 )  
( ARQUIVO DE REGRAS - CMOS 1.5 / ECPD15/1 REV. A )  
( AUTORES: LAERTE DAVI CLETO )  
( MARCOS KORDYAS DOSSA )  
( )  
( ULTIMA ATUALIZACAO: 22/03/91 )  
( )  
( ----- )
```

( DEFINICAO DE CAMADAS INTERMEDIARIAS )

```
C t a p; ( transistores )  
C x a d; ( interseccao de area ativa e implante n )  
C y a b; ( interseccao de area ativa e implante p )  
C r t x; ( transistor n )  
C z t y; ( transistor p )  
C f c t; ( contato sobre transistor )  
C j c v; ( contato sobre via )  
C k b d; ( coincidencia de implantes p e n )  
C i x s; ( interseccao de area ativa n com poco )  
C q y s; ( interseccao de area ativa p com poco )
```

( DEFINICAO DAS DISTANCIAS MAXIMAS DE AVALIACAO POR CAMADA )

```
D 8.0;  
M s 8.0;  
M a 4.8;  
M x 4.8;  
M y 4.8;  
M p 2.4;  
M d 2.4;  
M b 2.4;  
M c 2.0;  
M m 2.4;  
M v 2.2;  
M h 2.4;  
M t 2.4;  
M r 2.4;  
M z 2.4;  
M i 4.8;  
M q 4.8;
```

```
T 101 - dimensao do poco < 8;  
R 0 s 8.0;
```

```
T 101 - interseccao de pocos < 8;  
R 4 s s 8.0;
```

T 102 - distancia entre pocos  $\leq 8$ ;  
R 5 s s 8.0 0;

T 201 - dimensao da area ativa  $\leq 1.6$ ;  
R 0 a 1.6;

T 201 - interseccao entre areas ativas  $\leq 1.6$ ;  
R 4 a a 1.6;

T 201,505,506 - transistor mal formado;  
R 1 a p 2.0 1.6 2.4 1.6;

T 202 - distancia entre areas ativas  $\leq 3.2$ ;  
R 5 a a 3.2 0;

T 203 - extensao do poco alem da area p  $\leq 4.8$ ;  
R 2 s q 4.8;

T 204 - poco nao envolve area n;  
R 1 s x 3000 3000 3000 3000;  
R 3 x s 3000 3000 3000 3000 3000;  
R 3 i x 3000 3000 0 0 0;  
R 4 s x 3000;

T 205 - distancia entre poco e area n  $\leq 4.8$ ;  
R 5 x s 4.8 4.8;

T 207 - distancia entre poco e area p  $\leq 4.8$ ;  
R 5 y s 4.8 4.8;

T 501 - dimensao do polissilicio  $\leq 1.6$ ;  
R 0 p 1.6;

T 501 - interseccao poli/poli  $\leq 1.6$ ;  
R 4 p p 1.6;

T 502,503 - dimensao do transistor  $\leq 1.6$ ;  
R 0 t 1.6;

T 502,503 - interseccao entre transistores  $\leq 1.6$ ;  
R 4 t t 1.6;

T 504 - distancia poli/poli  $\leq 2.4$ ;  
R 5 p p 2.4 0;

T 507 - distancia entre poli e area ativa  $\leq 0.8$ ;  
R 5 p a 0.8 0.8;

T 601 - dimensao do implante n  $\leq 2.4$ ;  
R 0 d 2.4;

T 601 - interseccao entre implantes n  $\leq 2.4$ ;  
R 4 d d 2.4;

T 602 - distancia entre implantes n < 2.4;  
R 5 d d 2.4 0;

T 651 - dimensao do implante p < 2.4;  
R 0 b 2.4;

T 651 - interseccao entre implantes p < 2.4;  
R 4 b b 2.4;

T 652 - distancia entre implantes p < 2.4;  
R 5 b b 2.4 0;

T 660 - coincidencia entre implantes p e n;  
R 0 k 3000;

T 701 - dimensao do contato  $\diamond$  2;  
R 0 c -2.0;

T 702 - distancia entre contatos < 2;  
R 5 c c 2.0 2.0;

T 714 - superposicao de contatos;  
R 3 c c 3000 0 0 0 0;  
R 4 c c 3000;

T 703 - extensao da area ativa alem do contato < 1;  
R 2 a c 1.0;

T 705 - extensao do poli alem do contato < 1;  
R 2 p c 1.0;

T 713 - contato sobre transistor;  
R 0 f 3000;

T 707 - distancia entre contato e transistor < 1.4;  
R 5 t c 1.4 1.4;

T 801 - dimensao do metal < 2.4;  
R 0 m 2.4;

T 801 - interseccao metal/metal < 2.4;  
R 4 m m 2.4;

T 802 - distancia metal/metal < 2.4;  
R 5 m m 2.4 0;

T 804 - extensao do metal alem do contato < 1;  
R 2 m c 1.0;

T 752 - via sobre contato;  
R 0 j 3000;

T 753 - dimensao da via  $\diamond$  2;  
R 0 v -2.0;



T 754 - distancia entre vias < 2;  
 R 5 v v 2.0 2.0;

T 754 - sobreposicao de vias;  
 R 3 v v 3000 0 0 0 0;  
 R 4 v v 3000;

T 757 - distancia entre via e poli < 2.2;  
 R 5 v p 2.2 2.2;

T 757,758 - via sobre borda de poli;  
 R 1 v p 0 3000 3000 0;  
 R 3 p v 3000 0 0 3000 0;  
 R 4 v p 3000;

T 758 - extensao do poli alem da via < 2.2;  
 R 2 p v 2.2;

T 759 - distancia entre via e contato < 2;  
 R 5 v c 2.0 2.0;

T 760 - extensao de metal alem via < 1;  
 R 2 m v 1.0;

T 851 - dimensao do metal2 < 2.4;  
 R 0 h 2.4;

T 851 - interseccao metal2/metal2 < 2.4;  
 R 4 h h 2.4;

T 852 - distancia metal2/metal2 < 2.4;  
 R 5 h h 2.4 0;

T 853 - extensao de metal2 alem da via < 1;  
 R 2 h v 1.0;

T 901 - dimensao da depassivacao < 100;  
 R 0 g 100;

T 901 - interseccao de depassivacoes < 100;  
 R 4 g g 100;

T 902 - distancia entre depassivacoes < 70;  
 R 5 g g 70 0;

T transistor n dentro de poco n;  
 R 2 s r 3000;

T transistor p fora do poco n;  
 0 z s;

T contato sem metal;  
 0 c m;

T contato sem poli ou area ativa;  
O c p a;

T via sem metal;  
O v m;

T via sem metal2;  
O v h;

T area ativa sem implante;  
O a d b;

**ANEXO F**

**CELIDAS ESTANDAR EMPLEADAS  
- METODO CONVENCIONAL -**

## F.1 CELDAS ESTANDAR (BIBLIOTECA TENTOS)

Las siguientes son las celdas estándar provenientes de la biblioteca del sistema TENTOS empleadas en los diseños del circuito "Medio Sumador" y "Codificador/Decodificador HDBn".

### OBSERVACIONES:

- a) En las descripciones subsiguientes se indican los símbolos que representan a las celdas en el editor de esquemas EDESQ, su esquema circuitual, y las definiciones subcircuiales SPICE y NDL que describen este esquema.
  
- b) La numeración de los nodos del esquema circuitual corresponde a la NETLIST SPICE, para la NETLIST NDL se ha antepuesto la letra "x" a la numeración de los nodos, y se ha cambiado los nodos "Vcc" por "Vdd" y "0" a fin de adaptar la nominación de los nodos a la sintaxis del language NDL.

F.1.1 CELDA: INV (Inversor)

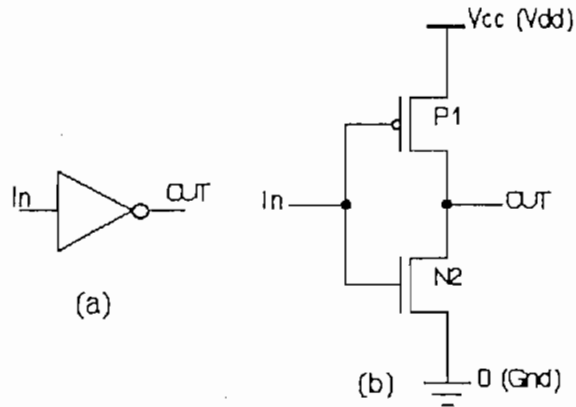


FIGURA F.1 Celda INV (a) Símbolo, (b) Esquema circuital.

```

NETLIST SPICE
*
* CELULA: INVERSOR
*
.SUBCKT INV In OUT vcc
MP1 OUT In vcc vcc PMOS
MN2 OUT In 0 0 NMOS
.ENDS INV
    
```

```

NETLIST NDL
DEFINE INV In Out;
PTRANS In Vdd OUT ;X(P1)
NTRANS In Gnd OUT ;X(N2)
END;
    
```

F.1.2 CELDA: NAND2 (Nand de 2 entradas)

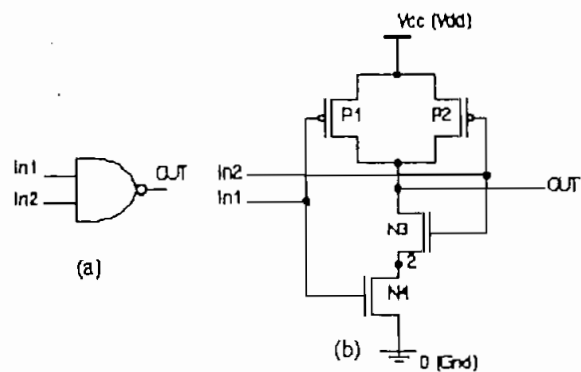


FIGURA F.2 Celda NAND2 (a) Símbolo, (b) Esquema circuital.

```

NETLIST SPICE
*
* CELULA: NAND 2 ENTRADAS
*
.SUBCKT NAND2 In1 In2 OUT vcc
MP1 OUT In1 vcc vcc PMOS
MP2 OUT In2 vcc vcc PMOS
MN3 OUT In1 2 0 NMOS
MN4 2 In2 0 0 NMOS
.ENDS NAND2
    
```

```

NETLIST MDL
DEFINE NAND2 In1 In2 OUT;
LOCAL x2;

PTRANS In1 Vdd OUT ;Z(P1)
PTRANS In2 Vdd OUT ;Z(P2)
NTRANS In1 x2 OUT ;Z(N3)
NTRANS In2 Gnd x2 ;Z(N4)
END;
    
```

F.1.3 CELDA: NAND3 (Nand de 3 entradas)

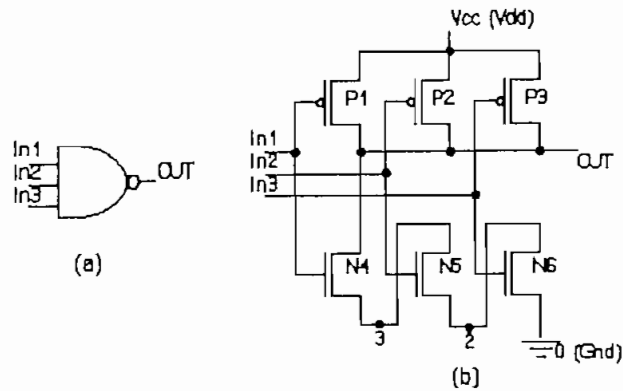


FIGURA F.3 Celda NAND3 (a) Símbolo, (b) Esquema circuital.

NETLIST SPICE

```

*
* CELULA: NAND 3 ENTRADAS
*
.SUBCKT NAND3 In1 In2 In3 OUT vcc
MP1 OUT In1 vcc vcc PMOS
MP2 OUT In2 vcc vcc PMOS
MP3 OUT In3 vcc vcc PMOS
MN4 OUT In1 3 0 NMOS
MN5 3 In2 2 0 NMOS
MN6 2 In3 0 0 NMOS
.ENDS NAND3
    
```

NETLIST MDL

```

DEFINE NAND3 In1 In2 In3 OUT;
LOCAL x2 x3;

PTRANS In1 Vdd OUT ;%(P1)
PTRANS In2 Vdd OUT ;%(P2)
PTRANS In3 Vdd OUT ;%(P3)
NTRANS In1 x3 OUT ;%(N4)
NTRANS In2 x2 x3 ;%(N5)
NTRANS In3 Gnd x2 ;%(N6)
END;
    
```

F.1.4 CELDA: NAND4 (Nand de 4 entradas)

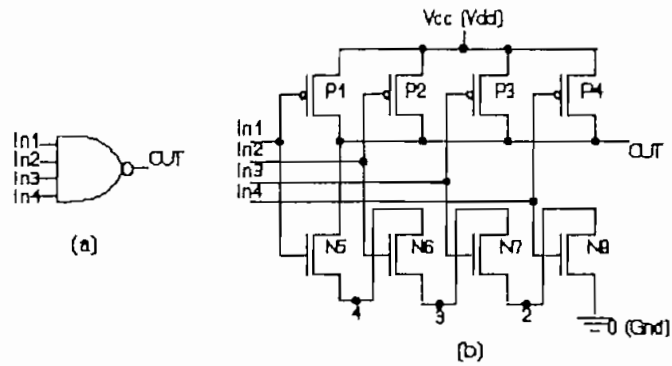


FIGURA F.4 Celda NAND4 (a) Símbolo, (b) Esquema circuital.

NETLIST SPICE

```

*
* CELULA: NAND 4 ENTRADAS
*
.SUBCKT NAND4 In1 In2 In3 In4 OUT vcc
MP1 OUT In1 vcc vcc PMOS
MP2 OUT In2 vcc vcc PMOS
MP3 OUT In3 vcc vcc PMOS
MP4 OUT In4 vcc vcc PMOS
MN5 OUT In1 4 0 NMOS
MN6 4 In2 3 0 NMOS
MN7 3 In3 2 0 NMOS
MN8 2 In4 0 0 NMOS
.ENDS NAND4
    
```

NETLIST MDL

```

DEFINE NAND4 In1 In2 In3 In4 OUT;
LOCAL x2 x3 x4;

PTRANS I1 Vdd OUT ;X(P1)
PTRANS I2 Vdd OUT ;X(P2)
PTRANS I3 Vdd OUT ;X(P3)
PTRANS I4 Vdd OUT ;X(P4)
NTRANS I1 x4 OUT ;X(N5)
NTRANS I2 x3 x4 ;X(N6)
NTRANS I3 x2 x3 ;X(N7)
NTRANS I4 Gnd x2 ;X(N8)
END;
    
```



F.1.5

CELDA: AND2 (And de 2 entradas)

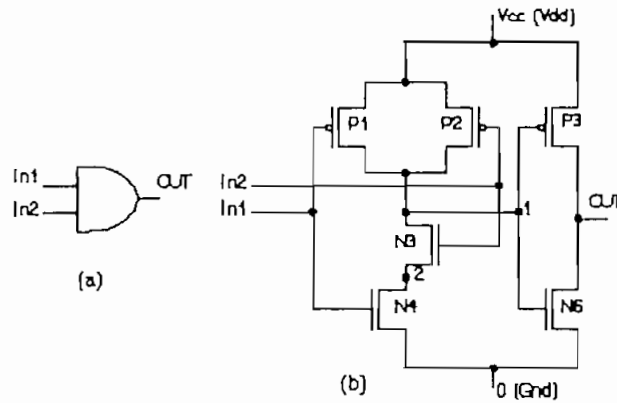


FIGURA F.5 Celda AND2 (a) Símbolo, (b) Esquema circuital.

NETLIST SPICE

```

†
† CELULA: AND 2 ENTRADAS
†
.SUBCKT AND2 In1 In2 OUT vcc
MP1 1 In1 vcc vcc PMOS
MP2 1 In2 vcc vcc PMOS
MN4 2 In1 0 0 NMOS
MN5 1 In2 2 0 NMOS
MN6 OUT 1 0 0 NMOS
MP3 OUT 1 vcc vcc PMOS
.ENDS AND2

```

NETLIST MDL

```

DEFINE AND2 I1 I2 Out;
LOCAL x2 x1;

PTRANS I1 Vdd x1 ;Z(P1)
PTRANS I2 Vdd x1 ;Z(P2)
NTRANS I1 Gnd x2 ;Z(N4)
NTRANS I2 x2 x1 ;Z(N5)
NTRANS x1 Gnd Out ;Z(N6)
PTRANS x1 Vdd Out ;Z(P3)
END;

```

F.1.6 CELDA: AND3 (And de 3 entradas)

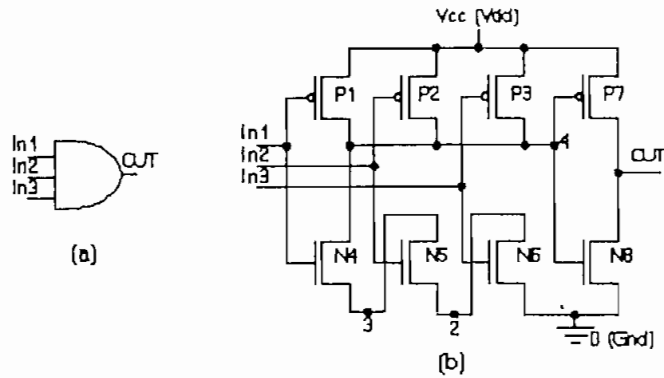


FIGURA 10.7 Celda AND3 (a) Símbolo, (b) Esquema circuital.

NETLIST SPICE

```

*
* CELULA: AND 3 ENTRADAS
*
.SUBCKT AND3 In1 In2 In3 OUT vcc
MP1 4 In1 vcc vcc PMOS
MP2 4 In2 vcc vcc PMOS
MP3 4 In3 vcc vcc PMOS
MN4 4 In1 3 0 NMOS
MN5 3 In2 2 0 NMOS
MN6 2 In3 0 0 NMOS
MP7 OUT 4 vcc vcc PMOS
MN8 OUT 4 0 0 NMOS
.ENDS AND3
    
```

NETLIST MDL

```

DEFINE AND3 In1 In2 In3 Out;
LOCAL x2 x3 x4;
PTRANS In1 Vdd x4 ;X(P1)
PTRANS In2 Vdd x4 ;X(P2)
PTRANS In3 Vdd x4 ;X(P3)
NTRANS In1 x3 x4 ;X(N4)
NTRANS In2 x2 x3 ;X(N5)
NTRANS In3 Gnd x2 ;X(N6)
PTRANS x4 Vdd Out ;X(P7)
NTRANS x4 Gnd Out ;X(N8)
END;
    
```

F.1.7 CELDA: AND4 (And de 4 entradas)

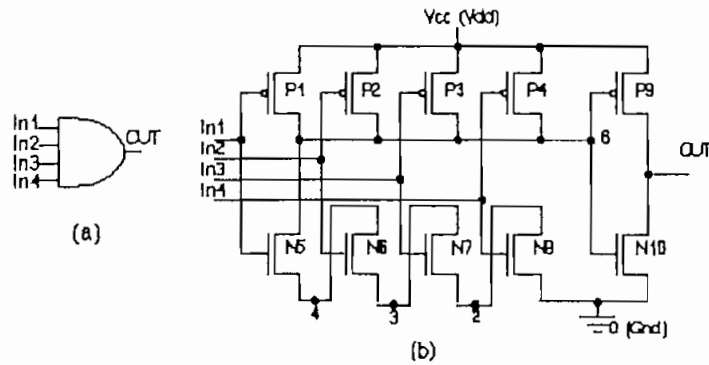


FIGURA F.7 Celda AND4 (a) Símbolo, (b) Esquema circuital.

NETLIST SPICE

```

*
* CELULA: AND 4 ENTRADAS
*
.SUBCKT AND4 In1 In2 In3 In4 OUT vcc
MP1 6 In1 vcc vcc PMOS
MP2 6 In2 vcc vcc PMOS
MP3 6 In3 vcc vcc PMOS
MP4 6 In4 vcc vcc PMOS
MN5 6 In1 4 0 NMOS
MN6 4 In2 3 0 NMOS
MN7 3 In3 2 0 NMOS
MN8 2 In4 0 0 NMOS
MP9 OUT 6 vcc vcc PMOS
MN10 OUT 6 0 0 NMOS
.ENDS AND4
    
```

NETLIST MDL

```

DEFINE AND4 In1 In2 In3 In4 Out;
LOCAL x2 x3 x4 x6;
PTRANS In1 Vdd x6 ;Z(P1)
PTRANS In2 Vdd x6 ;Z(P2)
PTRANS In3 Vdd x6 ;Z(P3)
PTRANS In4 Vdd x6 ;Z(P4)
NTRANS In1 x4 x6 ;Z(N5)
NTRANS In2 x3 x4 ;Z(N4)
NTRANS In3 x2 x3 ;Z(N7)
NTRANS In4 Gnd x2 ;Z(N8)
PTRANS x6 Vdd Out ;Z(P9)
NTRANS x6 Gnd Out ;Z(P10)
END;
    
```

F.1.8 CELDA: OR2 (Or de 2 entradas)

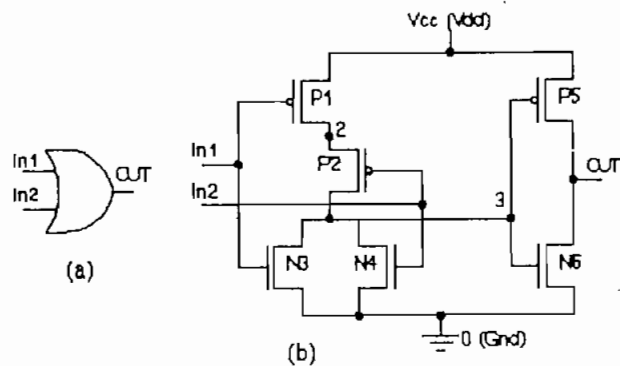


FIGURA F.8 Celda OR2 (a) Símbolo, (b) Esquema circuital.

NETLIST SPICE

```

*
* CELULA: OR 2 ENTRADAS
*
.SUBCKT OR2 In1 In2 OUT vcc
MP1 2 In1 vcc vcc PMOS
MP2 3 In2 2 vcc PMOS
MN3 3 In1 0 0 NMOS
MN4 3 In2 0 0 NMOS
MP5 OUT 3 vcc vcc PMOS
MN6 OUT 3 0 0 NMOS
.ENDS OR2

```

NETLIST MDL

```

DEFINE OR2 In1 In2 Out;
LOCAL x2 x3;

PTRANS In1 Vdd x2 ;X(P1)
PTRANS In2 x2 x3 ;X(P2)
NTRANS In1 Gnd x3 ;X(N3)
NTRANS In2 Gnd x3 ;X(N4)
PTRANS x3 Vdd Out ;X(P5)
NTRANS x3 Gnd Out ;X(N6)
END;

```

F.1.9 CELDA: OR3 (Or de 3 entradas)

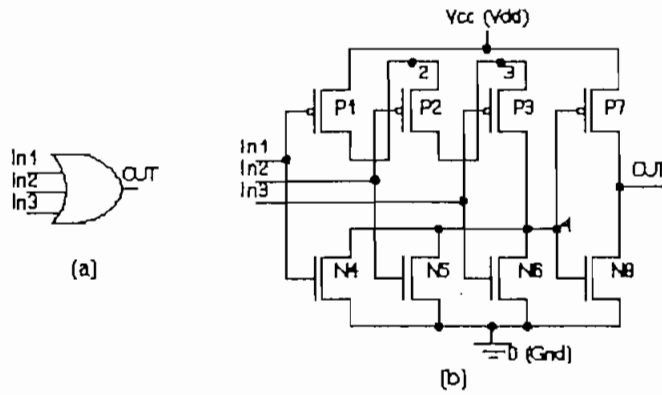


FIGURA F.9 Celda OR3 (a) Símbolo, (b) Esquema circuital.

NETLIST SPICE

```

*
* CELULA: OR 3 ENTRADAS
*
.SUBCKT OR3 In1 In2 In3 OUT vcc
MP1 2 In1 vcc vcc PMOS
MP2 3 In2 2 vcc PMOS
MP3 4 In3 3 vcc PMOS
MN4 4 In1 0 0 NMOS
MN5 4 In2 0 0 NMOS
MN6 4 In3 0 0 NMOS
MP7 OUT 4 vcc vcc PMOS
MN8 OUT 4 0 0 NMOS
.ENDS OR3
    
```

NETLIST MDL

```

DEFINE OR3 In1 In2 In3 Out;
LOCAL x2 x3 x4;

PTRANS In1 Vdd x2 ;Z(1)
PTRANS In2 x2 x3 ;Z(2)
PTRANS In3 x3 x4 ;Z(3)
NTRANS In1 Gnd x4 ;Z(4)
NTRANS In2 Gnd x4 ;Z(5)
NTRANS In3 Gnd x4 ;Z(6)
PTRANS x4 Vdd Out ;Z(7)
NTRANS x4 Gnd Out ;Z(8)
END;
    
```

F.1.10 CELDA:FFD MS (Flip Flop Maestro Esclavo)

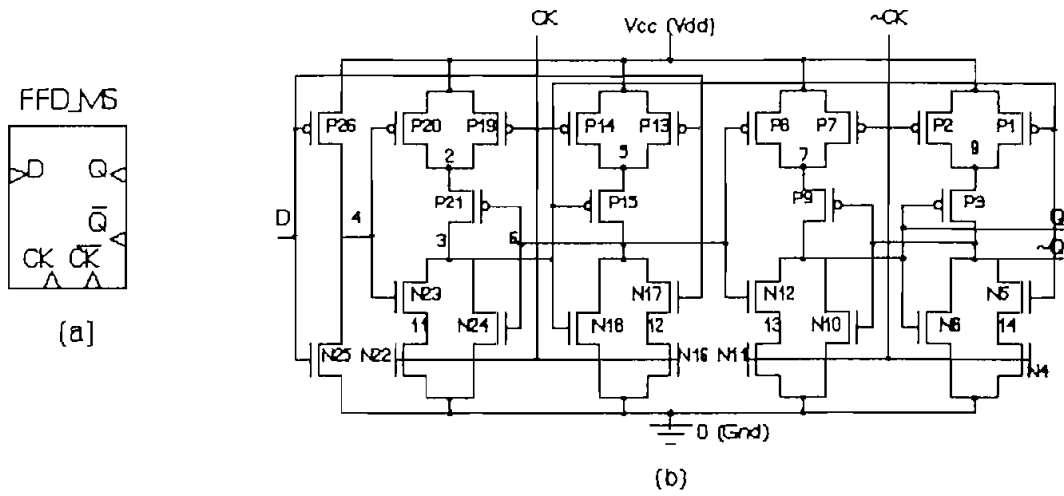


FIGURA F.10 Celda FFD\_MS (a) Símbolo, (b) Esquema circuital.

```

NETLIST SPICE
‡
‡ CELULA: FFD_MS.CEL
‡
.SUBCKT FFD_MS D CK ^CK ^Q Q vcc
MP1 9 3 vcc vcc PMOS
MP2 9 ^CK vcc vcc PMOS
MP3 ^Q 0 9 vcc PMOS
MN4 14 ^CK 0 0 NMOS
MN5 ^Q 3 14 0 NMOS
MN6 ^Q 0 0 0 NMOS
MP7 7 ^CK vcc vcc PMOS
MP8 7 6 vcc vcc PMOS
MP9 0 ^Q 7 vcc PMOS
MN10 0 ^Q 0 0 NMOS
MN11 13 ^CK 0 0 NMOS
MN12 0 6 13 0 NMOS
MP13 5 D vcc vcc PMOS
MP14 5 CK vcc vcc PMOS
MP15 6 3 5 vcc PMOS
MN16 12 CK 0 0 NMOS
MN17 6 D 12 0 NMOS
MN18 6 3 0 0 NMOS
MP19 2 CK vcc vcc PMOS
MP20 2 4 vcc vcc PMOS
MP21 3 6 2 vcc PMOS
MN22 11 CK 0 0 NMOS
MN23 3 4 11 0 NMOS
MN24 3 6 0 0 NMOS
MN25 4 D 0 0 NMOS
MP26 4 D vcc vcc PMOS
.ENDS FFD_MS
    
```

```

NETLIST MDL
DEFINE FFD_MS D CK CKN QN Q;
LOCAL x2 x3 x4 x5 x6 x7 x9 x11 x12 x13 x14;
PTRANS x3 Vdd x9; Z(P1)
PTRANS CKN Vdd x9; Z(P2)
PTRANS 0 x9 QN; Z(P3)
NTRANS CKN Gnd x14; Z(N4)
NTRANS x3 x14 QN; Z(N5)
NTRANS 0 Gnd QN; Z(N6)
PTRANS CKN Vdd x7; Z(P7)
PTRANS x6 Vdd x7; Z(P8)
PTRANS QN x7 0; Z(P9)
NTRANS QN Gnd 0; Z(N10)
NTRANS CKN Gnd x13; Z(N11)
NTRANS x6 x13 0; Z(N12)
PTRANS D Vdd x5; Z(P13)
PTRANS CK Vdd x5; Z(P14)
PTRANS x3 x5 x6; Z(P15)
NTRANS CK Gnd x12; Z(N16)
NTRANS D x12 x6; Z(N17)
NTRANS x3 Gnd x6; Z(N18)
PTRANS CK Vdd x2; Z(P19)
PTRANS x4 Vdd x2; Z(P20)
PTRANS x6 x2 x3; Z(P21)
NTRANS CK Gnd x11; Z(N22)
NTRANS x4 x11 x3; Z(N23)
NTRANS x6 Gnd x3; Z(N24)
NTRANS D Gnd x4; Z(N25)
PTRANS D Vdd x4; Z(P26)
END;
    
```

## F.2 CELDAS DE PADS (BIBLIOTECA PADLIB2 ECPD15)

### OBSERVACIONES

- a) A continuación se indica un listado de las celdas de PADS existentes en la biblioteca PADLIB2 de la empresa European Silicon Structures ES2 (Francia).
  
- b) Únicamente se indican las características eléctricas de las celdas que se han empleado en los diseños de los circuitos "Medio Sumador MED\_SUM" y del Codificador/Decodificador HDBn, el catálogo completo de celdas puede ser estudiado en la referencia:  
"ES2 ECPD15 & ECPD12 Library Databook - Standard Cells Libraries", European Silicon Structures Es2 - Francia, Abril 1991.
  
- c) No es posible indicar la estructura geométrica ni eléctrica de las celdas ya que esta información es reservada por el fabricante que se encarga de colocar los PADS en el lugar indicado por el diseñador el momento de fundir un circuito específico.

F.2.1 Catálogo de celdas

CELL NAME	FUNCTION	DATE	REVISION
IOS1K	4mA Bidireccional buffer. CMOS input	June 90	REV.1.4
IOS2K	8mA Bidireccional buffer. CMOS input	June 90	REV.1.4
IOS4K	8mA Bidireccional buffer. CMOS input	June 90	REV.1.4
IO6K	24mA Bidireccional buffer. CMOS input	June 90	REV.1.4
IOS1P	4mA Bidireccional buffer. TTL input	June 90	REV.1.4
IOS2P	8mA Bidireccional buffer. TTL input	June 90	REV.1.4
IOS4P	16mA Bidireccional buffer. TTL input	June 90	REV.1.4
IOS8P	24mA Bidireccional buffer. TTL input	June 90	REV.1.4
IPS8B	CMOS Input buffer	June 90	REV.1.4
IPS8C	CMOS Input buffer. 200kOhm pullup	June 90	REV.1.4
IPS8D	CMOS Input buffer. 10kOhm pullup	June 90	REV.1.4
IPS8E	TTL Input buffer	June 90	REV.1.4
IPS8H	TTL Input buffer. 200kOhm pullup	June 90	REV.1.4
IPS8I	TTL Input buffer. 10kOhm pullup	June 90	REV.1.4
IPS1E	CMOS Input buffer	June 90	REV.1.4
IPS4E	CMOS Input buffer	June 90	REV.1.4
IPS1J	CMOS Input buffer	June 90	REV.1.4
IPS4J	CMOS Input buffer	June 90	REV.1.4
OPS1U	4mA Output buffer	June 90	REV.1.4
OPS2U	8mA Output buffer	June 90	REV.1.4
OPS4U	16mA Output buffer	June 90	REV.1.4
OPS8U	24mA Output buffer	June 90	REV.1.4
OPS1W	4mA Tri-state output buffer	June 90	REV.1.4
OPS2W	8mA Tri-state output buffer	June 90	REV.1.4
OPS4W	16mA Tri-state output buffer	June 90	REV.1.4
OPS8W	24mA Tri-state output buffer	June 90	REV.1.4
OPS1Z	4mA Open drain output buffer	June 90	REV.1.4
OPS2Z	8mA Open drain output buffer	June 90	REV.1.4
OPS4Z	16mA Open drain output buffer	June 90	REV.1.4
OPS8Z	24mA Open drain output buffer	June 90	REV.1.4
GNDPY	0 Volt pad for periphery	June 90	REV.1.4
GNDCO	0 Volt pad for core	June 90	REV.1.4
PWRCO	5 Volt pad for core	June 90	REV.1.4
PWRPY	5 Volt pad for periphery	June 90	REV.1.4
PWRBK	5 Volt pad for blocks	June 90	REV.1.4
OSC1	Crystal oscillator 5 to 50 MHz		
OSC2	Crystal oscillator 0.1 to 5 MHz		
OSC3	Crystal oscillator 32 kHz		
RCD2	Low power RC oscillator		
PWRCDPDR	5 Volt pad for inner pad ring with power on reset	Feb. 91	REV.1.5
PWRPYPDR	5 Volt pad for periphery with power on reset	Feb. 91	REV.1.5

TABLA F.1 PADLIB2 (ECPD15).

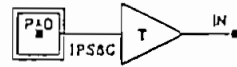


ES2 PadLib2 process ECPD15  
REV.1.4

JUN 98

Input Buffer  
TTL input

IPS8G



PARAMETER	VALUE	UNIT
Size	150*559	um2
Cin_PAD	6.50	pF
Fanout_IN	3.20	pF
total cap	8.25	pF
transistors	26	FALSE

PARAMETER	CONDITIONS	MILITARY		INDUSTRIAL		UNIT
		MIN	MAX	MIN	MAX	
V <sub>IH</sub>	VCC worst case VCC=5.5V	2.0	-	2.0	-	V
V <sub>IL</sub>	VCC worst case VCC=4.5V	-	0.8	-	0.8	V
I <sub>IL</sub>	V <sub>I</sub> =0, temp=full range	-10	10	-10	10	uA
I <sub>IH</sub>	V <sub>I</sub> =VCC, temp=full range	-10	10	-10	10	uA

PARAMETER	FROM	TO	MIN	TYP	MAX	MIL	UNIT
t <sub>phl</sub>	PAD	IN	0.98	2.34	4.61	5.57	ns
t <sub>plh</sub>	PAD	IN	0.11	0.26	0.51	0.62	ns
Δt <sub>phl</sub>	PAD	IN	0.059	0.14	0.28	0.33	ns/pF
Δt <sub>plh</sub>	PAD	IN	0.084	0.20	0.39	0.48	ns/pF

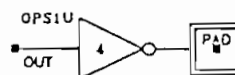
ES2 PadLib2

process ECPD15  
REV.1.4

JUN 98

Output Buffer  
4mA output

OPS1U



PARAMETER	VALUE	UNIT
Size	150*559	um2
Cin_OUT	0.20	pF
Fanout_PAD	100.0	pF
total_cap	9.73	pF
transistors	14	FALSE

PARAMETER	CONDITIONS	MILITARY		INDUSTRIAL		UNIT
		MIN	MAX	MIN	MAX	
IOL	VOL > 0.5V	-	+4.0	-	+8.0	mA
IOH	VOH < VCC - 0.5V	-	-4.0	-	-8.0	mA
VOH	IOH = -4.0mA & VCC = 4.5V	4.0	-	-	-	V
VOH	IOH = -5.5mA & VCC = 4.5V	-	-	4.0	-	V
VOL	IOL = +4.0mA & VCC = 4.5V	-	0.5	-	-	V
VOL	IOL = +5.5mA & VCC = 4.5V	-	-	-	0.5	V

PARAMETER	FROM	TO	MIN	TYP	MAX	MIL	UNIT
tphl	OUT	PAD	0.93	2.22	4.37	5.28	ns
tplh	OUT	PAD	0.98	2.33	4.59	5.55	ns
Δtphl	OUT	PAD	0.019	0.045	0.089	0.11	ns/pF
Δtplh	OUT	PAD	0.019	0.045	0.089	0.11	ns/pF

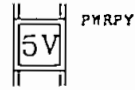
ES2 PadLib2

process ECPD15  
REV.1.4

JUN 98

5V pad  
for periphery

PWRPY



PARAMETER	VALUE	UNIT
Size	150*559	um2

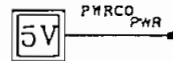
ES2 PadLib2

process ECPD15  
REV.1.4

JUN 98

5V pad  
for core

PWRCO



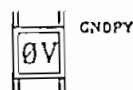
PARAMETER	VALUE	UNIT
Size	150*559	um2

ES2 PadLib2	process ECPD15 REV.1.4
-------------	---------------------------

JUN 98

0V pad  
for periphery

GNDPY



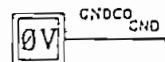
PARAMETER	VALUE	UNIT
Size	150*559	um2

ES2 PadLib2	process ECPD15 REV.1.4
-------------	---------------------------

JUN 98

0V pad  
for core

GNDCO



PARAMETER	VALUE	UNIT
Size	150*559	um2

**ANEXO G**  
**REGLAS DE DISEÑO**

# G.1 REGLAS DE DISEÑO DE LA EUROPEAN SILICON STRUCTURES (ES2)

: DUAL LAYER METAL 1.5um CMOS DESIGN RULES (EC7015/1)

: REF A

LAYER SYMBOLS

MASK No	TITLE	LAYOUT SYMBOL
1 0	N-WELL IMPLANT	
2 0	ACTIVE AREA	
5 0	POLYSILICON	
6 0	N+ SOURCE/DRAIN IMPLANT	
6 5	P+ SOURCE/DRAIN IMPLANT	
7 0	CONTACTS	
8 0	METAL 1	
7 5	VIAS	
8 5	METAL 2	
9 0	PASSIVATION	

## 1.0. TOPOLOGICAL DESIGN RULES

THE FOLLOWING CONVENTIONS ARE USED FOR THE DEFINITION OF THE LAYOUT RULES. LAYER B IS DEFINED AFTER LAYER A IN THE DESCRIPTION FLOW. ALL DIMENSIONS ARE ALWAYS EITHER ABSOLUTE MINIMA OR VALUES TO BE MET EXACTLY.

6.1. WIDTH : MINIMUM DISTANCE BETWEEN THE EDGES INSIDE A FIGURE



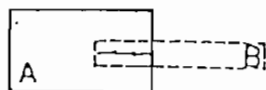
EUROPEAN SILICON STRUCTURES  
(ES2)

: REFERENCE : PAGE 5  
: AG1-0201-A : OF 30

6.2. SPACING : MINIMUM DISTANCE BETWEEN THE EDGES OF TWO FIGURES (SAME LAYER OR DIFFERENT LAYERS) TO PREVENT INTERACTION.



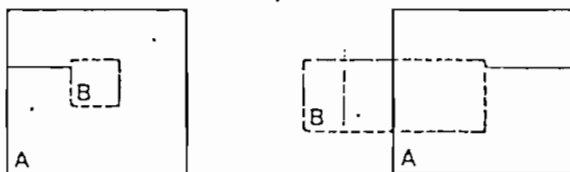
6.3. COINCIDENCE : MINIMUM DISTANCE COMMON TO TWO FIGURES IN LAYERS A AND B RESPECTIVELY.



6.4. OVERLAP : MINIMUM DISTANCE THAT FIGURE IN LAYER B HAS TO EXTERIO OUTSIDE FIGURE IN LAYER A



6.5. MARGIN : MINIMUM DISTANCE THAT FIGURE IN LAYER B HAS TO BE WITHIN FIGURE IN LAYER A



6.6. HOT DIFFUSION : ALL N+ DIFFUSION REGIONS OUTSIDE THE N-WELL WHICH HAVE A POTENTIAL NOT EQUAL TO THE SUBSTRATE VOLTAGE. ALL P+ DIFFUSION REGIONS INSIDE THE N-WELL WHICH HAVE A POTENTIAL NOT EQUAL TO THE N-WELL POTENTIAL.

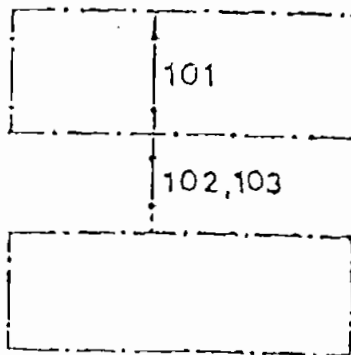
6.7. COLD DIFFUSION : OUTSIDE THE N-WELL A DIFFUSION WHICH HAS THE SAME POTENTIAL AS THE SUBSTRATE. INSIDE THE N-WELL A DIFFUSION WHICH HAS THE SAME POTENTIAL AS THE N-WELL.

6.8. HOT N-WELL : N-WELL NOT CONNECTED TO THE MOST POSITIVE VOLTAGE (VDD).

6.9. COLD N-WELL : N-WELL CONNECTED TO THE MOST POSITIVE VOLTAGE (VDD).

LAYER : NUMBER : 10  
 NAME : N-WELL IMPLANT

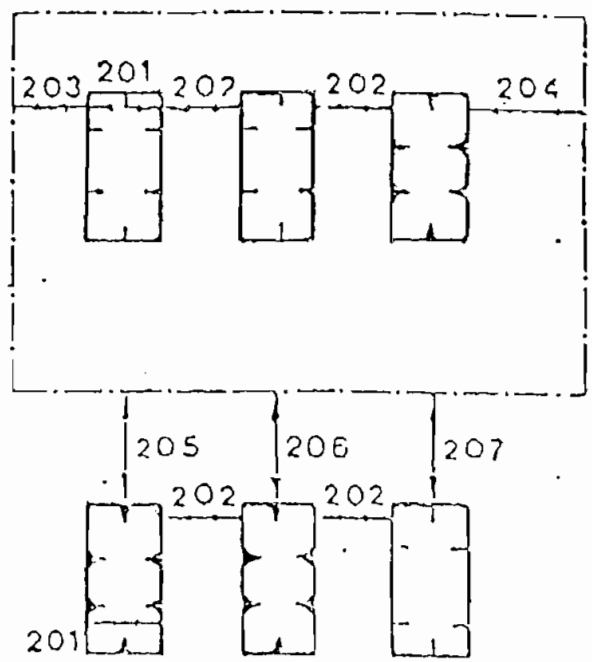
RULE NUMBER	PARAMETER	MIN. DIMENS. (um)
101	WIDTH OF N-WELL	6.0
102	SPACING BETWEEN COLD WELLS IF BELOW 6 um, THEN MERGE	6.0
103	SPACING BETWEEN WELLS WITH AT LEAST ONE NOT CONNECTED TO VDD (HOT WELL).	12.8
104	THE USAGE OF N-WELL REGION FOR RESISTORS IS NOT ALLOWED EXCEPT AFTER SPECIFIC AGREEMENT WITH ES2/MTD, SILICON PROCESSING DPT.	





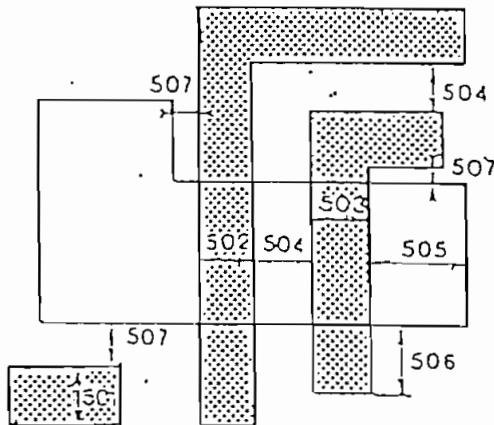
LAYER : NUMBER : 2 0  
: NAME : ACTIVE AREA

RULE NUMBER	PARAMETER	MIN. DIMENS. (um)
201	WIDTH OF N+ / P+ ACTIVE AREA FOR INTERCONNECTS	1.6
2011	WIDTH OF GATE REGION IN A TRANSISTOR	2.0
202	SPACING BETWEEN TWO ACTIVE AREAS OR DIFFUSIONS	3.2
2021	SPACING BETWEEN TWO CORNERS OF ACTIVE AREAS OR DIFFUSION AFTER SIZING BY 0.8 PER EDGE	1.6
203	MARGIN OF P+ AREA TO N-WELL	4.8
204	MARGIN OF N+ AREA TO N-WELL (N+ AREA INSIDE WELL FOR WELL TAP)	0.0
205	SPACING BETWEEN N+ AREA AND N-WELL (COLD WELL)	4.8
206	SPACING BETWEEN N+ AREA AND N-WELL (HOT WELL)	5.0
207	SPACING BETWEEN P+ AREA AND N-WELL (P+ AREA OUTSIDE WELL FOR SUBSTRATE TAP)	4.8



NUMBER : 50  
 NAME : POLYSILICON

NUMBER	PARAMETER	MIN. DIMENS. (um)
501	WIDTH OF POLYSILICON : INTERCONNECT	1.6
502	WIDTH OF POLYSILICON : PMOS	1.6
503	WIDTH OF POLYSILICON : PMOS	1.6
504	SPACING BETWEEN TWO STRIPES OF POLYSILICON	2.4
505	MARGIN OF POLYSILICON TO ACTIVE AREA : ACTUAL VALUE SHOULD BE DETERMINED BY THE DESIGN ENGINEER WITH : CONSIDERATION TO SPREADING RESISTANCE.	2.4
506	POLYSILICON OVERLAP OF ACTIVE AREA	1.6
507	SPACING BETWEEN POLYSILICON AND ACTIVE AREA	0.8

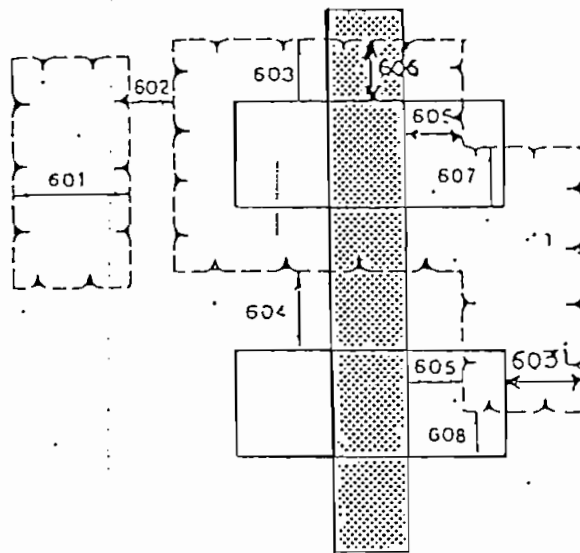


TITLE 1 DUAL LAYER METAL 1.5um CMOS DESIGN RULES (ECP015/1)

1.  
: KEY A

LAYER 1 NUMBER : 6 0  
: NAME : N+ SOURCE/DRAIN IMPLANT

RULE NUMBER	PARAMETER	MIR. DIMENS. (um)
601	: WIDTH	2.4
602	: SPACING : MERGE IF LESS THAN 2.4 um	2.4
603	: OVERLAP OF N+ ACTIVE AREA ON SUBSTRATE	2.0
6031	: OVERLAP OF N+ ACTIVE AREA ON N-WELL	0.8
604	: SPACING TO P+ ACTIVE AREA ON N-WELL	2.0
6041	: SPACING TO P+ ACTIVE AREA ON SUBSTRATE	0.8
605	: SPACING TO POLYSILICON GATE OVER P+ ACTIVE AREA	2.0
606	: OVERLAP OF POLYSILICON GATE OVER N+ ACTIVE AREA	2.0
607	: COINCIDENCE OF N+ IMPLANT TO ACTIVE AREA FOR WHEN N+ IMPLANT : DOES NOT OVERLAP ACTIVE AREA.	2.0
608	: N+ IMPLANT MARGIN TO ACTIVE AREA FOR N-WELL CONTACT	2.0

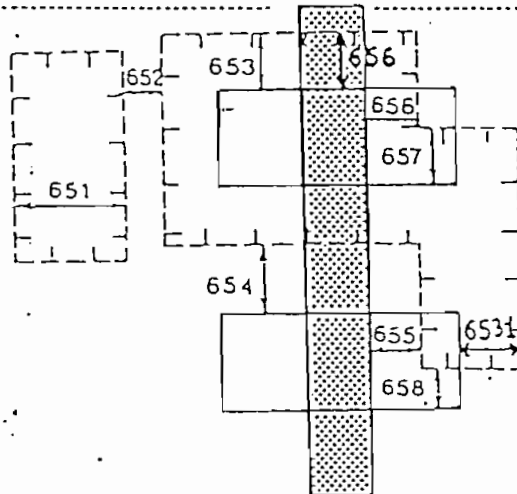


EUROPEAN SILICON STRUCTURES  
(ES2)

: REFERENCE : PAGE 10  
: AG1-DRO4-A : OF 30

LAYER : NUMBER : 6 S  
NAME : P+ SOURCE/DRAIN IMPLANT

RULE NUMBER	PARAMETER	MIB. DIMENS.(um)
651	: WIDTH	2.4
652	: SPACING : MERGE IF LESS THAN 2.4 um	2.4
653	: OVERLAP OF P+ ACTIVE AREA ON N-WELL	2.0
6531	: OVERLAP OF P+ ACTIVE AREA ON SUBSTRATE	0.8
654	: SPACING TO N+ ACTIVE AREA ON SUBSTRATE	2.0
6541	: SPACING TO N+ ACTIVE AREA ON N-WELL	0.8
655	: SPACING TO POLYSILICON GATE OVER N+ ACTIVE AREA	2.0
656	: OVERLAP OF POLYSILICON GATE OVER P+ ACTIVE AREA	2.0
657	: COINCIDENCE OF P+ IMPLANT TO ACTIVE AREA WHEN P+ IMPLANT DOES NOT OVERLAP ACTIVE AREA	2.0
658	: P+ IMPLANT MARGIN TO ACTIVE AREA FOR SUBSTRATE CONTACT	2.0
659	: ACTIVE AREA MUST BE N+ OR P+ IMPLANTED	
660	: NO COINCIDENCE BETWEEN N+ AND P+ IMPLANT	
661	: ADJACENT (BUTTING) P+ AND N+ ACTIVE AREAS MUST BE AT THE SAME POTENTIAL.	

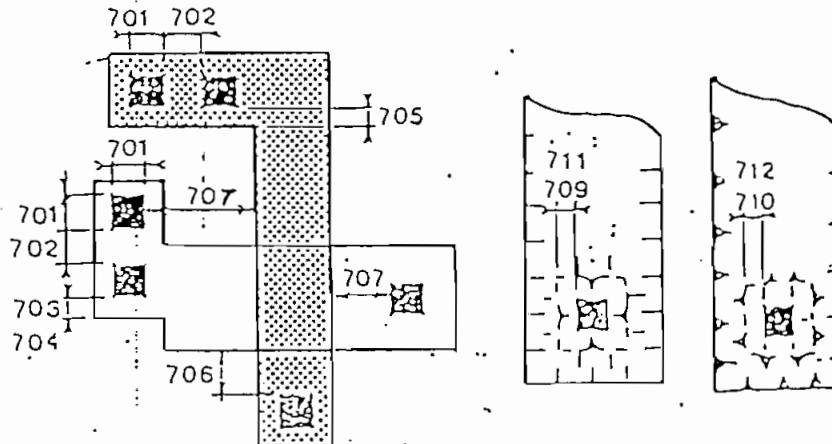


TITLE : DUAL LAYER METAL 1.5um CMOS DESIGN RULES (EC0013/1)

1  
REV A

LAYER : NUMBER : 10  
: NAME : CONTACTS

RULE NUMBER	PARAMETER	UNIT	MIN. DIM.
701	: MINIMUM AND MAXIMUM CONTACT WIDTH		2.0
702	: SPACING		2.0
703	: DIFFUSION CONTACT MARGIN TO ACTIVE AREA		1.0
705	: POLY CONTACT MARGIN TO POLYSILICON		1.0
706	: POLY CONTACT SPACING TO ACTIVE AREA		1.4
707	: DIFFUSION CONTACT SPACING TO POLYSILICON GATE		1.4
709	: MARGIN OF DIFFUSION CONTACTS TO N+ IMPLANT (COLO DIFFUSIONS ONLY)		0.6
710	: MARGIN OF DIFFUSION CONTACTS TO P+ IMPLANT (COLO DIFFUSIONS ONLY)		0.6
711	: SPACING OF DIFFUSION CONTACTS TO P+ IMPLANT (COLO DIFFUSIONS ONLY)		0.6
712	: SPACING OF DIFFUSION CONTACTS TO N+ IMPLANT (COLO DIFFUSIONS ONLY)		0.6
713	: CONTACT TO POLYSILICON ARE NOT ALLOWED OVER ACTIVE AREA		
714	: SHORTING CONTACTS ARE NOT ALLOWED		
715	: CONTACTS MUST BE ON POLYSILICON OR AN ACTIVE AREA		
716	: MAXIMUM CURRENT THROUGH A CONTACT IS AT :		
		700C	0.70 (mA)
		850C	0.50 " "
		1000C	0.40 " "
		1100C	0.15 " "

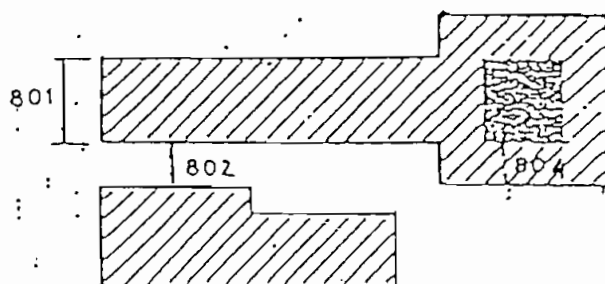


EUROPEAN SILICON STRUCTURES  
(ES2)

: REFERENCE : PAGE 1  
: AC1-0004-A : DF ]

CR : NUMBER : 8 8  
 : NAME : METAL 1

ULZ NUMBER	PARAMETER	MIN. DIMENS. (µm)
801	WIDTH	2.4
802	SPACING	2.4
8021	SPACING BETWEEN TWO CORNERS OF METAL 1 AFTER SIZING BY 0.2 PER EDGE	2.0
804	OVERLAP OF CONTACT	1.0
805	ALL CONTACTS MUST BE COVERED WITH METAL 1	
806	MAXIMUM AVERAGE CURRENT FOR A LINE OF WIDTH Y IS, IN mA :	
	- AT 70 DC : $0.3 \cdot Y \cdot \text{SQRT}(Y/2.4)$	
	- AT 85 DC : $0.2 \cdot Y \cdot \text{SQRT}(Y/2.4)$	
	- AT 100 DC : $0.15 \cdot Y \cdot \text{SQRT}(Y/2.4)$	
	- AT 140 DC : $0.06 \cdot Y \cdot \text{SQRT}(Y/2.4)$ WHERE Y IS THE WIDTH IN MICRONS	
	MAXIMUM PEAK CURRENT IS : $7.5 \cdot Y$ (mA).	
807	THE USAGE OF METAL 1 FOR CAPACITOR IS NOT ALLOWED SINCE RELATED DIELECTRIC LAYERS ARE SUBJECT TO CHANGE WITH PROCESS OPTIMIZATION	

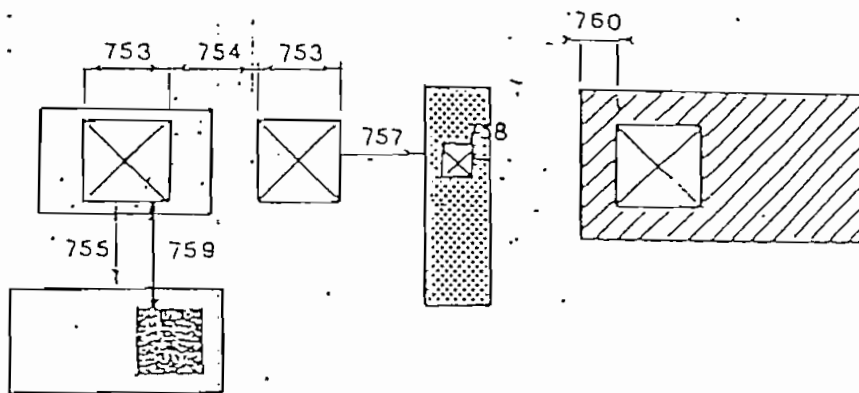


EUROPEAN SILICON STRUCTURES  
 (ES2)

: REFERENCE : PAGE 13  
 : AC1-0901-A : OF 30

LAYER : NUMBER : 7 5  
 : NAME : VIAS

RULE NUMBER	PARAMETER	MIN. DIM.(µm)
751	VIA OPENING IS ALLOWED ONLY BETWEEN METAL 1 AND METAL 2	
752	STACKED VIA OR CONTACT NOT ALLOWED	
753	MINIMUM AND MAXIMUM WIDTH (EXCEPT FOR BOXING PADS).	2.0
754	SPACING	2.0
757	SPACING TO POLYSILICON	2.2
758	MARGIN TO POLYSILICON	2.2
759	SPACING TO CONTACT	2.0
760	MARGIN TO METAL 1	1.0
761	MAXIMUM CURRENT THROUGH A VIA IS AT	
	- 70 DC : 2.0 (mA)	
	- 85 DC : 1.4 "	
	- 100 DC : 1.0 "	
	- 140 DC : 0.5 "	

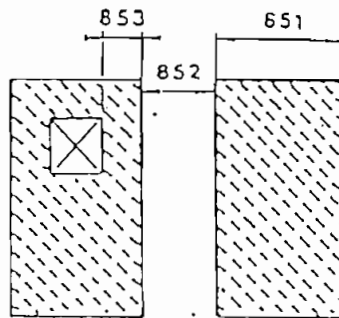


4. LAYER METAL 1.5um CMOS DESIGN RULES (ECPD15/1)

REV A

ME1 : M1  
ME2 : METAL 2

NUMBER	PARAMETER	MIN. DIMENS. (um)
	WIDTH	2.4
	SPACING	2.4
1	SPACING BETWEEN TWO CORNERS AFTER SIZING BY 0.2	2.0
	OVERLAP OF VIA	1.0
	MAXIMUM CURRENT THROUGH A LINE OF WIDTH W IS AT :	
	- 70 DC : $0.80 * W * \text{SQRT}(W/2.4)$ (mA)	
	- 85 DC : $0.60 * W * \text{SQRT}(W/2.4)$ "	
	- 100 DC : $0.40 * W * \text{SQRT}(W/2.4)$ "	
	- 140 DC : $0.20 * W * \text{SQRT}(W/2.4)$ "	
	MAXIMUM PEAK CURRENT IS $15 * W$ (mA).	
5	THE USAGE OF METAL 2 FOR CAPACITOR IS NOT ALLOWED SINCE RELATED DIELECTRIC LAYERS ARE SUBJECT TO CHANGE FOR PROCESS OPTIMIZATION	



EUROPEAN SILICON STRUCTURES  
(ES2)

REFERENCE : PAGE 15  
AC1-0000-A : OF 30



1. DUAL LAYER METAL 1.5um CHDS DESIGN RULES ((CP015/1))

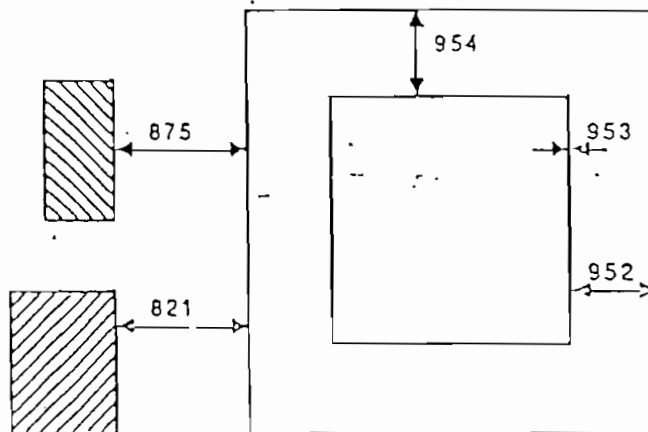
1. KEY A

1. NUMBER : -

1. NAME : BONDING PADS

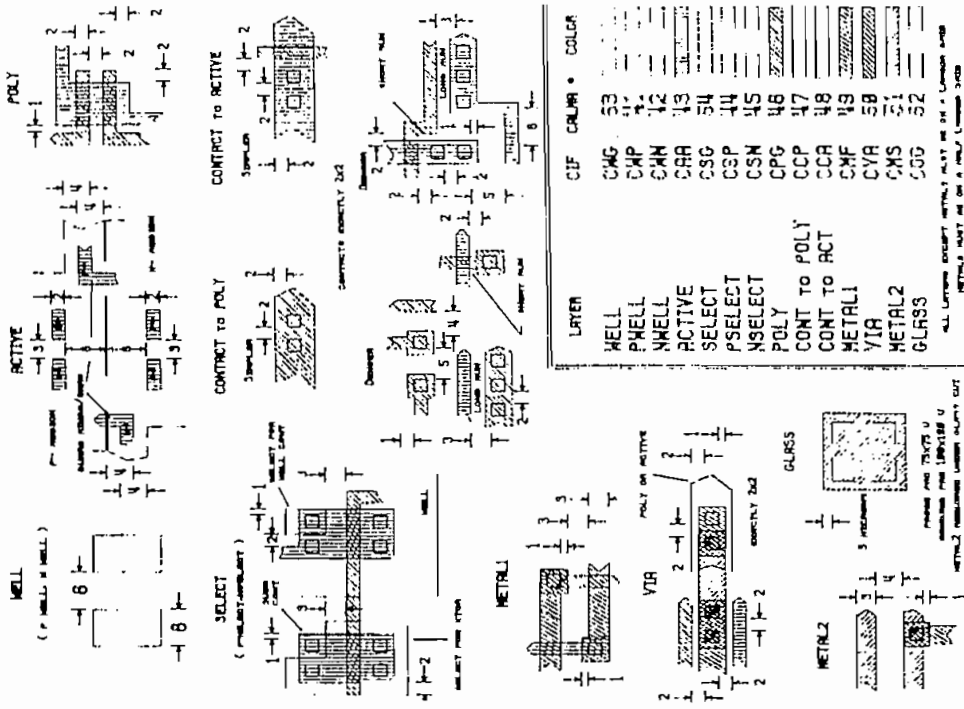
1. C NUMBER : PARAMETER : MIN. DIMENS.(um)

951	BONDING PADS SHALL CONSIST OF METAL 2, VIA, METAL 1	
952	MARGIN OF PASSIVATION OPENING TO METAL 2	5.0
953	MARGIN OF PASSIVATION OPENING TO PAD VIA, MINIMUM AND MAXIMUM VALUE	0.0
954	MARGIN OF PASSIVATION OPENING TO METAL 1	5.0
875	SPACING OF METAL 2 PAD TO UNRELATED METAL 2, METAL 1, POLYSILICON OR ACTIVE AREA	25.0
821	SPACING OF METAL 1 PAD TO UNRELATED METAL 1, METAL 2, POLYSILICON OR ACTIVE AREA	25.0
	NOTE : RULE 753 DOES NOT APPLY FOR BONDING PADS.	



# G.2 REGLAS DE DISEÑO DE LA MOSIS

## MOSIS CMOS SCALABLE RULES



### TECHNOLOGIES AND REQUIRED LAYERS

Process	Technology	Required layers
p well and n subs twin tub	SCP	CWP, CSP
n well and p subs twin tub	SCN	CWN, CSN
All*	SCG	CWG, CSG
All**	SCE	CWP, CWN, CSP, CSN

\*For a p well or n subs twin tub process, MOSIS sets CWP = CWG and CSP = CSG. For an n well or p subs twin tub process, MOSIS sets CWN = CWN and CSN = CSG.  
\*\*For a p well or n subs twin tub process, MOSIS ignores CWN and CSN. For an n well or p subs twin tub process, MOSIS ignores CWP and CSP.

### Questions to Fabricators

#### MOSIS

- merges contact-to-active and contact-to-poly layers to generate the contact mask
- can apply bloats or shrinks to all layers
- can adjust active overlap of contact:independently of bloat to active
- can adjust poly overlap of contact independently of bloat-to-poly

#### FABRICATORS

- What value of lambda can you support? What bloats or shrinks do you require?

# LAYER NAMES AND COLORS

LAYER CIF CALXA \* COLOR

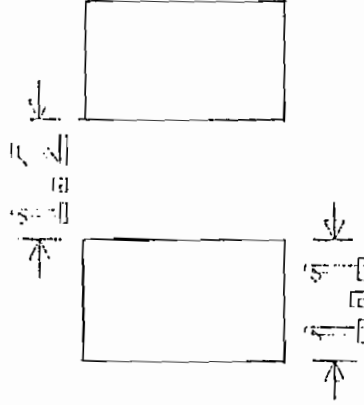
LAYER	CIF	CALXA	* COLOR
WELL	53		
PWELL	41		
NWELL	42		
ACTIVE	43		
SELECT	54		
PSELECT	44		
NSELECT	45		
POLY	46		
CONT TO POLY	47		
CONT TO ACT	48		
METAL1	49		
VIA	50		
METAL2	51		
PLASS	52		

# 1. WELL (NWELL, PWELL)

LANBCHS

1.1 WIDTH

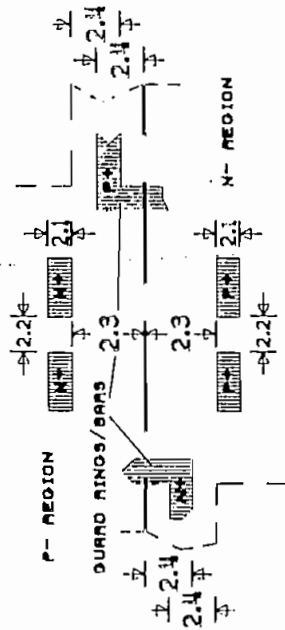
1.2 SPACE



NOTES: IF P AND N WELLS  
SUBMITTED, THEY MAY BE  
OVERLAP BUT THEY MAY BE  
COINCIDENT

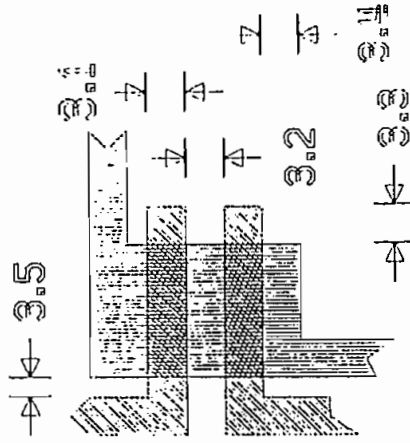
## 2. ACTIVE

	LAMBDAS
2.1 WIDTH	2
2.2 SPACE	3
2.3 SOURCE/DRAIN ACTIVE TO WELL EDGE	6
2.4 GUARD RING/BAR OVERLAP OF WELL	4



## 3. POLY

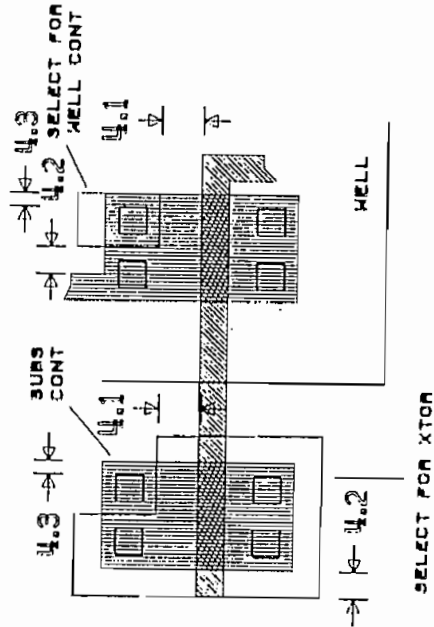
	LAMBDAS
3.1 WIDTH	2
3.2 SPACE	2
3.3 GATE OVERLAP OF ACTIVE	2
3.4 ACTIVE OVERLAP OF GATE	2
3.5 FIELD POLY TO ACTIVE	1



4. SELECT  
(PSELECT, NSELECT)

LAMBDOAS

- 4.1 SELECT SPACE (OVERLAP) TO (OFF) GATE 3
- 4.2 SELECT SPACE (OVERLAP) TO (OFF) ACTIVE 2
- 4.3 SELECT SPACE (OVERLAP) TO (OFF) CONTACT TO WELL OR SUBSTRATE 1

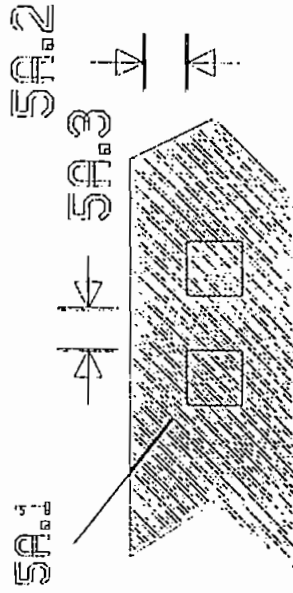


NOTES: IF BOTH PSELECT AND NSELECT SUBMITTED, THEY MAY BE COINCIDENT BUT MUST NOT OVERLAP

5A. SIMPLER CONTACT TO POLY

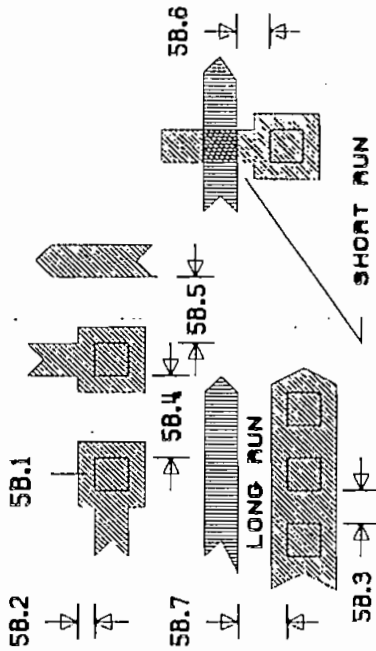
LAMBDOAS

- 5A.1 CONTACT SIZE EXACTLY 2x2
- 5A.2 POLY OVERLAP 2
- 5A.3 SPACING 2



**5B. DENSER CONTACT TO POLY**  
 LAMBODAS  
 2X2

- 5B.1 CONTACT SIZE, EXACTLY 1
- 5B.2 POLY OVERLAP OF CONTACT 2
- 5B.3 SPACING ON SAME POLY 5
- 5B.4 SPACING ON DIFF POLY 4
- 5B.5 SPACE TO OTHER POLY 2
- 5B.6 SPACE TO ACT, SHORT RUN 2
- 5B.7 SPACE TO ACT, LONG RUN 3

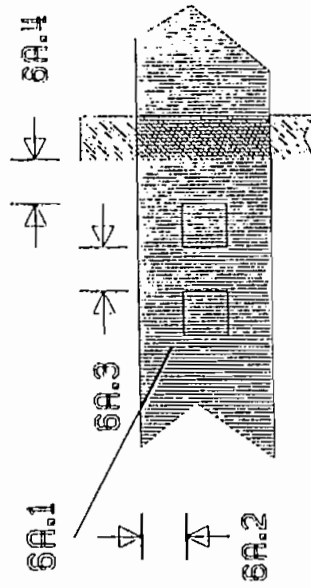


NOTE: YOUR ASSOCIATING CONTACTS WITH POLY OR ACTIVE ALLOWS MOSIS TO INDEPENDENTLY BLOAT THE LAYER AND THE LAYER OVERLAP OF THE CONTACT

**6A. SIMPLER CONTACT TO ACTIVE**

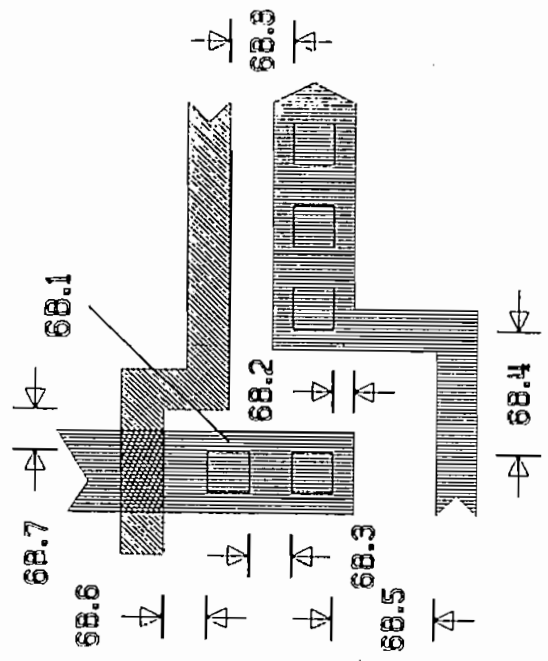
LAMBODAS

- 6A.1 CONTACT SIZE EXACTLY 2X2
- 6A.2 ACTIVE OVERLAP 2
- 6A.3 SPACING 2
- 6A.4 SPACE TO GATE 2



**68. DENSER CONTACT TO ACTIVE**  
LAMBDOAS 2x2

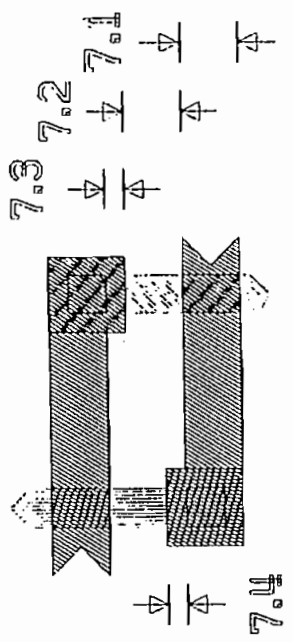
- 68.1 CONTACT SIZE, EXACTLY 1
- 68.2 ACTIVE OVERLAP 2
- 68.3 SPACING ON SAME ACTIVE 6
- 68.4 SPACING ON DIFF ACT 5
- 68.5 SPACE TO GATE 2
- 68.6 SPACE TO FIELD POLY, SHORT RUN 2
- 68.8 SPACE TO FIELD POLY, LONG RUN 3



**7. METAL1**

LAMBDOAS

- 7.1 WIDTH 3
- 7.2 SPACE TO METAL1 3
- 7.3 OVERLAP OF CONTACT TO POLY 1
- 7.4 OVERLAP OF CONTACT TO ACTIVE 1

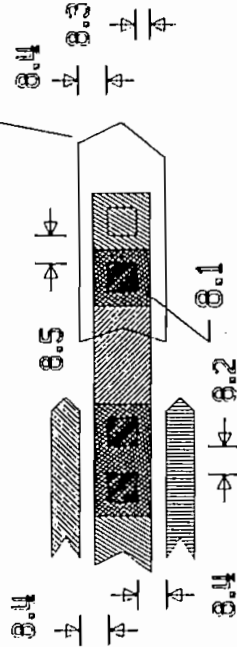


## 8. VIA

LAMBDA

2X2

- 8.1 SIZE, EXACTLY 2
- 8.2 SEPARATION TO VIA 2
- 8.3 OVERLAP BY METAL 1
- 8.4 SPACE TO POLY OR ACTIVE EDGE 2
- 8.5 SPACE TO CONTACT 2



NOTE: OBJECTIVE IS VIA ON A FLAT SURFACE. VIA STACKED OVER CONTACT NOT ALLOWED.

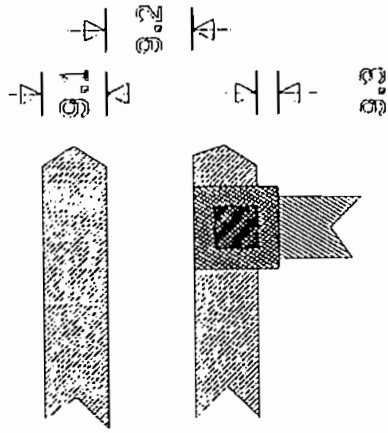
## 9. METAL2

LAMBDA

- 9.1 WIDTH 3

- 9.2 SPACE TO METAL2 1

- 9.3 OVERLAP OF VIA 1





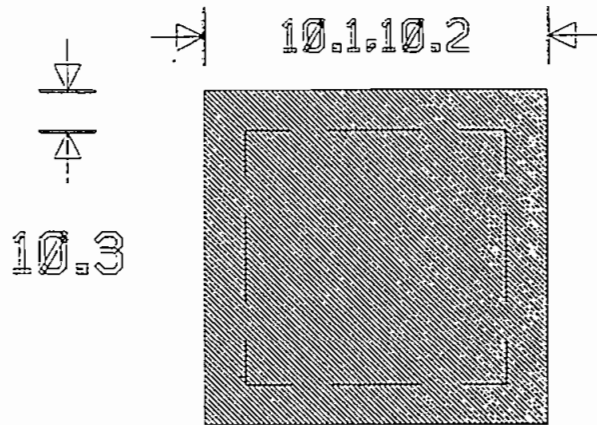
# 10. GLASS

MICRONS

10.1 BONDING PAD 100x100

10.2 PROBE PAD 75x75

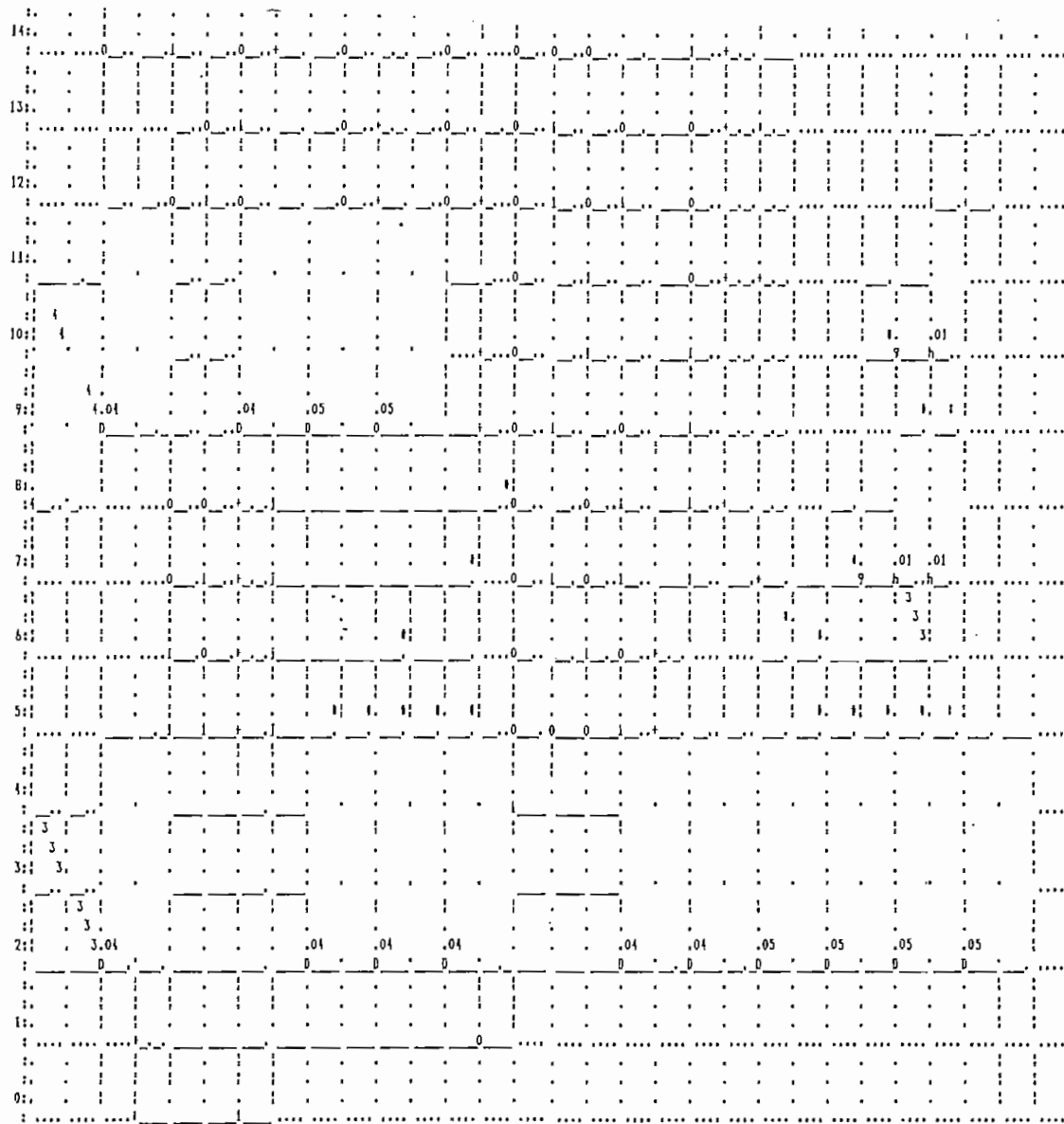
10.3 PAD TO GLASS EDGE 5



NOTE: THERE MUST BE METAL2 UNDER A GLASS CUT

**ANEXO H**

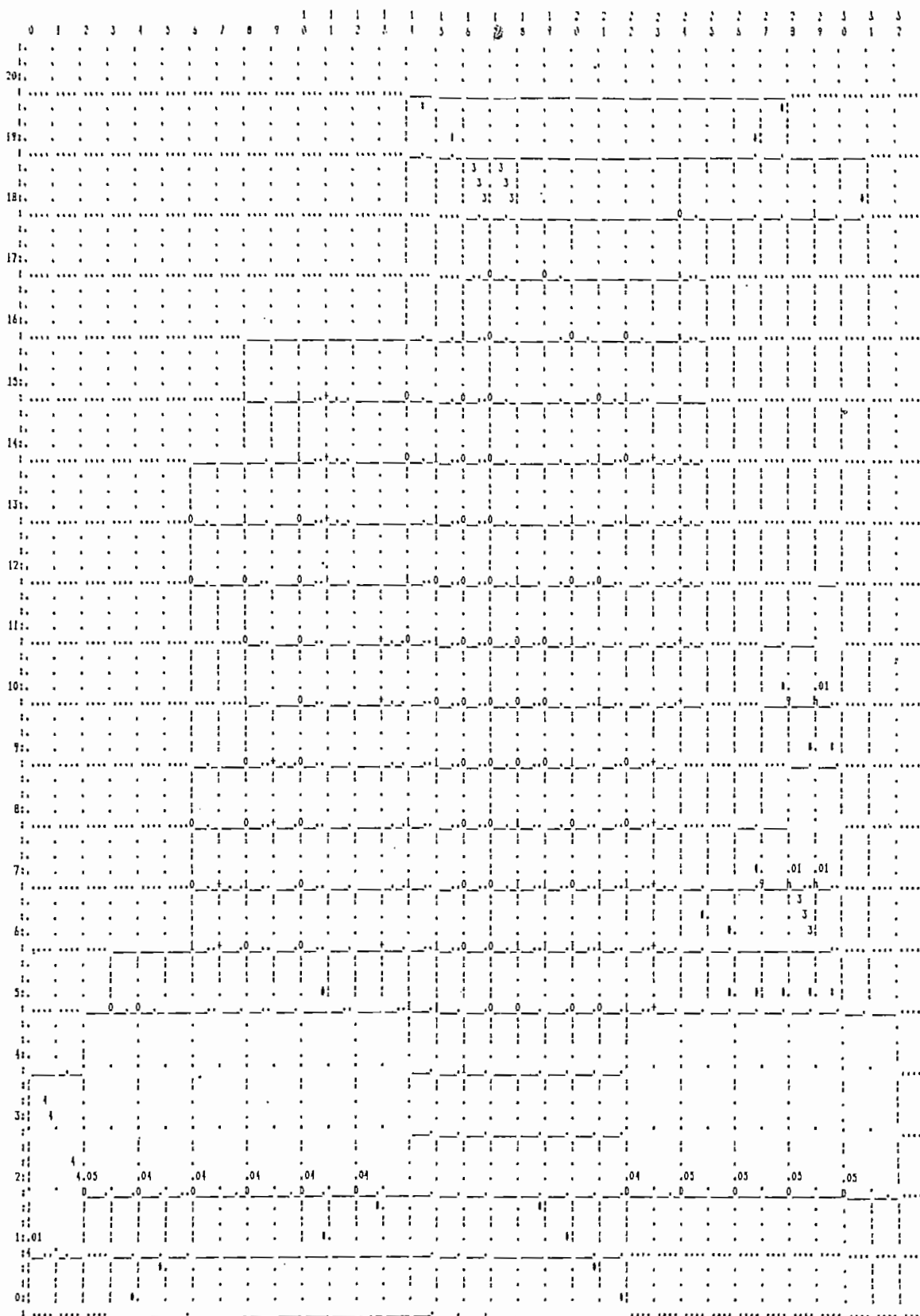
**PLANOS DEL CIRCUITO  
CODIFICADOR DECODIFICADOR HDB<sub>n</sub>**



Node names:

- A10 [10,9,0]
- A11 [17,6,0]
- A12 [15,2,0]
- A13 [14,5,0]
- A14 [15,4,0]
- A20 [17,16,0]
- A21 [17,15,0]
- A22 [13,19,0]
- A23 [11,17,0]
- CLK [8,1,4]

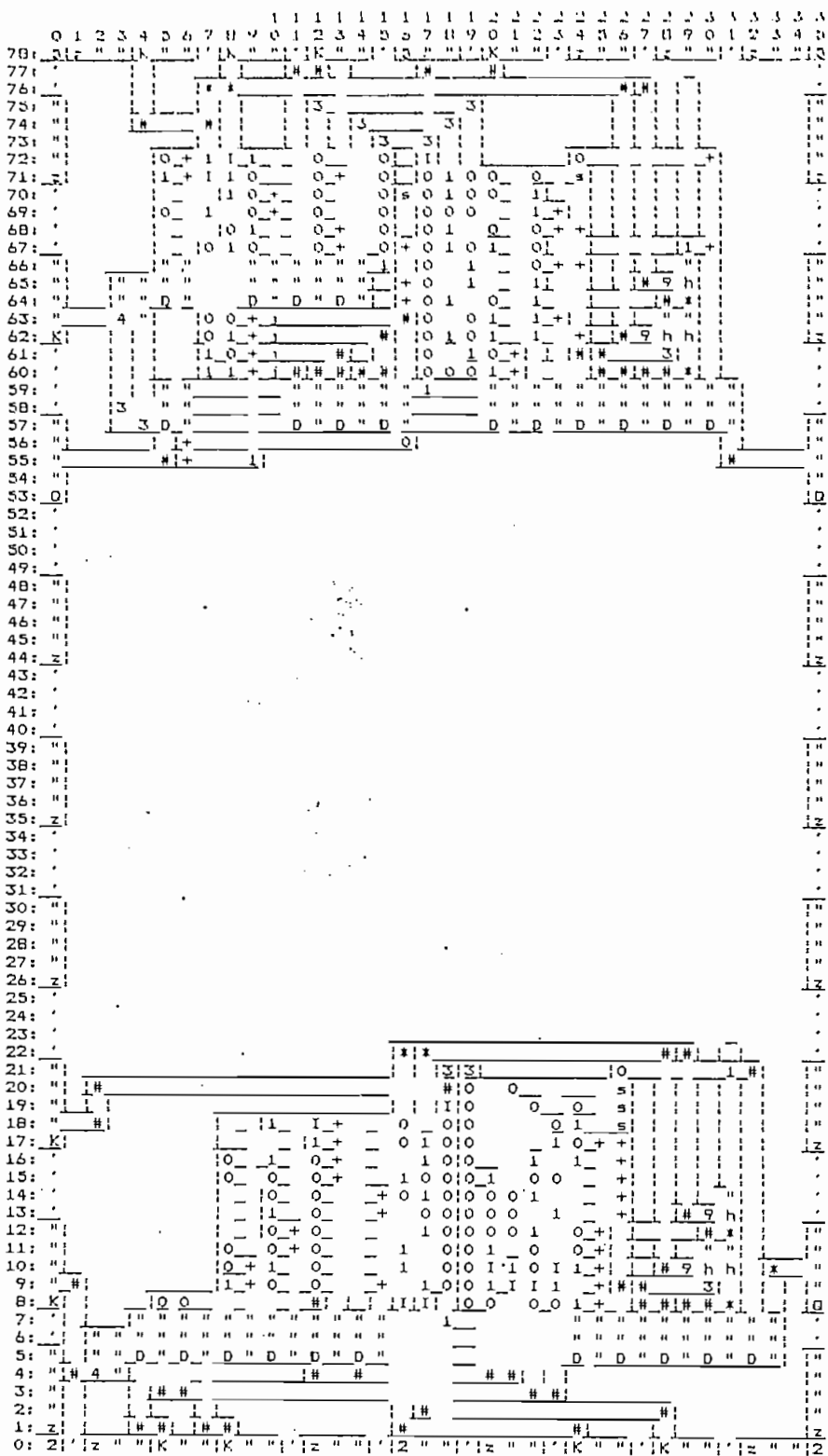
- D10 [15,10,0]
- D11 [13,7,0]
- D12 [15,3,0]
- D20 [17,21,1]
- D21 [14,13,1]
- D22 [12,21,0]
- D23 [11,13,0]
- D24 [12,20,0]
- D25 [6,18,0]
- DATA [12,8,0]
- H0 [20,5,0]
- H1 [20,4,0]
- OUT0 [1,28,0]
- OUT1 [0,2,0]
- START [20,14,0]



code names:

- A10 [13,10,0]
- A11 [5,8,0]
- A12 [14,8,0]
- A13 [12,15,0]
- A14 [10,14,0]
- A20 [5,19,0]
- A21 [11,18,0]
- A22 [12,20,0]
- A23 [13,21,6]
- A24 [14,22,0]
- CLK [1,1,4]

- D10 [11,11,0]
- D11 [9,13,0]
- D12 [5,7,0]
- D13 [7,9,0]
- D20 [18,24,1]
- D21 [9,24,0]
- D22 [13,22,0]
- DATA0 [6,5,6]
- DATA1 [9,2,6]
- M0 [2,15,9]
- M1 [3,14,0]
- WE [1,31,1]
- START [2,16,9]



Node names:  
 CLK\_C [62,0,32]  
 CLK\_D [8,0,32]  
 DATO [78,12,32]  
 HDBnO\_C [78,8,32]  
 HDBnO\_D [0,28,32]  
 HDBnI\_C [78,4,32]  
 HDBnI\_D [0,24,32]  
 IN\_LSB [0,0,32]  
 IN\_MSB [0,5,32]  
 OUT [8,75,32]  
 OUT\_LSB [55,35,32]  
 OUT\_MSB [55,0,32]  
 START\_C [78,29,32]  
 START\_D [17,0,32]