UNIVERSIDAD NACIONAL
AUTONOMA DE MEXICO

FACULTAD DE ESTUDIOS SUPERIORES
CUAUTITLAN

V·N A M

"REDISEÑO DE LA TARJETA DE CONTROL DEL SISTEMA DE LA ESTACION DE REBOMBEO VENTA DE CARPIO (PEMEX)"

T E S I S

QUE PARA OBTENER EL TITULO DE

INGENIERO MECANICO ELECTRICISTA

P R E S E N T A N :

CARLOS ALBERTO NAVA MARTINEZ
AURELIO VELAZQUEZ ROQUE

ASESOR: M. en C. NICOLAS CALVA TAPIA

CUAUTITLAN IZCALLI, EDO. DE MEX. 1992

UNAM – Dirección General de Bibliotecas Tesis Digitales Restricciones de uso

INDICE

PREFACIO.

I.- Generalidades.

I.1. Introducción.

I.2. Aspectos Generales de la Industria Petrolera.

I.3. Panorama General de la Unidad Terminal Programable (PTU).

I.4. Sistemas Digitales.

I.5. Circuitos Lógicos.

I.6. Aspectos Generales de los PLD's.

II.- Dispositivos Lógicos Programables (PLD's).

II.1. Introducción.

II.2. Metodologías de Diseño de Circuitos Integrados.

II.3. Historia de los PLD's.

II.4. Familias, Arquitecturas y Tecnologías.

II.4.1. Notación.

II.4.2. Familias.

II.4.2.1. Memoria de Sólo Lectura Programable (PROM).

II.4.2.2. Arreglo Lógico Programable en Campo (FPLA).

II.4.2.3. Secuenciador Lógico Programable en Campo (FPLS).

II.4.2.4. Arreglo de Compuertas Programable en Campo (FPGA).

II.4.2.5. Decodificador de Direcciones Programable en Campo (FPAD).
II.4.2.6. Lógica de Arreglos Programables (PAL, ZPAL, HAL, ZHAL, NML).
II.4.2.7. Lógica de Arreglo Genérico (GAL).
II.4.2.8. Lógica Programable y Eléctricamente Borrable (PEEL).
II.4.2.9. Dispositivos Lógicos Programables Borrables (EPLD's).
II.4.2.10. Macrológica Programable (PML).
II.4.2.11. Circuito Integrado de Aplicación Específica Programable, Borrable Eléctricamente (ERASIC).
II.4.2.12. Arreglo de Celdas Lógicas (LCA).
II.4.2.13. Arreglo de Compuertas Canalizado Configurable en Escritorio.
II.4.2.14. Familias de Aplicación Específica.
II.4.3. Arquitecturas.
II.4.4. Tecnologías.

III.- Sistemas de Soporte para Diseño con PLD's.
III.1. Introducción.
III.2. Ingeniería Asistida por Computadora (CAE).
III.3. Sistemas de Soporte.
III.3.1. Lenguajes Dependientes de la Manufactura.
III.3.2. Lenguajes Independientes de la Manufactura.
III.4. Soporte de Software Diseñador del Futuro (Future Designer) para Diseño con PLD's.

IV.- Tarjeta de Control del Sistema de la Unidad Terminal Programable (PTU).

IV.1. Introducción.

IV.2. Unidad Terminal Programable (PTU).

IV.3. Tarjeta de Control del Sistema.

V.- Rediseño de la tarjeta de Control del Sistema.

V.1. Introducción.

V.2. Diseño con PLD's.

V.2.1. Alcances y Limitaciones.

V.2.2. Alternativas de Diseño.

V.2.3. Selección del Dispositivo Adecuado.

V.2.4. Determinación de Formas de Diseño.

V.2.5. Otros Aspectos Importantes en el Diseño.

V.3. Proceso de Rediseño.

V.3.1. Procedimiento de Rediseño del Diagrama SYSCON7.

V.3.1.1. Captura Esquemática.

V.3.1.2. Utilización de las Formas de GATES.

V.3.1.3. Programación y Verificación.

VI.- Conclusiones.

VI.1. Conclusiones.

APENDICES

BIBLIOGRAFIA

PREFACIO

Se han hecho diferentes trabajos acerca de sistemas digitales, en la mayoría de las veces siguiendo una metodología tradicional, esto es, basados en dispositivos convencionales (por ejemplo, CI's 74XX) para realizar algún diseño.

Siguiendo la tendencia en cuanto al avance en el desarrollo de dispositivos lógicos, es necesario dar a conocer nuevas metodologías de diseño. Un dispositivo capaz de contener la lógica que el diseñador requiera mediante una adecuada programación, sin sujetarse a una función específica, presenta una alternativa para la implementación de nuevos métodos, tal es el caso de los Dispositivos Lógicos Programables (PLD's).

La investigación que se plantea en esta tesis no pretende ser un trabajo teórico sobre PLD's, tampoco una descripción detallada del funcionamiento de la Unidad Terminal Programable (PTU) ni de la tarjeta de Control del Sistema a rediseñar. Esta investigación propone una aplicación de PLD's mediante una metodología de diseño, basándose en los conceptos y constitución de la tarjeta de Control del Sistema, así como la teoría de PLD's.

CAPITULO I

GENERALIDADES

I.1. INTRODUCCION.

El objetivo de la presente tesis, es mejorar una parte del equipo de supervisión y control de la estación de rebombeo "VENTA DE CARPIO" de PEMEX, llamada Unidad Terminal Programable (PTU), mediante el rediseño de la tarjeta de Control del Sistema, utilizando los Dispositivos Lógicos Programables (PLD's), para lograr: reducción del número de dispositivos con los que actualmente cuenta la tarjeta, minimización de fallas, facilidad de mantenimiento e implementación de circuitos lógicos, utilizando la tecnología más reciente en cuanto a diseño de sistemas digitales se refiere.

El desarrollo del presente trabajo, se basa en el análisis de la tarjeta en su estado original, captura de los diagramas eléctricos, utilizando la parte del soporte de software para PLD's llamada DASH de la compañía DATA I/O, procesamiento de los diagramas, utilizando la parte de soporte de software llamada GATES, del mismo paquete, con el propósito de obtener ecuaciones booleanas, mapa de fusión para PLD's (archivo JEDEC), documentación general de cada diagrama, para posteriormente programar los dispositivos óptimamente seleccionados, revisión, análisis y prueba del rediseño final.

La presente tesis se compone de seis capítulos, los cuales abarcan desde un marco teórico hasta el rediseño de la tarjeta de Control del Sistema. El capítulo uno es una introducción al desarrollo del proyecto, abordando un panorama general de la industria petrolera, conceptos sobre el equipo utilizado en las estaciones de rebombeo, referencias sobre sistemas digitales, definiciones de circuitos lógicos y conceptos generales sobre PLD's.

En el capítulo dos nos referimos a la parte teórica del desarrollo de PLD's. Aquí se hace mención de la historia, arquitecturas, familias y tecnologías de los Dispositivos Lógicos Programables. También se compara la lógica convencional con el desarrollo con PLD's.

En el capítulo tres abordamos el tema de soportes de software para diseño con PLD's, considerando historia, ventajas y desventajas entre cada paquete, resaltando el soporte de software utilizado en la presente tesis.

En el capítulo cuatro se analiza la PTU y las tarjetas que componen a esta estación terminal, así como el funcionamiento de la tarjeta de Control del Sistema, en particular. En esta parte de la tesis, se define la circuitería de la tarjeta en función de dispositivos convencionales y en estado original, esto es, tal y como funciona antes de implementarla con PLD's.

El capítulo quinto abarca el análisis del proceso de rediseño, introduciendo metodología, diagramas, documentación y archivos JEDEC, así como sus ventajas y desventajas en cuanto a costos y mantenimiento.

El capítulo seis se refiere a las conclusiones, aquí hacemos un análisis final de los resultados obtenidos en la tesis, así como una evaluación de objetivos, perspectivas, alcances y limitaciones de la tarjeta final.

Al final de la tesis incluimos algunos apéndices con información complementaria acerca de referencias que consideramos importantes.

Es importante señalar que el soporte de software para diseño con PLD's utilizado en el trabajo de tesis fue instalado, probado, analizado, documentado y aplicado, en el IMP, considerando que este paquete es uno de los primeros instalados en México, dejando un amplio margen de perspectivas para su aplicación posterior.

Cabe también señalar que, dada la poca información, referente al desarrollo de PLD's, nuestro trabajo consistió también, en una búsqueda constante, en un esfuerzo conjunto y con el más firme propósito de dejar una base para generaciones posteriores, en el gran campo de los Dispositivos Lógicos Programables.

I.2. ASPECTOS GENERALES DE LA INDUSTRIA PETROLERA.

A través de los años, el desarrollo de la industria petrolera ha tenido que apoyarse en bases científicas, como la Física, Química, Geografía, Electrónica, etc., todas estas ramas de la ciencia han servido para facilitar la explotación de hidrocarburos.

Los principales pozos de explotación se encuentran en lugares distantes del centro del país, tales como: Tabasco, Campeche, Veracruz, etc.. Para poder trasladar los productos que se obtienen en las diferentes instalaciones de PEMEX, se tiene que hacer uso de estaciones de rebombeo.

Las estaciones de rebombeo se encargan de recibir los hidrocarburos y trasladarlos a los lugares a los que son destinados. Tal es el caso de la estación de VENTA DE CARPIO, en ésta se manejan productos como: crudo, diesel y gasolina. Aquí se cuenta con un Sistema de Adquisición de Datos, el cual monitorea los puntos por donde proviene el combustible, además de los lugares a donde será destinado éste.

I.3. PANORAMA GENERAL DE LA UNIDAD TERMINAL PROGRAMABLE (PTU).

El sistema utilizado en la estación de rebombeo de VENTA DE CARPIO, forma parte de todo un complejo sistema denominado SCADA (Control Supervisor y Adquisición de Datos). Dentro de éste sistema se cuenta con: equipo de cómputo para procesamiento de datos, sensores para tomar la señal desde el fenómeno natural, y equipos de captación de datos que respaldan a los sistemas de cómputo.

Uno de los equipos más utilizados, es la Unidad Terminal Programable (PTU), la cual funciona como una unidad remota de adquisición de datos y control. La PTU se comunica con una estación maestra, misma que recibe la información necesaria del estado actual del sistema, a fin de poder ejecutar un control, para garantizar su funcionamiento.

Las características terminales remotas de la PTU son:

- Capacidades de expansión flexible.
- Construcción modularizada.
- Capacidades de operación sobre un amplio rango de condiciones ambientales.
- Inicialización automática y autoconfiguración.
- Aislamiento óptico para contacto o entradas digitales.

La PTU consta de las siguientes tarjetas:

- Las tarjetas de Control del Sistema y Unidad Aritmética Lógica y Registro General (GRALU), las cuales forman el procesador TRW 2000.
- La(s) tarjeta(s) de memoria.
- La tarjeta del adaptador de canal. Asociada con ésta, la tarjeta terminal E/S de comunicación.
- Las tarjetas lógicas que proveen las funciones básicas de E/S.
- La tarjeta de bus expansor bidireccional (receptor / conductor).

Una descripción más detallada de estas tarjetas será tratada en el capítulo cuatro.

I.4. SISTEMAS DIGITALES.

La electrónica ha revolucionado al mundo en todos los aspectos, desde los más complejos sistemas de Ingeniería o investigación científica, hasta aspectos sociales o psicológicos. Gran parte de la participación activa de la electrónica, es debido a su acelerado desarrollo y mejoramiento de la tecnología empleada, en especial en el desarrollo de la Electrónica Digital.

Hoy en día, en el campo de la Ingeniería en particular, la Electrónica Digital es una pieza clave en el diseño y funcionamiento de diversos sistemas. Es aquí donde entra el

término de Sistemas Digitales, cuya definición e importancia no podríamos encerrar en unas cuantas líneas o párrafos, esto es debido a que los Sistemas Digitales son tan amplios que necesitaríamos incluir desde algún concepto básico hasta la estructura o equipo más complejo.

Sobre los Sistemas Digitales podemos mencionar que maneja elementos discretos de información, tales como impulsos eléctricos. El computador digital de uso general, es el ejemplo más común de un Sistema Digital. Un nombre más adecuado que se le podría designar al computador digital es el de "sistema de procesamiento de información discreta".

La sofisticación de la computadora digital y/o la especialización del equipo para alguna función específica en cualquier campo (científico, técnico, administrativo, etc.) ha dado origen a innumerables Sistemas Digitales. En particular, en los diferentes procesos de la Industria Petrolera, los Sistemas digitales encuentran amplio campo de aplicación.

I.5. CIRCUITOS LOGICOS.

Dentro de los Sistemas Digitales, los circuitos lógicos forman parte esencial en su constitución y funcionamiento. El gran desarrollo que ha tenido el estudio de semiconductores ha originado la mejoría de dispositivos electrónicos, desde los sencillos transistores hasta los circuitos integrados (CI's) de integración a gran escala e integración a muy grande escala(LSI y

VLSI), tales como los dispositivos más versátiles de la serie 74 hasta los más avanzados microprocesadores, reduciendo en gran medida los problemas de operación, consumo de energía, temperatura, tiempo de vida del dispositivo, etc., además de optimizar espacio realmente útil.

Un circuito lógico, es en general un dispositivo o dispositivos electrónicos alambrados con el fin de realizar una tarea específica, para dar como respuesta un dato o datos, previamente concebidos en un diseño lógico. La respuesta depende de las entradas del circuito y el proceso que este realice, pudiendo ser representada por medio de ecuaciones Booleanas.

Con los circuitos lógicos se hace una serie de decisiones necesarias para obtener una respuesta lógica a problemas que tienen un conjunto de condiciones. Las señales son interpretadas en sistema binario, esto es, como un uno o un cero, los cuales representan niveles de voltaje.

Los circuitos lógicos se clasifican en:

a) Combinacionales. Un circuito lógico combinacional consta de variables de entrada, compuertas lógicas y variables de salida. Las variables de salida son determinadas directamente en cualquier momento de la combinación presente de entradas, sin tener en cuenta los estados anteriores de las entradas. En la fig. 1.1 se muestra un diagrama de bloque representativo de un circuito lógico combinacional.

FIG. 1.1.

b) Secuenciales. Los circuitos lógicos secuenciales, constan de un circuito combinacional al cual se le conectan elementos de memoria para formar un camino de realimentación. El circuito secuencial recibe la información binaria de las entradas externas, las cuales, junto con el estado presente de los elementos de memoria determinan el valor binario de las terminales de salida. Además determinan la condición de cambio de estado en los elementos de memoria. Por lo tanto, las salidas de un circuito secuencial no solo dependen de las entradas presentes, sino también de las pasadas y su comportamiento se define por medio de una secuencia de tiempo de las entradas, salidas y estados internos, teniendo la capacidad de ser estos circuitos síncronos o asíncronos. El diagrama a bloque de la fig. 1.2 representa un circuito secuencial.

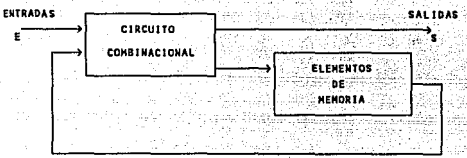


FIG. 1.2.

I.6. ASPECTOS GENERALES DE LOS PLD's.

Dentro de los circuitos lógicos, el concepto más revolucionario son los CI LSI y VLSI, los cuales tienen una alta densidad de compuertas lógicas en un tamaño relativamente pequeño, lo que los hace ser muy eficientes. Los CI's tales como los microprocesadores y memorias son de VLSI, los cuales son escenciales en cualquier Sistema Digital. Sin embargo, además, surge en la actualidad una nueva y poderosa alternativa para diseñar circuitos lógicos, los Dispositivos Lógicos Programables (PLD's). Dentro de los dispositivos lógicos, los PLD's son el más reciente concepto en el diseño de Sistemas Digitales, con un nuevo enfoque en cuanto a la implementación de funciones lógicas se refiere.

La gran ventaja que ofrecen los PLD's, es su capacidad de poder ser programados para obtener una función lógica que se requiera en algún diseño, pudiendo soportar grandes cantidades de lógica combinacional y/o secuencial reducidas a su mínima expresión, logrando la sustitución de dispositivos convencionales de escala de integración pequeña y mediana (SMI y MMI) e incluso dispositivos más sofisticados. De esto podemos observar dos aspectos principales. El primero es que los PLD's son, hoy en día, una opción más en implementación de lógica digital que ofrecen mejorías sobre los CI's convencionales, en ascencia la optimización de espacio realmente útil por compuerta, derivado de la cantidad de CI para obtener alguna función lógica dada, teniendo iguales o mejores condiciones de operación tales como tiempo de respuesta y consumo de energía. El segundo aspecto es que son concebidos como dispositivos de propósito general que no estan sujetos a una función específica y en los cuales, el diseñador puede plasmar la lógica que sólo su imaginación podría limitar. Una vez programados, realizan la función particular implementada que normalmente ha sido verificada a detalle.

Con un gran respaldo en cuanto a variedad de familias y arquitecturas y numerosas herramientas de Ingeniería Asistida por Computadora (CAE) disponibles en el mercado para la creación de circuitos lógicos, los PLD's son utilizados cada vez con más frecuencia en los Sistemas Digitales. En particular, en el Instituto Mexicano del Petróleo se tiene el firme propósito de implementar PLD's en el desarrollo de proyectos de equipos electrónicos, siempre con el fin de utilizar la mejor tecnología (como en todas las áreas del ambiente petrolero) que mantenga a la vanguardia el trabajo realizado en la superación de la Industria del Petróleo.

CAPITULO II

DISPOSITIVOS LOGICOS PROGRAMABLES (PLD's)

II.1. INTRODUCCION.

En la ingeniería de diseño de circuitos electrónicos, existen gran número de herramientas, de las cuales podemos hacer uso y la elección adecuada de éstas, determinará el éxito del proyecto. Hoy en día, una de las tecnologías nuevas con gran futuro de la que disponemos son los Dispositivos Lógicos Programables (PLD's).

Un PLD, es un circuito integrado, que tiene la versatilidad de ser programado, con el fin de obtener de él una gran variedad de funciones lógicas. Con ésta característica, un PLD puede tener diversas aplicaciones, con una programación interna adecuada.

Los dispositivos MOS/CMOS de alta densidad, PROM's, EPROM's, EEPROM's y RAM's entran en el campo de los circuitos integrados de la definición anterior, aunque estos dispositivos no son considerados tradicionalmente como PLD's. Generalmente, estos dispositivos tienen un tiempo de acceso lento, para ser usados como dispositivos lógicos, a exepción de las PROM's, motivo por el cual sólo incluiremos a éstas en el estudio de PLD's.

Los dispositivos LSI de propósito especial, tales como los controladores de interrupción, UART's (Transmisor-Receptor Asincrónico Universal), etc. no son considerados como PLD's debido a que su diseño lógico tiene una sola función particular, que no puede ser alterada, para hacer más versátil al dispositivo. Sin embargo, un PLD, puede programarse para que realice la función (entre una de muchas) de circuito periférico de microprocesadores, sustituyendo en algunos casos a los dispositivos de propósito especial.

II.2. METODOLOGIAS DE DISEÑO DE CIRCUITOS INTEGRADOS.

En la actualidad, es asombrosa la cantidad de dispositivos electrónicos que existen, para el desarrollo de cualquier tipo de sistema y en cualquier campo de aplicación. La necesidad de desarrollar diferentes funciones electrónicas, a originado que la variedad de dispositivos disponibles sea realmente grande. En particular, los circuitos integrados han tenido gran desarrollo en cuanto a su función, arquitectura, encapsulados, características de funcionamiento, etc. La selección adecuada de los circuitos integrados en particular (y de todos los dispositivos electrónicos en general) es determinada por las necesidades que marcan la función que éste vaya a realizar (arquitectura, velocidad de respuesta, consumo de energía, temperatura, etc.). Hoy en día, los diseñadores de lógica digital en específico, pueden seleccionar de entre seis categorías primarias de CI:

- Dispositivos Estandar de Integración a Pequeña Escala/ Integración a Mediana Escala (SSI/MSI).
- Dispositivos Estandar de Integración a Gran Escala/

Integración a Muy Grande Escala (LSI/VLSI).

- Dispositivos de Arreglos de Compuertas.
- Dispositivos de Celdas Estandar.
- Dispositivos "Full-Custom".
- Dispositivos Lógicos Programables.

Sus características son mencionadas en seguida.

DISPOSITIVOS ESTANDAR DE INTEGRACION A PEQUEÑA ESCALA / INTEGRACION A MEDIANA ESCALA (SSI/MSI).

La categoría de los dispositivos estandar de Integración a Pequeña Escala/Integración a Mediana Escala (SSI/MSI) incluyen dispositivos tales como la serie 4000-CMOS y la serie 74XXX TTL/CMOS. Antes de disponer de la técnica de lógica programable, estos dispositivos eran la base de los sistemas digitales.

DISPOSITIVOS ESTANDAR DE INTEGRACION A GRAN ESCALA / INTEGRACION A MUY GRANDE ESCALA (LSI/VLSI).

La categoría de los dispositivos estandar de Integración a Gran Escala/Integración a Muy Grande Escala (LSI/VLSI) incluyen dispositivos tales como microprocesadores y dispositivos periféricos relacionados, los cuales, en especial los microprocesadores, han revolucinado la industria de diseño digital, debido a su gran funcionalidad y costo relativamente bajo.

DISPOSITIVOS DE ARREGLOS DE COMPUERTAS.

Estos dispositivos contienen una cantidad enorme de compuertas. Son generalmente prefabricados con alguna función especial, con sus particulas de silicón completas, excepto por dos ó tres capas finales metalizadas, las cuales proporcionan el alambrado de interconexión para las compuertas, definiendo de este modo las funciones lógicas que serán implementadas en el arreglo. Son llamados también en algunas ocaciones dispositivos "semi-custom", esto es, dispositivos con una aceptable eficiencia en la utilización de sus compuertas.

Los dispositivos de arreglos de compuertas tienen disponibles desde poco menos de 1000 compuertas a 50000 compuertas o más, siendo el caso más común de 10000 compuertas en el dispositivo.

Estos CI son una atractiva alternativa para ser usados en lugar de los dispositivos SSI/MSI ya que las funciones de muchos dispositivos pueden ser convenientemente contenidas en un simple arreglo de compuertas.

DISPOSITIVOS DE CELDAS ESTANDAR.

Los dispositivos de celdas estandar, al igual que los dispositivos de arreglos de compuertas, son dispositivos "semi-custom". Sin embargo, su diferencia radica en que no son prefabricados con una función especial y no contienen gran cantidad de compuertas. En su lugar, un sistema CAD (Diseño Asistido por Computadora) usado en el diseño, proporciona una

"librería de celdas" las cuales definen muchos bloques de funciones de los estandar SSI, MSI y LSI. Una vez diseñada la interconexión de las diferentes "celdas estandar", es generada la máscara apropiada para producir la deseada función en el dispositivo.

DISPOSITIVOS "FULL-CUSTOM".

Estos dispositivos son muy económicos al considerar su gran versatilidad que nos proporciona su funcionamiento. Son generalmente, dispositivos LSI/VLSI, sin que esto signifique que todos los dispositivos LSI/VLSI sean dispositivos "Full-Custom". Microprocesadores, microcontroladores y sus periféricos son algunos ejemplos de estos dispositivos.

Por su alto volumen de producción, son relativamente económicos para construir sistemas digitales, además de ser un CI de extremada eficiencia en cuanto al uso de espacio se refiere.

DISPOSITIVOS LOGICOS PROGRAMABLES.

Los dispositivos lógicos programables, combinan características encontradas en los dispositivos de celdas estandar y los dispositivos de arreglos de compuertas, dando como resultado, un dispositivo de gran versatilidad y utilidad. Son fabricados en grandes volumenes, con arquitecturas estandar, motivo por el cual reduce su costo de producción.

Similar a los dispositivos de arreglos de compuertas y celdas estandar, los PLD's son CI que funcionan de diferente forma, dependiendo de la lógica programada. También, semejante a los arreglos de compuertas, los PLD's pueden reemplazar muchos dispositivos estandar lógicos SSI/MSI, dependiendo de la complejidad del PLD y la lógica a implementar en el dispositivo.

Un PLD ofrece la ventaja sobre un dispositivo de arreglo de compuertas de poder ser usado en pequeños o grandes volumenes de aplicación, ya que existen gran cantidad de arquitecturas que se ajustan al tamaño de la lógica requerida (un sencillo PLD podría reemplazar sólo 4 ó 5 dispositivos SSI).

II.3. HISTORIA DE LOS PLD's.

En tiempos anteriores a la lógica programable, tradicionalmente los diseños de lógica digital consistian de un gran número de circuitos integrados TTL SSI y MSI combinandose entre sí para formar la función lógica deseada. Fue hasta 1975 cuando Signetics Corp. introdujo el primer dispositivo lógico programable que no fuera memoria, el entonces 82S100 (ahora PLS100), Arreglo Lógico Programable en Campo (FPLA)(*), de encapsulado tipo DIP-ancho de 28 terminales. En ese tiempo, Napoleone Cavlan, de Signetics, realizó mejorias en el diseño de sistemas digitales implementando este dispositivo, motivo por lo cual, hoy se le considera "el padre de la lógica programable".

(*) ESTA FAMILIA SE DESCRIBIRA EN LA PARTE II.4.2.2.

Aunque el nuevo dispositivo ofrecia ventajas, no tuvo mucha aceptación entre los diseñadores digitales. Su volumen no solucionaba el problema de economizar espacio, además de tener gran dificultad para diseñar con estos dispositivos debido a que no se contaba con un software de soporte adecuado. Para diseñar y modificar la lógica digital se requeria una carta donde se especificaba la lógica interna por medio de ceros y unos, altos y bajos, lo cual hacia que el trabajo fuera tedioso, prolongado y con alto riesgo de error. El tiempo de respuesta y el costo eran otros factores que no mejoraban significativamente en comparación con los diseños tradicionales.

Posteriormente, John Martin Birkner, trabajando para Computer Automation, estructuró a los dispositivos de Lógica de Arreglos Programables (PAL)(*) de arquitectura similar a los FPLA's. Su diferencia radicó en que mientras los FPLA's contaban con dos arreglos programables (arreglo AND y arreglo OR), los PAL's solo tenían al arreglo AND programable y el arreglo OR era fijo. Por consecuencia, el tamaño de encapsulado se redujo considerablemente, además, el software de soporte con que contaban era mucho mejor. Los primeros dispositivos PAL fueron los ahora populares PAL16L8, PAL16R4, PAL16R6 y PAL16R8.

Debido a que las compañías que empezaron a diseñar con PAL's encontraban problemas de manufactura primaria, tuvieron la necesidad de buscar nuevas alternativas para satisfacer sus requerimientos de producción, surgiendo de esto la familia de dispositivos de Lógica de Arreglo Rigido (HAL)(*).

(*) ESTAS FAMILIAS SE DESCRIBIRAN EN LA PARTE II.4.2.6.

MMI, en donde colaboraban Cavlan y Birkner, perfeccionó el proceso de manufactura de los PAL's y en 1978 publicó el primer manual de PAL's, donde además se explicaba el concepto y beneficios que ofrecian. También incluia un listado fuente en FORTRAN del paquete PALASM(*), el cual compilaba ecuaciones lógicas en un formato que podía ser transmitido a un programador de PAL's. Además simulaba y verificaba la programación del dispositivo.

En 1977, Signetics introdujo el 82S103 Arreglo de Compuertas Programable en Campo (FPGA)(**) y en 1979 introdujo el 82S105 Secuenciador Lógico Programable en Campo (FPLS)(***), este último conteniendo flip-flops en su arquitectura interna.

En lo referente a software de apoyo para diseño con PLD's, Signetics creó el compilador para PLD's AMAZE(*).

En septiembre de 1983, Assisted Technology produjo la versión 1.01 de su compilador de PLD's Compilador Universal para Lógica Programable (CUPL)(*), soportando 29 dispositivos.

En 1984, Data I/O (líder de fabricación de equipos de programación de dispositivos EPROM's, PROM,s y PLD's) introdujo su compilador de PLD's ABEL(*), con caracteristicas similares al

(*) PARA MAYOR INFORMACION ACERCA DE SISTEMAS DE SOPORTE DE SOFTWARE PARA PLD's, REFERIRSE AL CAPITULO III.

(**) ESTA FAMILIA SE DESCRIBIRA EN LA PARTE II.4.2.4.

(***) ESTA FAMILIA SE DESCRIBIRA EN LA PARTE II.4.2.3.

compilador CUPL, aunque con mayor respaldo financiero de mercado, es aún hoy en día, líder en unidades vendidas, dejando muy atrás a los compiladores PALASM y AMAZE de MMI y Signetics respectivamente.

En febrero de 1984, Xilin desarrolló una familia de arreglos de compuertas programables. A diferencia de los PLD's existentes, los Arreglo de Celdas Lógicas (LCA)(*) de Xilin no contenían términos producto y términos suma. Consistían básicamente de una matriz de Bloques Lógicos Configurables (CLB's) rodeada de Bloques de Entrada/Salida (IOB's). Los LCA's fueron los primeros PLD's que usaron celdas RAM estáticas (SRAM).

En julio de 1984 tuvo lugar un avance significativo de PLD's. Altera Corp. introdujo al mercado el EPJ00 el cual implementaba el uso de tecnología CMOS/EPROM, que como sabemos, tienen la capacidad de ser borrados con luz ultravioleta. Estos dispositivos fueron llamados por Altera Dispositivos Lógicos Programables Borrables (EPLD's)(**).

Lattice Semiconductor Corp. introdujo en 1985 una nueva familia de PLD's, llamados Lógica de Arreglo Genérico (GAL)(***), usando tecnologías CMOS/EEPROM, la cual permite borrar eléctricamente al dispositivo, siendo este proceso más rápido que con luz ultravioleta. El término asignado a estos dispositivos fue el de PLD's Borrable Eléctricamente (EEPLD's).

(*) ESTA FAMILIA SE DESCRIBIRA EN LA PARTE II.4.2.12.

(**) ESTA FAMILIA SE DESCRIBIRA EN LA PARTE II.4.2.9.

(***) ESTA FAMILIA SE DESCRIBIRA EN LA PARTE II.4.2.7.

En 1985, Signetics introdujo el concepto Macro Lógica Programable (PML)(*). El primer PML fue el PLHS501 Unidad Lógica Aleatoria, con tecnología bipolar. Una familia de arquitectura parecida a los PML's la introdujo Exel Microelectronic en 1987, llamada Circuito Integrado de Aplicación Específica Programable, Borrable Eléctricamente (ERASIC)(**).

Un curso nuevo tomaron los PLD's en 1986 cuando se implementaron Aplicaciones Específicas de PLD's (ASPLD's)(***), siendo programados de fabricación. Intel, Altera, AMD, MMI, TI entre otros implementaron este tipo de dispositivos.

En 1988 ocurrieron dos inovaciones en el mercado de PLD's. La primera fue que Actel introdujo una nueva serie de areglos de compuertas programables, llamados "programable una sola vez" (OTP) que elimina la necesidad de configurar el dispositivo cada vez que fuera energizado. La segunda inovación tuvo lugar cuando Gazelle Microcircuits anunció el primer PLD basado en tecnología de Arseniuro de Galio (GaAs) que proporcionaba una rápida energización del dispositivo.

Como podemos observar, la acelerada y reciente historia del desarrollo de la industria de PLD's tuvo lugar, en su mayor parte, en el transcurso de la década de los 80's, sobre todo en su primera mitad. Alrededor de esos años, en igual forma, aparecieron en el mercado libros, manuales, software y hardware en gran

(*) ESTA FAMILIA SE DESCRIBIRA EN LA PARTE II.4.2.10.

(**) ESTA FAMILIA SE DESCRIBIRA EN LA PARTE II.4.2.11.

(***) ESTA FAMILIA SE DESCRIBIRA EN LA PARTE II.4.2.14.

cantidad y variedad para soportar el diseño con PLD's. También se implementaron cursos de diseño con PLD's en Universidades e Industrias que ayudaron a comercializar a los PLD's.

La industria de PLD's, procuró no cometer el mismo error que se tuvo en el desarrollo de EPROM's, consistente en no contar con un formato de archivos-objeto estandar para programar EPROM's. Tomando en cuenta este problema se creó el "Joint Electron Device Engineering Council" (JEDEC) que es un archivo estandar compilado de salida para PLD's, capaz de ser usado por todos los actuales y futuros fabricantes de PLD's. En octubre de 1983 el "Consejo de Productos de Ingeniería de Estado Solido JEDEC" publicó el estandar JEDEC No.3 "Formato de Transferencia Estandar entre Sistemas de Preparación de Datos y Programador de PLD's". En Mayo de 1986 tuvo una nueva revisión el formato dando como resultado el estandar JEDEC No.3-A. Para beneficio de todos, este estandar fue aceptado universalmente por toda la industria de PLD's.

En la actualidad sigue incrementandose el uso de PLD's en el diseño de lógica digital en el mundo y particularmente, en México empieza a tomar importancia la implementación de PLD's en Institutos de Investigación y Universidades de Educación Superior. Con los antecedentes históricos que respaldan a los PLD's, se vislumbra un gran futuro y desarrollo en el campo de la Electrónica Digital.

II.4. FAMILIAS, ARQUITECTURAS Y TECNOLOGIAS.

El propósito de esta sección es mostrar de manera general las familias, arquitecturas y tecnologías disponibles de PLD's. Consideramos esto importante debido a que es necesario conocer los dispositivos con los que podríamos contar en el diseño, así como su arquitectura interna para sacar el mayor provecho posible del dispositivo.

Un motivo por lo que los PLD's han tenido gran auje en la implementación de diseños lógicos, es que existen muchas familias, arquitecturas y tecnologías disponibles que permiten al diseñador seleccionar el dispositivo más óptimo que se adapte a las necesidades del diseño, en cuanto a complejidad del circuito a implementar, velocidad, consumo de energía y reprogramabilidad.

Puede existir confusión entre los términos "familias" y "arquitecturas" de PLD's ya que en realidad existen gran cantidad de ambas y en algunos casos con un nombre común. Una familia puede tener varias subfamilias y/o multiples arquitecturas y a su vez, familias y arquitecturas, pueden tener diferentes tecnologías de fabricación (tal como bipolar, CMOS, etc.). Por ejemplo, un dispositivo de Lógica de Arreglos Programables (PAL) representa ambos términos, "arquitectura" (teniendo un arreglo AND programable y un arreglo OR fijo) y "familia" (nombre genérico que los fabricantes adoptaron para identificar a los dispositivos con la arquitectura mencionada), existiendo muchos dispositivos con este nombre con algunas variaciones en sus configuración interna. Similarmente, las familias FPLA y FPLS representan arquitecturas especificas y son ellas mismas familias dentro de la familia de PLS's.

La fabricación cada vez mayor de PLD's, con arquitecturas cada vez más complejas, ha dado motivo para que los fabricantes le den a sus dispositivos nombres diferentes a los de sus competidores, generandose con esto una gran número de familias. Lógicamente los códigos que se les asignan a los dispositivos para identificar sus características de velocidad, temperatura, encapsulado, etc., también varía significativamente pudiendo ocacionar algunas confusiones.

En las siguientes páginas serán descritas las familias de PLD's en la forma más aceptada generalmente, mencionando algunas características propias de su arquitectura. Posteriormente se hará una breve mención de las arquitecturas y tecnologías que actualmente podemos encontrar en PLD's, haciendo énfasis en las arquitecturas utilizadas en el rediseño de la tarjeta de Control del Sistema.

Sin embargo, previamente, es necesario conocer las nomenclaturas y notaciones comunes que se usan en la descripción de las arquitecturas de los PLD's. Por tal motivo, a continuación se presenta una pequeña sección que trata la notación común en los diagramas eléctricos de PLD's.

II.4.1. NOTACION.

Debido a la gran cantidad de entradas y compuertas que poseen internamente los PLD's, la industria fabricante de estos dispositivos ha adoptado una notación estandar para la fácil representación de las arquitecturas de los PLD's.

La fig. 2.1 muestra una compuerta AND de 16 entradas con una notación convencional.

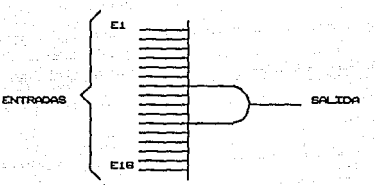


FIG. 2.1

La misma compuerta AND de 16 entradas es mostrada en la fig. 2.2 con una notación usada para PLD's. Una sola línea horizontal es intersectada por 16 líneas de entrada verticales. La línea horizontal no indica que las 16 líneas son conectadas juntas. Esto sólo indica que la línea de intersección es una entrada de la compuerta. Además, el punto de intersección es un fusible, donde una "X" en la intersección indica una conexión por medio de éste. Un PLD nuevo tiene todos sus fusibles intactos, o sea, todas sus intersecciónes muestran una "X".

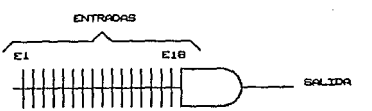


FIG. 2.2

Por ejemplo, si queremos implementar la lógica mostrada en la fig. 2.3 en un PLD que tenga una arquitectura como la mostrada en la fig. 2.4, la conexión apropiada sería la mostrada en la fig. 2.5; las "X" representan las conexiones lógicas.

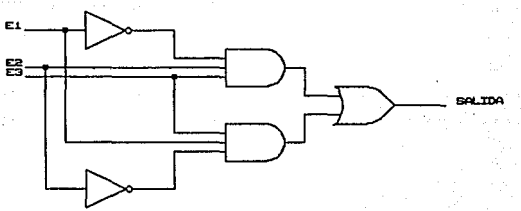


FIG. 2.3

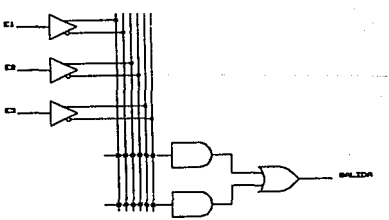


FIG. 2.4

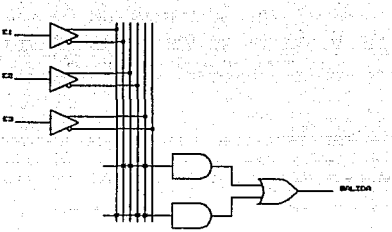


FIG.2.5

Una vez especificada la notación convencional de los PLD's, podremos entender con más facilidad sus familias y arquitecturas que a continuación se describen.

II.4.2. FAMILIAS

II.4.2.1. MEMORIA DE SOLO LECTURA PROGRAMABLE (PROM).

Las Memorias de Sólo Lectura (PROM's), fueron los primeros dispositivos usados como PLD's. Su arquitectura es muy simple. Consiste de un arreglo de celdas de memoria con líneas de direcciones para entradas y líneas de datos para salidas, las cuales, juntas indican la matriz de la PROM.

Las PROM's consisten de un arreglo AND fijo seguido por un arreglo OR programable, mostrado en la fig. 2.6, ilustrando una PROM de 16 x 4. Las "X's" indican intersecciones programables (fusibles); los puntos indican conexiones fijas. Puesto que el arreglo AND es fijo, uno de los 16 términos producto es seleccionado en base a la lógica patrón presentada.

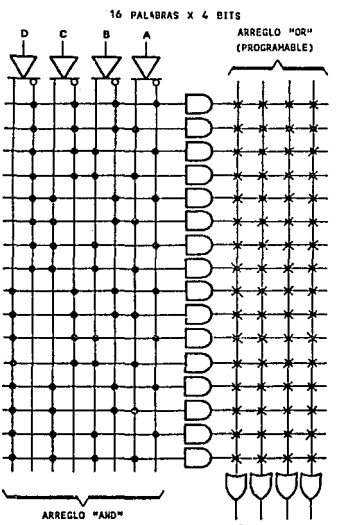


FIG. 2.6

Las 16 salidas de los términos producto se conectan a cada uno de los cuatro términos suma (OR) de salida. Al remover algún fusible de unión, el respectivo término producto de salida es eliminado de el respectivo término suma.

El término suma de salida será lógicamente alto cuando el término producto con el fusible intacto sea direccionado y será lógicamente bajo cuando los términos producto con fusibles removidos son direccionados. La fig. 2.7 ilustra una PROM programada con su respectiva lógica.

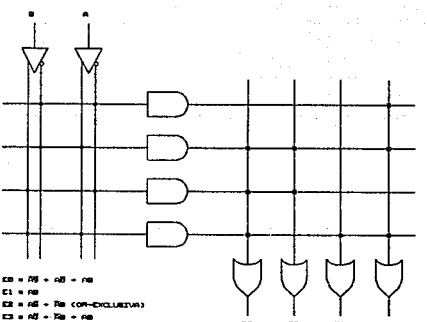


FIG. 2.7

Existen también las PROM con salidas registradas, donde las salidas de los términos suma no son transferidas a las terminales de salida hasta que una señal de reloj de sincronización se recibe.

Debido a la importancia que tiene la velocidad de acceso para utilizar una PROM como dispositivo lógico (típicamente se requiere 60 ns de tiempo de acceso) se ha usado tecnologías bipolares de alta velocidad y recientemente tecnologías CMOS de alta velocidad permitiendo el desarrollo de las EPROM's (PROM's Borrables) con tiempos de acceso de 55 ns o incluso hasta de 25 ns.

MMI fabricó una variante de PROM's, a las cuales llamó Elementos Lógicos Programables (PLE's), que son simplemente considerados como una familia de PROM's desde el punto de vista de diseño.

II.4.2.2. ARREGLO LOGICO PROGRAMABLE EN CAMPO (FPLA).

Fue el primer PLD que no pertenecía a la familia de las PROM's. Su arquitectura básica consiste de un arreglo AND programable seguido por un arreglo OR programable, como se muestra en la fig. 2.8. La salida del término producto alimenta al arreglo OR programable, que permite que cualquiera de los términos producto sea conectado a cualquiera de los términos suma. Un simple término producto puede conectarse a varios términos suma.

La característica de tener además un arreglo AND programable adiciona considerable flexibilidad a los FPLA's, sobre todo para usar el número exacto de términos producto necesarios para cada salida.

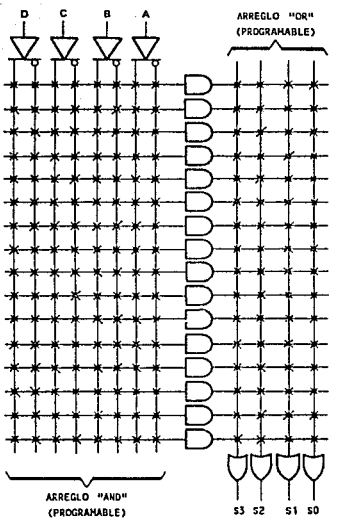


FIG. 2.8

Algunas desventajas que presentan los FPLA's son que no existe suficiente softwarte de soporte de diseño, relativa respuesta lenta, costo relativamente alto además de ser un CI de tamaño grande, esto último debido a que los arreglos AND y OR ocupan gran espacio por su arquitectura programable.

Los FPLA's son frecuentemente descritos en términos de la configuración de su matriz. Por ejemplo el PLS153 es descrito como un 18 x 42 x 10 FPLA, indicando que tiene 18 entradas en el arreglo AND, 42 términos producto y 10 salidas.

Mientras que Signetics, así como otros fabricantes, tiene una familia de dispositivos FPLA's, el FPLA también representa una arquitectura que es presentada en otras familias de PLD's.

II.4.2.3. SECUENCIADOR LOGICO PROGRAMABLE EN CAMPO (FPLS).

Esta famila es muy parecida a la familia de los FPLA's en cuanto a su arquitectura. Algunos sólo tienen salidas registradas con realimentación y otros además tienen registros que pueden usarse como entradas o salidas exteriormente[7].

II.4.2.4. ARREGLO DE COMPUERTAS PROGRAMABLE EN CAMPO (FPGA).

El concepto básico de estos dispositivos es mostrado en la fig. 2.9. Contiene un arreglo AND programable con salidas de polaridad programable. Puesto que una compuerta AND puede implementarse fácilmente en una compuerta NAND (invirtiendo la salida), una compuerta NOR (invirtiendo todas las entradas) o una compuerta OR (invirtiendo todas las entradas y salidas), cada una de las compuertas AND del FPGA puede convertirse en cualquiera de las compuertas anteriores. También pueden ser fácilmente implementadas como un inversor. Las entradas de las compuertas AND son fácilmente invertidas, ya que el estado verdadero y su complemento (negación) de cada señal de entrada es presente en el arreglo AND del dispositivo.

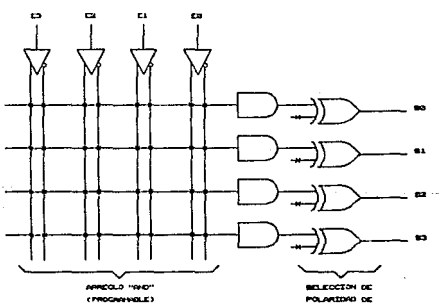


FIG. 2.9

II.4.2.5. DECODIFICADOR DE DIRECCIONES PROGRAMABLE EN CAMPO (FPAD).

Estos dispositivos son idénticos en cuanto a su función y arquitectura a los FPGA's[7].

II.4.2.6. LOGICA DE ARREGLOS PROGRAMABLES (PAL, ΣPAL, HAL, ΣHAL, MML).

La familia de PAL's se ha convertido en la familia de PLD's más popular de todas las que se disponen. Consiste de un arreglo AND programable seguido por un arreglo OR fijo. Fue creada para superar algunos inconvenientes presentados por los FPLA's. La fig. 2.10 muestra la arquitectura básica del PAL.

Las ventajas que tiene sobre los FPLA's son que ocupan menos área real de material semiconductor debido a que presinden del arreglo OR programable, lo cual influye directamente en el costo. También disminuye el tiempo de respuesta considerablemente.

Una característica que estableció primeramente la familia de los PAL's y despues la adoptaron otras familias de PLD's, es que cuenta con un fusible de seguridad, que si es programado, inhibe la verificación y hace casi imposible determinar la lógica que contenga el dispositivo, de esta manera el PAL es protegido contra copia.

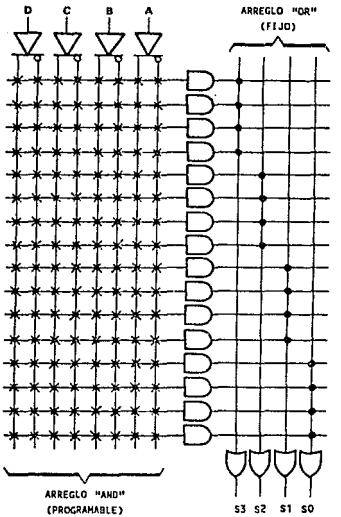


FIG. 2.10

Existen arquitecturas de PAL's que utilizan "macroceldas programables" (tal como el PAL22V10) que proporciona una alternativa más a los diseñadores que buscan un dispositivo aún más versatil en la configuración de sus salidas, ya sean combinacionales o registradas o ser usadas bidireccionalmente.

Los ZPAL son PAL's llamados de "energía cero" ("zero power"); construidos con tecnología CMOS, tienen baja corriente de operación, requiriendo no más de 100μA, incluso algunos no necesitan más de 10μA.

Algunos fabricantes han implementado a los PAL's de máscara programable, llamados por la compañía MMI como Lógica de Arreglo Fijo (HAL) y HAL de Energía Cero (ZHAL), ambos, versiones de "máscara programable" de los estandar PAL y ZPAL respectivamente. National Semiconductor también fabrica estos dispositivos y los ha llamado dispositivos de Lógica Mascarada de National (NML).

II.4.2.7. LOGICA DE ARREGLO GENERICO (GAL).

La familia de los GAL[7] es un grupo de PLD's Borrables Eléctricamente (EEPLD's) que tienen una arquitectura genérica de macroceldas programables, tal como los PAL22V10. Lattice los llama como Macroceldas Lógicas de Salida (OLM's).

II.4.2.8. LOGICA PROGRAMABLE Y ELECTRICAMENTE BORRABLE (PEEL).

La familia de los PEEL utilizan tecnología EEPROM y son arquitectónicamente muy similares a los GAL.

Utilizan macroceldas programables que permiten salidas registradas o combinacionales, con polaridad de salida programable.

II.4.2.9. DISPOSITIVOS LOGICOS PROGRAMABLES BORRABLES (EPLD's).

El término EPLD en un principio se utilizó para identificar a una familia de PLD's fabricados por Altera y posteriormente se convirtió en un término general en la industria de PLD's para identificar a los dispositivos borrables con luz ultravioleta que utilizan tecnología de celdas EPROM.

Su tecnología está basada en macroceldas programables así como un arreglo AND programable y un arreglo OR fijo. Tienen salidas registradas y combinacionales y tienen terminales que pueden ser configuradas como entradas o salidas, además de poder programar el estado activo de sus terminales. Algunos poseen terminales para suministrar señales de reloj a los registros de macroceldas.

Aún dentro de la familia de los EPLD's existen varias diferencias en cuanto a sus arquitecturas, sin embargo, estas no son muy significativas, ya que, aunque cada fabricante introduce varias modificaciones internas para sus diversos dispositivos EPLD's, en términos generales la arquitectura escencial es la misma.

Debido a la versatilidad que presenta, ésta familia fue la seleccionada para llevar a cabo el rediseño de la tarjeta de Control del Sistema . Los EPLD's utilizados fueron el EP610(*), EP910(*) y el CY 7C331(**). Los dos primeros tienen la misma arquitectura, sólo varían en su tamaño y capacidad. Cuentan con terminales bidireccionales programables que facilitan la designación del número de entradas/salidas en las particiones lógicas del rediseño. Usan tecnología CMOS EPROM basada en macroceldas programables. En estos dispositivos se implementó la lógica para salidas combinacionales. Por otro lado, el CY 7C331, debido a su arquitectura y a su tipo de registro, fue el adecuado para utilizarlo en la programación de particiones con salidas registradas.

II.4.2.10. MACRO LOGICA PROGRAMABLE (PML).

Los PML's son una familia de dispositivos de Signetics, que utiliza tecnología bipolar, con una arquitectura de Un Sólo Plano de Multinivel (MLSP), también llamada Lógica de Pliegues, la cual usa un simple arreglo NAND o NOR en conjunto con una estructura central interconectada programable, permitiendo implementar lógicas de multiple nivel, en adición para conectarse con entradas o salidas "macros", o sea, bloques funcionales.

(*) REFERIRSE AL APENDICE B Y A LA CITA BIBLIOGRAFICA No. (5) PARA MAYOR INFORMACION.

(**) REFERIRSE AL APENDICE B Y A LA CITA BIBLIOGRAFICA No. (12) PARA MAYOR INFORMACION.

II.4.2.11. CIRCUITO INTEGRADO DE APLICACION ESPECIFICA PROGRAMABLE, BORRABLE ELECTRICAMENTE (ERASIC).

La familia de los Circuitos Integrados de Aplicación Específica Programables, Borrables Eléctricamente (ERASIC's) son exclusivos de Exel Microelectronics. Tienen una arquitectura similar a la de la familia de PML's. La diferencia radica en la tecnología utilizada ya que los ERASIC emplean tecnología CMOS EEPROM (por lo que son considerados EEPLD's) mientras que los PML's utilizan tecnología bipolar.

II.4.2.12. ARREGLO DE CELDAS LOGICAS (LCA).

Esta familia fue introducida por Xillin y por MMI. Presenta una característica única dentro de las familias de los PLD's en cuanto a arquitectura se refiere, ya que utiliza celdas de almacenamiento de RAM Estática (SRAM) para configurar sus funciones lógicas, a diferencia de las técnicas tradicionales de la programación de arreglos AND/OR. La estructura y función básica es el programar Bloques de Lógica Configurable (CLB).

II.4.2.13. ARREGLO DE COMPUERTAS CANALIZADO CONFIGURABLE EN ESCRITORIO.

Estos dispositivos introducidos por Altera Corp. consisten de un arreglo de módulos lógicos configurables que se interconectan para implementar diseños lógicos, similar a la estructura interna

interna de los LCA's, sólo que estos dispositivos no utilizan celdas de memoria SRAM volátiles.

II.4.2.14. FAMILIAS DE APLICACION ESPECIFICA.

Estas familias surgieron al observarse el frecuente uso de PLD's en aplicaciones comunes en diseños lógicos. Debido a esto se implementaron dispositivos con diseños programados de fabrica para proporcionar un recurso más a los diseñadores en sus proyectos digitales.

Existen básicamente dos áreas primarias de arquitecturas de aplicaciones específicas: Aplicaciones de Máquinas de Estado y Aplicaciones de Interfaces de Bus, con una gran cantidad de dispositivos cada una de ellas[7].

II.4.3. ARQUITECTURAS.

Se han mencionado las características de funcionamiento de algunas arquitecturas existentes dentro de las familias de los PLD's con el fin de poder identificar las diferencias que existen entre las familias mencionadas. Resumiendo, podemos clasificar básicamnente a los PLD's en seis tipos de arquitecturas:

1.- Arquitectura PROM/PLE. Incluye a los dispositivos que utilizan principalmente a un arreglo AND fijo y a un arreglo OR programable con o sin registros.

2.- Arquitectura PLS/PLA. Incluye a los dispositivos que básicamente utilizan a un arreglo AND programable seguido por un arreglo OR también programable con o sin flip-flops en sus salidas para ser configurados como dispositivos con salidas combinacionales o registradas. Los dispositivos característicos de esta arquitectura son los pertenecientes a la familia PLS de Signetics.

3.- Arquitectura PAL. Esta arquitectura se refiere a los dispositivos que constan de un arreglo AND programable seguido de un arreglo OR fijo con o sin salidas combinacionales y/o registradas. Los dispositivos más representativos que usan esta arquitectura son los PAL's de MMI.

4.- Arquitecturas convencionales mejoradas. Son aquéllas arquitecturas en las que se han mejorado la versatilidad de los planos AND/OR con la incrustación de macroceldas de entrada/salida configurables dandoles un potencial de aplicación mayor. Los dispositivos que basan su funcionamiento en esta arquitectura son los de las familias de los GAL's, PEEL's y EPLD's.

5.- Arquitecturas ortodoxas. Estos dispositivos usan a los planos AND/OR conceptualmente diferente que las otras arquitecturas, con bloques funcionalmente complejos. Algunas veces son utilizados planos NAND/NOR. Algunas familias representativas de estas arquitecturas son la ERASIC de Exel Microelectronics y la

familia de los PML's.

6.- Arquitecturas híbridas. Estas arquitecturas combinan las arquitecturas convencionales de PLD's (PROM/PLE, PLS/PLA y PAL) con funciones estandar específicas (bloques de memoria EPROM, contadores, decodificadores, etc.). A su vez, esta arquitectura se puede clasificar de la siguiente manera:

a) Secuenciadores Programables.- Entre éstos se encuentran los EPS448 de Altera, AmPAL23S8 de AMD, TIBPSG507 de TI, etc.
b) Dispositivos orientados a ductos.- Entre éstos se encuentran los 5CBIC de Intel, EPB1400 de Altera, PLX448 de PLX Technology, etc.
c) Aplicaciones de Mapeo.- Se encuentran entre éstos los 82C339 de Harris, MAP168 y MAP30 de Wafer Scale Integration, etc.
d) Interruptores DIP Programables.- Algunos dispositivos de éstos son los DS1290 y DS1291 de Dallas Semiconductor.

II.4.4.- TECNOLOGIAS.

En la fabricación de los PLD's, como en todos los circuitos integrados, la forma de como elaborarlos puede ser de diferente manera, dependiendo de la tecnología usada que proporcionará las características deseadas del dispositivo. Un PLD puede ser manufacturado con diferente tecnología, independientemente de la arquitectura interna que tenga, sin afectar la función propiamente lógica con la que halla sido diseñado. Así pues, dispositivos con

diferente tecnología pueden proporcionar la misma función lógica, la diferencia radica en la forma de obtener dicha función manejando las señales electrónicas de manera especial.

El circuito básico de cada tecnología es una compuerta NAND o una compuerta NOR y de estas se parte a la construcción de funciones lógicas más complejas.

Las tecnologías utilizadas en la fabricación de PLD's son básicamente tres:

a). Tecnología Bipolar.

La tecnología bipolar utiliza como elemento principal al transistor de juntura bipolar de tipo pnp ó npn. El funcionamiento en el que trabajan los transistores para formar circuitos lógicos es, básicamente, en su estado de corte y saturación. Las dos variantes de la Tecnología Bipolar son la Lógica de Transistor-Transistor (TTL) y la Lógica de Emisor Acoplado (ELC). Dentro de los dispositivos lógicos en general, la Tecnología Bipolar TTL es la más usada para el diseño de sistemas digitales. Los circuitos básicos de ésta tecnología son tres:

- Compuerta NAND-TTL con salida TOTEM-POLE (Fig. 2.11).
- Compuerta NAND-TTL con salida TOTEM-POLE de Tres Estados (Fig. 2.12).
- Compuerta OR/NOR-ECL (Fig. 2.13).

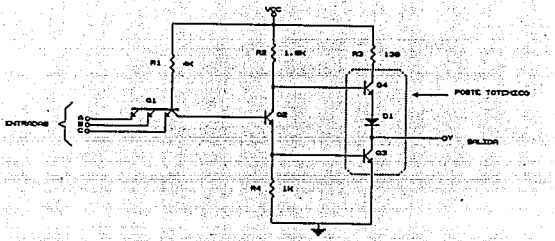
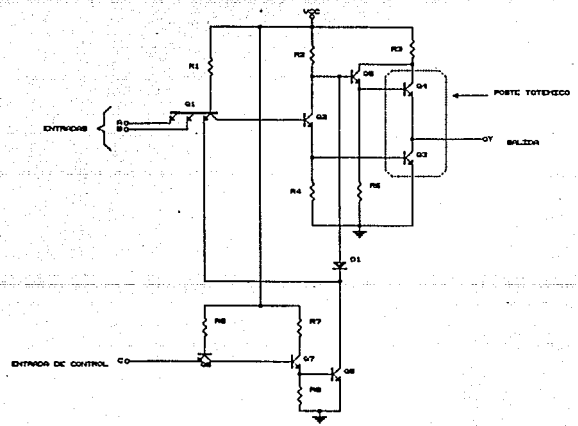


FIG. 2.11



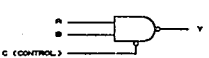


FIG. 2.12

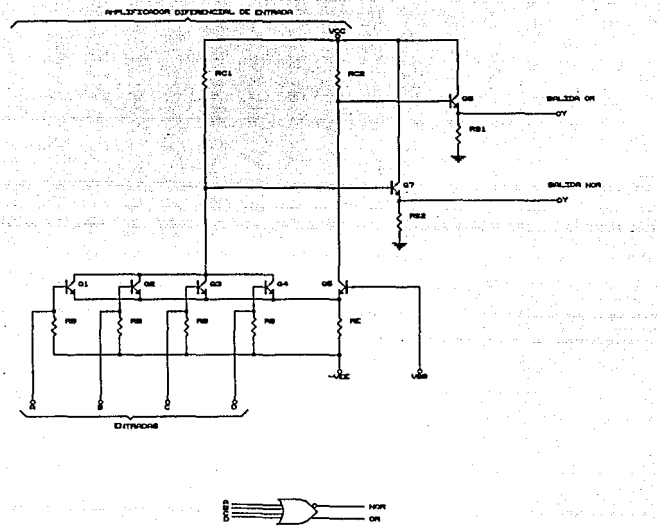


FIG. 2.13

b). Tecnología de Semiconductor de Oxido de Metal Complementado (CMOS).

La tecnología de Semiconductor de Oxido de Metal Complementado (CMOS) hace uso de transistores unipolares MOS tanto

del tipo de canal p como del tipo de canal n. Ambos forman parte del circuito lógico, siendo está característica el motivo de que se le llame "complementado". La ventaja que ofrece la tecnología CMOS es que ambos transistores pueden ser fabricados en un mismo substrato, reduciendo espacio, disipación de potencia, tiempo de retardo de propagación y margen de ruido. Los circuitos básicos de ésta tecnología son dos:

- Compuerta NOR-CMOS (Fig. 2.14).
- Compuerta NAND-CMOS (Fig. 2.15).

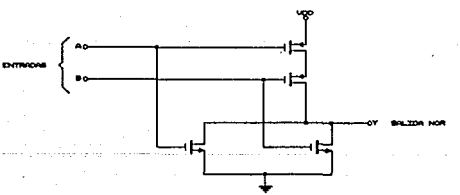


FIG. 2.14

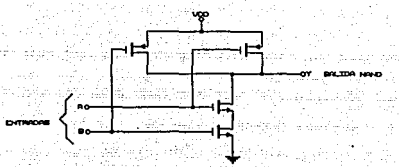


FIG. 2.15

c). Tecnología Arseniuro de Galio (GaAs).

Esta tecnología es la más reciente de todas; en los últimos años, las investigaciones acerca del Arseniuro de Galio (GaAs) a dado exelentes resultados, mejorando algunas características del Silicio y Germanio, principalmente la velocidad de movilidad de los electrónes. El material básico es un monocristal de Arseniuro de Galio semiaislante, dopado con cromo y pulido por vía mecanoquímica. Su arquitectura sigue la teoría clásica de Schockley, utilizando transistores FET-GaAs. Los circuitos básicos de ésta tecnología son dos:

- MESFET - GaAs.
- Compuertas GaAs.

CAPITULO III

SISTEMAS DE SOPORTE PARA DISEÑO CON PLD's

III.1. INTODUCCION.

Este capítulo abarca el desarrollo de software, como soporte para el diseño con PLD's, desde los primeros programas hasta los más recientes.

La creación de paquetería relacionada con el diseño con PLD's ha sido fundamental para el desarrollo de los dispositivos de lógica programable.

El nacimiento de los primeros compiladores lógicos fue a partir de 1978 y desde ese momento numerosas compañías empezaron a desarrollar software y hardware de soporte para diseños con PLD's, dentro del campo de los sistemas digitales.

III.2 INGENIERIA ASISTIDA POR COMPUTADORA (CAE).

En el proceso de diseño, una manera fácil de llevar a cabo éste, es separar el trabajo en partes y unir éstas al final, sin embargo, cuando se unen las partes regularmente se tienen muchos problemas.

problemas.

Muchos procesos de diseño, son ahora automáticos y utilizan la computación como herramienta, así la unión de las partes resulta más fácil de realizar.

Esta manera de diseño por medio de computadora, se conoce como Ingeniería Asistida por Computadora (CAE), la cual ofrece diversas posibilidades para el desarrollo de un diseño.

Las ventajas de emplear herramientas CAE, es que permiten realizar gran número de procesos en tiempos relativamente cortos, con la misma o mejor funcionalidad.

Los sistemas CAE para diseño con PLD's pueden comprender una computadora personal o una red. En general, la diferencia entre estos sistemas radica en la utilización tanto del equipo como del soporte de software que será utilizado; las diferencias en cuanto equipo se basan en la calidad de gráficos, velocidad de procesamiento y capacidad de almacenamiento. En lo referente al soporte de software las principales diferencias estan referidas al rango de funciones que ofrecen, tales como: Entrada del diseño, Simulación Lógica, Programación del dispositivo, etc.

La figura 3.1 muestra la secuencia básica para el diseño con PLD's, algunas de estas partes no son soportadas por los diferentes paquetes para desarrollo de PLD's.

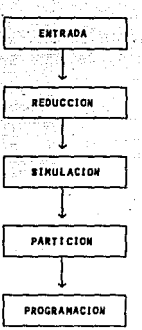


FIG. 3.1.

La parte de entrada abarca la forma de introducir un diseño, esto es, algunos paquetes pueden soportar: ecuaciones booleanas, tablas de verdad, formas de onda, máquina de estados, etc.; otros sólo aceptan algunos de los tipos antes referidos. Usualmente la entrada es por captura esquemática, así, el sistema CAE tendrá una librería de funciones lógicas estandar, estas son llamadas a la pantalla por medio de comandos del teclado o mouse. Despues de la entrada es importante la compilación del diseño, para poder llegar a su procesamiento.

En la parte de reducción se hace la minimización del diseño, es decir, aquí se eliminan las redundancias que pudieran existir y se optimizan las ecuaciones que se introdujeron o generaron. Los diferentes paquetes de software ponen gran interes en esta etapa.

La parte de simulación contiene la etapa de verificación del diseño, aquí se determina si el trabajo fue bien elaborado o si existen errores que se tengan que corregir. Esta etapa es de suma importancia, debido a que es la base para implementar correctamente las funciones lógicas en los dispositivos. Antes de simular por medio del sistema CAE se tienen que conocer las señales de entrada y salida, para verificar si el comportamiento del circuito es el esperado. Un error puede ser corregido regresando a la rutina de datos de entrada.

Despues de la simulación, la partición del diseño cobra una gran importancia; aquí, se establece qué parte de la lógica quedará programada en cada dispositivo. En el proceso de partición se debe conocer el dispositivo, para saber que cantidad de lógica será soportada por éste. Una vez que se ha hecho la partición el paso siguiente es la generación del archivo JEDEC para programación.

Finalmente, la parte de programación, que se compone del archivo JEDEC y la etapa de traslado de éste hacia el programador, es la etapa en la cual se establecen las características finales del dispositivo a programar, así como también se hace el enlace hacia el programador.

El proceso de diseño por medio de computadora termina cuando se programa el dispositivo; para esto, se disponen diferentes programadores con diferentes funciones. La programación comúnmente se lleva a cabo por medio de la comunicación de la PC con el programador.

Las tres posibles configuraciones de sistemas CAE para diseño con PLD's, evaluadas en función al soporte de posibilidades que poseen, son: Sistema CAE completo más programador, PC dedicada a software más programador, Programador Universal.

El sistema CAE completo más programador es capaz de soportar captura esquemática, ecuaciones lógicas, expansión booleana, simulación, síntesis de formas de onda y generación de autoprueba. En lo referente a la segunda configuración, esta soporta ecuaciones lógicas, expansión booleana, simulación limitada y poco grado de prueba. Por último, el programador universal soporta tabla de verdad y vectores de entrada, por medio de su propio software y utilizando la PC como interfaz.

III.3 SISTEMAS DE SOPORTE

El desarrollo de software, es la clave para diseñar con dispositivos de lógica programable. Ahora, en lugar de definir la lógica de PLD's como un simple mapa de fusibles, es posible especificar estos usando ecuaciones booleanas, tablas de verdad, diagramas de estados y formas de onda. Esto aumenta la eficiencia del diseñador.

El software para diseño con PLD's puede ser clasificado en dos partes: los lenguajes dependientes de la manufactura y los lenguajes independientes de la manufactura.

III.3.1 LENGUAJES DEPENDIENTES DE LA MANUFACTURA

El uso de software como soporte para el diseño con PLD's vino, en el primer intento, dependiendo de la manufactura del dispositivo. De éste modo se crearon diferentes tipos de paquetes, cada uno soportando los dispositivos que pertenecian al mismo fabricante.

Los principales programas dependientes de la manufactura fueron los siguientes:

a). PALASAM 2
b). AHAZE
c). PLAAN
d). HELP
e). PLPL
f). APEEL
g). A+PLUS
h). iPLDS II
i). SISTEMA DE DESARROLLO ERASIC
j). XACT

a). ENSAMBLADOR PAL 2 - PALASAM 2

El primer lenguaje fue una especie de Fortran basado en un Ensamblador-PAL (PALASM) de la compañía MMI, cuya implementación de nuevos dispositivos, dejó fuera el software existente; ésto

llevó a que los fabricantes tuvieran que hacer una programación compatible con los nuevos PLD's. Así, el lenguaje dependiente de la manufactura, soportaba sólo los productos ofrecidos por el fabricante que además proveía el lenguaje.

Una vez establecido el PALASAM, se procedió a dar lugar a la estandarización; con esto nace una segunda generación, el compilador PALASAM 2, el cual soporta virtualmente todos los dispositivos fabricados por MMI. Este es capaz de manejar PLD's asíncronos (los que usan un reloj separado para cada salida registrada, como el PAL20RA10). También tiene mejora en la simulación por vectores de prueba.

PALASAM2 es en escencia diseñado para ecuaciones booleanas de entrada y es adecuado para bajas , medias o complejas aplicaciones de PALs.

El archivo fuente creado por el diseñador es llamado Especificación de Diseño PAL (PDS), el cual es procesado para verificar su sintaxis y generar un archivo "PALASAM.TRE", el cual es usado para determinar si la definición lógica es adecuada en el PAL especificado, así como también para su simulación. De este modo se genera un mapa de fusibles y se crean dos archivos, con extensiónes ".XTP" y ".JED".

El archivo XTP contiene el mapa de fusibles generado, y el JED el archivo de salida JEDEC, el cual es utilizado para ser tranferido a un programador de PLD's.

b). ECUACIONES AUTOMATICAS MAP y ZAP - AMAZE

Ecuaciones Automáticas Map y Zap (AMAZE) creado por Signetics Corp. ofrece mejor funcionalidad para diseños más complejos. El programa principal es BLAST, éste es utilizado para simular e identificar el dispositivo.

Este software permite a varias entradas, salidas y registros de estado agruparse, en un mismo conjunto de ecuaciones, para su posterior procesamiento y de este modo quedar juntos en un mismo dispositivo. Elimina redundancias de productos término, soporta simulación y vectores de prueba automáticos, para un diseño específico. Puede trasladar también diseños esquemáticos desde paquetes tales como: DASH[8] de Future Net Corp. u OrCAD[13] de Systems Corp.

c). ANALISIS LOGICO PROGRAMABLE - PLAN

National Semiconductor Corp. ofrece un soporte de software para sus PLD's, éste es llamado Análisis lógico Programable (PLAN), aceptable para algunas aplicaciones. Dispone de minimización lógica, sobre ecuaciones booleanas de entrada, permite simulación por medio de vectores de prueba, y es capaz de generar también su archivo JEDEC.

d). LENGUAJE ENRIQUECIDO DE HARRIS PARA LOGICA PROGRAMABLE - HELP

Este paquete Lenguaje Enriquecido de Harris para Lógica Programable (HELP) de la compañía Harris Co., no selecciona el PLD, pero ofrece chequeo de error del dato de entrada a la vez, únicamente acepta ecuaciones como entrada en forma de suma de productos.

e). LENGUAJE DE PROGRAMACION PARA LOGICA PROGRAMABLE - PLPL

El paquete de lenguaje de Programación para Lógica Programable (PLPL), introducido por AMD, es el más avanzado de los lenguajes básicos dependientes de la manufactura. Tiene una capacidad de soporte más grande, permite a grupos de señales ser definidos y usados en descripciones de pin y ecuaciones booleanas. Soporta ecuaciones complejas, estructura de descripción de máquina de estados, contiene minimización e incluye simulación.

f). ENSAMBLADOR PARA LOGICA PROGRAMABLE BORRABLE ELECTRICAMENTE - APEEL

Este software Ensamblador para Lógica Programable Borrable Eléctricamente (APEEL) de Internacional CMOS Technology, es introducido en 1987 como un compilador de soporte para dispositivos lógicos borrables eléctricamente (EEPLD). Este es simple y adecuado para diseños básicos. Soporta simulación y vectores de prueba, incluye la construcción de una pantalla de editor y salidas en archivo estandar JEDEC, no soporta

minimización.

g). SISTEMA DE USUARIO DE LOGICA PROGRAMABLE ALTERA - A+PLUS

El paquete Sistema de Usuario de Lógica Programable Altera (A+PLUS) del fabricante Altera Co., soporta EPLD's CMOS, permite realizar su diseño por ecuaciones boleanas, máquina de estados y captura esquemática y sus operadores lógicos son similares al paquete anterior.

A+PLUS permite la minimización, por medio de teoremas de Morgan, su software puede seleccionar el mejor dispositivo de Altera y puede automáticamente asignar el número de pin para todas las salidas y entradas de las señales.

Para la captura esquemática, Altera ofrece instrucciones de librería que contienen dispositivos TTL y CMOS de la familia 7400 y 4000 respectivamente, mediante el paquete esquemático Logi Caps. Para trasladar el archivo esquemático a un archivo de diseño, donde posteriormente será procesado, se utiliza un algorítmo de traslación Macro Función, el cual elimina redundancias.

Este paquete contiene minimización y la salida final es dejada en un archivo JEDEC. La simulación puede ser realizada antes de dejar el diseño programado en un dispositivo.

h). SISTEMA DE DESARROLLO PROGRAMABLE INTEL II - iPLDS II

La primera versión del Sistema de Desarrollo Programable

Intel II (iPLDS) fue creada para soportar dispositivos de Altera, pero a medida que Intel fue desarrollando sus propios dispositivos, surgió la necesidad de crear su propio soporte de software, iPLDS II.

El software básico permite diseñar por medio de ecuaciones boleanas, como entradas. Opcionalmente el paquete permite máquina de estados y captura esquemática. Para esto Intel ofrace Schema II PLD, el cual consta de una librería de dispositivos más comunes SSI/MSI TTL y primitivos EPLD's.

i). SISTEMA DE DESARROLLO ERASIC

El Sistema de Desarrollo ERASIC, de la compañía Exel, ofrece un completo soporte de software como los demás paquetes. Desarrolla Mulcimap y Multisim para generación de archivo JEDEC y simulación, respectivamente.

j). TECNOLOGIA DE DISEÑO ASISTIDO POR COMPUTADORA AVANZADO DE XILINX - XACT

Xilinx, ofrece el paquete Tecnología de CAD avanzado de Xilinx (XACT), que no es conformado para arquitecturas de PLD's convencionales. Las entradas del diseño pueden ser de manera gráfica, ecuaciones booleanas o mapas de Karnough. La entrada esquemática es generada usando varios paquetes de software para esquemáticos, utilizando una especial librería de dispositivos Xilinx.

III.3.2 LENGUAJES INDEPENDIENTES DE LA MANUFACTURA

El aumento en la fabricación de PLD's dió como resultado que surgieran soportes de software que no dependieran de la manufactura del dispositivo. Así comenzaron a desarrollarse diferentes paquetes que soportan una gama de dispositivos indiferentemente de su manufactura, éstos son:

a). CULP
b). PLDesigner
c). PLD Design System
d). LOG/IC
e). etc.

a). COMPILADOR UNIVERSAL PARA LOGICA PROGRAMABLE - CULP

El Compilador Universal para Lógica Programable (CULP) de Logical Devices, en su versión 1.01, es creado en 1983, soportando 29 dispositivos. Actualmente contiene muchos dispositivos incluyendo PROM.

Este lenguaje utiliza ecuaciones booleanas, tablas de verdad y diagramas de estado como entradas. También se puede manejar captura esquemática usando PC-Caps y otros. El compilador maneja cuatro niveles de reducción lógica, seleccionable por el diseñador, esto elimina redundancias dentro del diseño. También soporta simulación por medio de vectores de prueba.

b). DISEÑADOR DE DISPOSITIVO LOGICO PROGRAMABLE - PLDesigner

El software llamado Diseñador de Dispositivo Lógico Programable (PLDesigner) del fabricante Minc, implementa nuevas y únicas características para operaciones avanzadas. En adición a los métodos de entrada como: ecuaciones booleanas, tablas de verdad, máquinas de estados y captura esquemática; también soporta entradas por formas de onda. Esto permite al diseñador introducir una señal por medio de forma de onda y obtener una salida en forma de onda de un dispositivo secuencial.

El paquete no requiere que se le indique el tipo de PLD. Al final de la compilación la programación indica en que posibles dispositivos puede fijarse éste.

El software soporta complejos dispositivos y arquitecturas, incluyendo avanzados Intel/Altera. Utiliza un estructura en lenguaje Pascal y avanzadas capacidades de simulación.

c). SISTEMA DE DISEÑO DE PLD

El Sistema de Diseño de PLD de Hewlett-Packard, fue creado en 1988. Este paquete permite introducir la lógica independientemente del dispositivo a usar. El software puede seleccionar el dispositivo o los dispositivos que soportarán la lógica definida, desde la lista de dispositivos disponibles especificados por el diseñador, o puede proveer una lista de dispositivos aceptables donde el usuario puede seleccionar.

Es capáz de particionar un diseño en múltiples PLD's. Permite al diseñador identificar la velocidad crítica de la señal. Las entradas pueden ser por ecuaciones booleanas, tablas de verdad, diagramas de estado, formas de onda y captura esquemática.

d). LOG/IC

El software LOG/IC de Elan, Kontron, Isdata, permite desarrollar la lógica independientemente del dispositivo a utilizar. Soporta partición del diseño automática. Las entradas pueden ser por medio de ecuaciones lógicas, tablas de verdad, diagramas de estado y opcionalmente captura esquemática. Incluye simulación por medio de vectores de prueba.

e). Existen otros soportes de software de menor capacidad como son: Ambiente de Diseño Lógico Programado (PLDE) de Aldec, Sistema de Desarrollo de Lógica Programable (ELDS) de Pistohl Electronic Tool Co., PLD Master de Dasy Systems, etc.

III.4 SOPORTE DE SOFTWARE DISEÑADOR DEL FUTURO (FUTURE DESIGNER) PARA DISEÑO CON PLD's.

Existe otro soporte de software para diseño con PLD's, FUTURE DESIGNER, este merece especial atención por ser el utilizado para el rediseño en la presente tesis. Este paquete presenta algunas características que otros paquetes no contienen.

Una diferencia importante entre FUTURE DESIGNER y los demás paquetes, es la capacidad de dispositivos que soporta en su librería, así como también la capacidad de poder accesar a ella y almacenar más dispositivos.

Otra diferencia radica en las capacidades de simulación de diseños. Si bien el paquete tiene limitaciones, tambien sus características son adecuadas y poderosas.

Future Designer es creado por DATA I/O y FUTURE NET Co., con el fin de soeportar diferentes manufacturas, arquitecturas y tecnologías.

El paqute consta de cinco partes: ABEL, GATES, DASH, PLDlinx y UNISITE. La interrelación entre las partes se presenta en la figura 3.2.

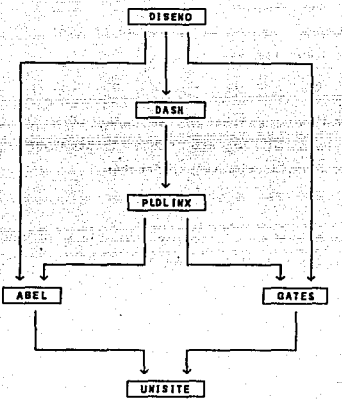


FIG: 3.2.

DISEÑO

En la parte de diseño es donde se lleva a cabo la creación del diseñador, esta puede ser expresada como:

- Diagrama esquemático
- Ecuaciones booleanas
- Máquinas de estados
- Tablas de verdad
- Combinación de las anteriores

Dependiendo el tipo de entrada, el flujo sigue una ruta determinada, es decir, para ecuaciones booleanas, máquinas de estados y tablas de verdad, la entrada es directamente por la parte de Gates o Abel. Cuando la entrada es un diagrama esquemático, este es creado en Dash y enlazado por medio de PLDlinx hacia Gates o Abel, donde posteriormente será procesado.

Esta etapa es la parte fundamental en cualquier diseño de PLD's, debido a que si algún error es cometido en el planteamiento del diseño, el resultado no será el esperado, independientemente de los pasos y el paquete utilizado.

DASH

Esta parte es un interactivo editor gráfico. Es diseñado para ayudar a la creación de un dibujo de esquema electrónico funcional. Se pueden cargar, crear, conectar, y anotar componentes, para elaborar un circuito electrónico.

El dibujo es hecho en una pantalla de gráficos, que se denomina área de despliegue gráfico. Contiene una zona de comandos de línea, que sirven para conducirse dentro del dibujo, y una línea de mensajes, donde DASH se comunica con el diseñador, a través de información del estado del área gráfica y mensajes de error.

DASH contiene varias librerías que pueden ser utilizadas para construir los esquemas. Estas poseen diferentes dispositivos de varias tecnologías, como: TTL y CMOS, ademas de varios fabricantes

varias tecnologías, como: TTL y CMOS, ademas de varios fabricantes como: Motorola e Intel.

PLDLINX

Este software, tiene la función de trasladar automáticamente, los diseños realizados con DASH a ABEL o GATES. Traslada la conectividad de datos producidos en el esquemático a ecuaciones booleanas, por medio de un archivo .abl o .sds para el software de ABEL o GATES, respectivamente.

Durante el proceso de traslado, PLDlinx asigna la declaración de las variables a los puntos localizados dentro del esquemático. El programa lee el archivo de los dispositivos especificados e identifica cuales serán salidas, así genera ecuaciones booleanas para cada salida declarada.

PLDlinx se compone de diferentes archivos que sirven de auxilio para el desarrollo, estos son: pldlinx.sym, pldlinx.flb y abel3lib.dev. El primero contiene una librería propia con dispositivos utilizables para captura esquemática, la cual contiene dispositivos que algunas veces no se encuentran en otras librerías.

El archivo .flb contiene funciones lógicas que se utilizan cuando se hace la transferencia del esquema a ecuaciones. Estas funciones se pueden variar y aumentar entrando al archivo.

El tercer archivo contiene la descripción de los dispositivos soportados por PLDlinx, éste se utiliza para verificar el asignamiento de las entradas y salidas del PLD seleccionado.

ABEL

Esta parte es básica para realizar el diseño en un PLD, permite utilizar una combinación de esquemáticos (con PLDlinx), ecuaciones, tablas de verdad o diagramas de estado para describir un diseño. Automáticamente convierte la descripción del comportamiento del diseño en una representación estructural para implementación en un PLD. La síntesis lógica, incluyendo reducción, puede ser desarrollada. De este modo se genera el archivo JEDEC para programar el PLD.

GATES

Este sintetizador lógico es una herramienta de software para diseño múltiple. Permite usar una combinación de esquemáticos, ecuaciones booleanas tablas de verdad y diagramas de estados, para definir un diseño. Incluye un interactiva interface de usuario y alto nivel funcional para verificar los errores en el diseño.

Puede desarrollar reducción, factorización, partición y selección del dispositivo. Permite describir elementos de circuito como: contadores, multiplexores, decodificadores y máquinas de estado en términos de ecuaciones, tablas de verdad o descripción de estados. El archivo de salida es el JEDEC.

UNISITE

El UNISITE es un programador universal que contiene un software propio, el cual proporciona un menú de opciones para selección de la función a realizar. Se puede enlazar con una PC para intercomunicarse, así como también puede utilizar la pantalla como monitor para desplegar su información.

Soporta todos los dispositivos de tecnologías programables como son: EPROM, bipolar, NOS, CMOS y ECL; así como tipos de encapsulado: DIP, PLCC, SOIC. Unisite provee un soporte de programación, verificación, y prueba de todas las marcas programables de memorias, lógica, y dispositivos microcontroladores.

CAPITULO IV

TARJETA DE CONTROL DEL SISTEMA DE LA UNIDAD TERMINAL PROGRAMABLE (PTU)

IV.1. INTRODUCCION.

Debido a que el objetivo del presente trabajo, es el rediseño de la tarjeta llamada Control del Sistema (SYSTEM CONTROL), es importante conocerla en su estado original, es decir, antes del rediseño. Por esta razón, este capítulo describe en primer lugar, a la Unidad Terminal Programable (PTU), considerando sus características principales y las partes que la componen, con el fin de ubicar a la tarjeta dentro del equipo. En segundo lugar, se hace referencia a la tarjeta, describiendo su funcionamiento dentro de la unidad y sus diagramas eléctricos. Además se hace mención de los bloques funcionales dentro de la tarjeta.

IV.2. UNIDAD TERMINAL PROGRAMABLE (PTU).

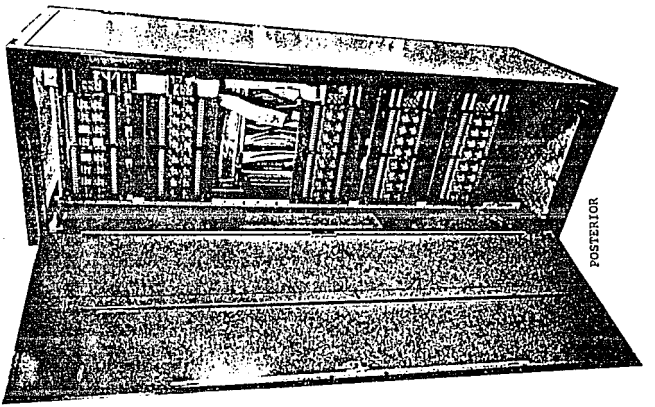
Siendo una terminal remota, la Unidad Terminal Programable (PTU) se comunica con una estación o estaciones maestras vía enlace de comunicación y provee información o ejecuta un control funcional como sea requerido por el sistema. El equipo cuenta con software almacenado permanentemente en memoria (Firmware) que extiende las capacidades funcionales de un equipo convencional de

Control Supervisor y Adquisición de Datos (SCADA), el cual lleva a cabo la recolección de datos de estado, alarmas o señales de entrada analógica y control de contactos de salida o puntos de prueba para señales analógicas de dispositivos terminales, adicionando complejas secuencias de control, múltiples interfaces de comunicación, procesamiento de datos locales para diferentes requerimientos industriales y otras funciones similares.

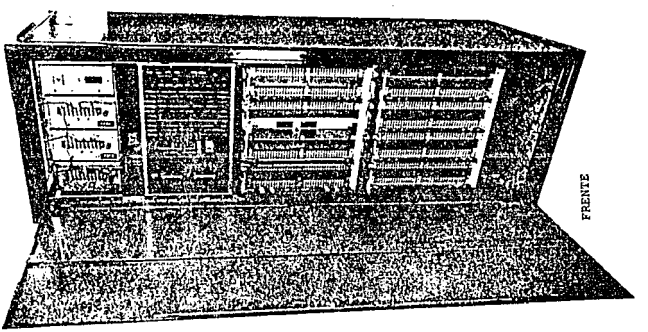
La PTU es mostrada en la figura 4.1. Las características del gabinete son: construcción de uso rudo, paneles laterales removibles y puertas frontal y trasera, que facilitan la instalación, mantenimiento y expansión.

Una PTU básica, está compuesta de los siguientes elementos, tal como se muestra en la figura 4.2:

- Compartimiento para tarjetas de suministro de energía con módulos de suministro de energía y un panel de control de energía.
- Compartimiento para ensamble de tarjetas lógicas que contiene procesador, memoria y tarjetas lógicas de E/S.
- Panel de E/S con tarjetas terminales de E/S que proporcionan conectores para cables de E/S.
- Ensamble de cables de Interconexión.
- Gabinete con compartimientos de montaje, postes de tierra y conectores de suministro de energía.



POSTERIOR



FRENTE

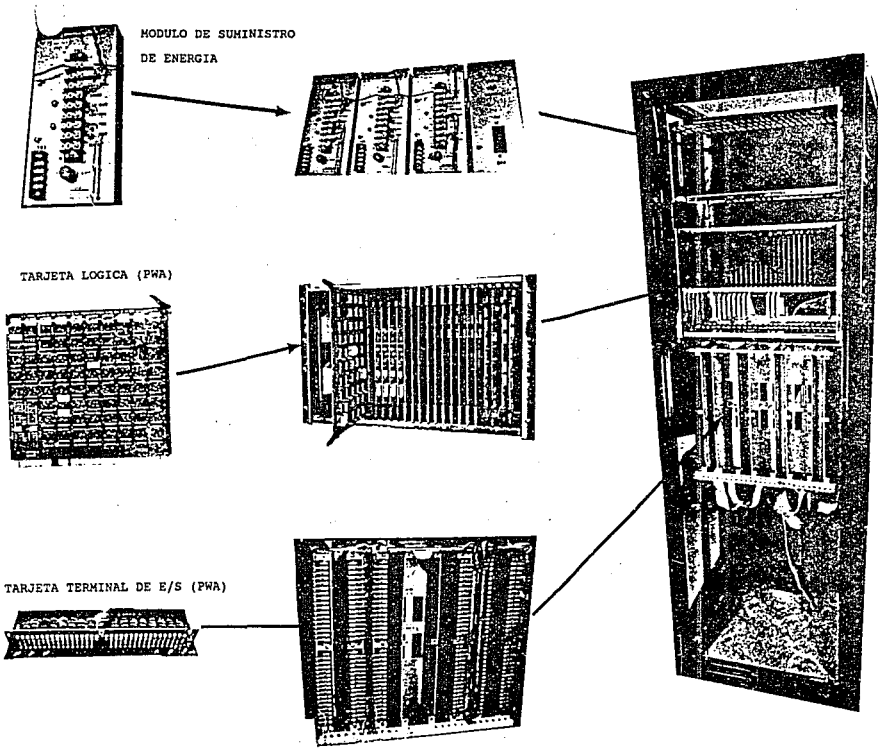

MODULO DE SUMINISTRO
DE ENERGIA
TARJETA LOGICA (PWA)
TARJETA TERMINAL DE E/S (PWA)

FIG. 4.2

Otros elementos que pueden ser encontrados en una PTU básica son: el modem y panel de relés con conectores para cables de campo. Adicionalmente, la capacidad de expansión de una PTU permite diferentes configuraciones de número de tarjetas de E/S, paneles de E/S, paneles de relés y número de gabinetes.

Las dimensiones físicas de la PTU son aproximadamente 209.2 cm. de alto, 73.8 cm. de ancho y 83.8 cm. de fondo.

Los requerimientos de energía más comunes para la PTU son:

- 125 Vcd.
- 48 Vcd.
- 28 Vcd.

El funcionamiento de la PTU se basa en los siguientes bloques, su interacción se muestra en el diagrama de la figura 4.3:

PROCESADOR TRW 2000.

El procesador de la PTU es llamado TRW 2000 por sus fabricantes (TRW Controls), este es el controlador básico del sistema y un elemento del proceso de datos. El multibus de arquitectura paralela de 16 bits le da un rápido y flexible tiempo real de procesamiento de datos. Las características del procesador son, principalmente: (1) separación del Programa en Memoria y Datos de Memoria, (2) habilidad para direccionar Datos de Memoria,

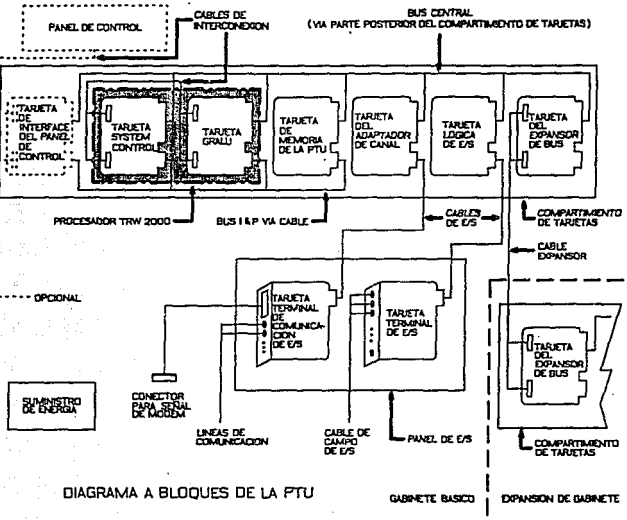


DIAGRAMA A BLOQUES DE LA PTU

FIG. 4.3

registros de dispositivos de E/S y Programa en Memoria de una misma manera en el Bus Central (el cual es distinto del Bus de Instrucciones dedicado al programa en memoria), y (3) habilidad para direccionar 64 kilopalabras.

La estructura de direccionamiento de 64 kilopalabras de la arquitectura TRW 2000 es particionada de manera que los 4k más altos de direcciones son dedicados a direccionar registros de dispositivos E/S. Los siguientes 28k de direcciones son reservados para el programa en memoria y los 32k más bajos son dedicados a datos de memoria.

El procesador TRW 2000 está compuesto de dos tarjetas de circuito impreso, la tarjeta de Control del Sistema y la tarjeta GRALU. Estas tarjetas deben ser instaladas en las ranuras 2 y 3, respectivamente, del compartimiento de tarjetas lógicas.

MEMORIA DE LA PTU.

Este bloque contiene el software almacenado permanentemente en la memoria de la PTU. Proporciona un programa (sólo lectura) y datos (lectura/escritura) de memoria para el procesador. El almacenamiento del programa es implementado con PROM's.

Una o más tarjetas de memoria pueden ser utilizadas en una PTU, tal como sea requerido. Cada tarjeta proporciona una capacidad para programa en memoria de hasta 4k x 16 bits y una capacidad para datos de memoria de 1k x 16 bits.

Las tarjetas de memoria son instaladas adyacentes a las tarjetas del procesador TRW 2000 iniciando en la ranura 4.

ADAPTADOR DE CANAL Y TERMINAL DE COMUNICACION E/S.

El adaptador de canal es un dispositivo de E/S que es operado bajo un programa de control del procesador para transferir datos en serie (código NRZ). La tarjeta del adaptador de canal proporciona la interface de comunicación entre la PTU y la estación maestra o dispositivos de control. Algunas de sus caracteristicas son: velocidad de bits seleccionable de 75 a 9600 BPS, interface de modem, temporización extendida para "petición de envío/listo para envío" (request-to-send/clear-to-send) e inserción/borrado de cero (después de seis bits continuos en "1"). La tarjeta del adaptador de canal puede ser instalada en cualquier ranura para tarjetas E/S pero es generalmente localizada enseguida de la tarjeta(s) de memoria de la PTU.

Asociada con la tarjeta Adaptador de Canal está la tarjeta Terminal de Comunicación E/S y el ensamble de cable de interconexión. La tarjeta terminal proporciona el conector del modem y las conexiones para la línea(s) de comunicación.

TARJETAS LOGICAS DE E/S y TARJETAS TERMINALES DE E/S.

Las tarjetas lógicas de E/S proporcionan las funciones básicas de E/S, tales como Estado de entrada, entrada A/D y control de salida. Realizan el acondicionamiento de las señales

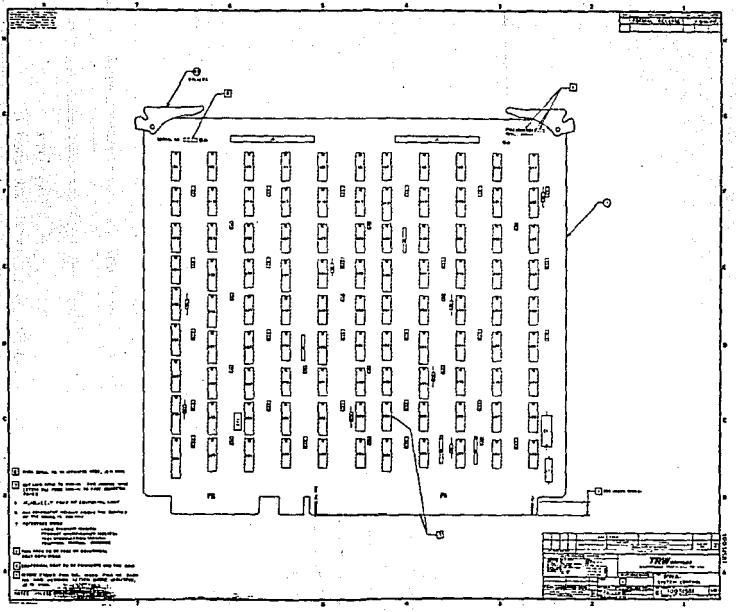


FIG. 4.4

externas, esto es, aisla y filtra los contactos de entradas, conversión analógica a digital y aislamiento de contactos de salidas. Los aspectos más importantes al respecto de este bloque de tarjetas son:

Entrada Discreta. La tarjeta de entrada discreta opera como un dispositivo estático que suministra 16 bits de datos de entrada acondicionados al bus central, cuando son apropiadamente direccionados por el procesador. Elimina, para cada entrada, picos y descargas de alto voltaje. Cuenta con voltajes opcionales de +5, +24 y +48 volts de c.d.

La tarjeta de entradas discretas puede ser instalada en cualquier ranura de E/S y requiere de un cable que lo conecte a su terminal asociada.

Entrada Analógica. El sistema de entrada analógica consiste de una tarjeta convertidora A/D y una o más tarjetas multiplexoras A/D. Cada una cableada a su propia terminal Analógica de E/S mediante un cable plano. Las tarjetas A/D pueden ser instaladas en cualquier ranura de E/S, pero generalmente son localizadas en el compartimiento de tarjetas básicas, en el cual, la tarjeta Convertidora A/D se encuentra primero, seguida consecutivamente por la tarjeta Multiplexora.

Salida de Control. Consiste de una tarjeta de salida de control, un montaje de Panel de Relés y cables de interconexión. Este dispositivo proporciona una manera segura de manejar 16 controles de relés mediante manejadores de alta y baja corriente.

Los 16 relés de control son montados en el panel de relés. La tarjeta de salidas de control puede ser instalada en cualquier ranura de E/S y requiere de un cable que la interconecte al montaje de Panel de Relés asociado.

EXPANSOR DE BUS.

El Expansor de Bus es bidireccional (receptor/conductor), el cual provee una extensión del bus principal de la TRW 2000, fuera del primer compartimiento de tarjetas básicas, en un compartimiento de tarjetas adicionales. Cada expansión del bus requiere del uso del Expansor de Bus, que incluye dos tarjetas Expansoras de Bus, dos cables de interconexión y un compartimiento de expansión de tarjetas. La tarjeta de Expansión de Bus en el compartimiento de expansión de tarjetas es instalada en la ranura 1. La tarjeta de Expansión de Bus en el compartimiento de tarjetas básicas puede ser instalada en cualquier ranura de tarjetas de E/S.

IV.3. TARJETA DE CONTROL DEL SISTEMA.

Como se mencionó anteriormente, la tarjeta SYSTEM CONTROL es una de dos tarjetas que componen al procesador TRW 2000. Esta se encarga del acondicionamiento de señales para realizar un contol funcional en el sistema. Esta tarjeta contiene la lógica requerida para:

1) Controlar la ejecución de instrucciones del programa en memoria.
2) Controlar la actividad del Bus Central de la TRW 2000.
3) Controlar las interrupciones (prioritarias) del sistema.

La tarjeta se divide en varios bloques funcionales distribuidos en un circuito impreso que contiene 99 CI's de la serie TTL, con los que se realizan todas las funciones lógicas; también contiene 6 conectores (PA, PB, JC, JD, JE, JF). El conector PA de 110 pines y el PB de 50 pines están conectados a la ranura 2 del compartimiento de tarjetas lógicas, mediante estos conectores la tarjeta se comunica con el Bus Central, los conectores JC y JD de 50 pines conectan a la tarjeta de Control del Sistema con la tarjeta GRALU mediante cable plano, los conectores JE y JF de 16 y 8 pines respectivamente son para pruebas internas de la tarjeta. La distribución de componenetes y los conectores se muestran en la figura 4.4.

Las señales que manejan los conectores, así como su identificación en el diagrama correspondiente se muestra en la figura 4.5 y 4.6.

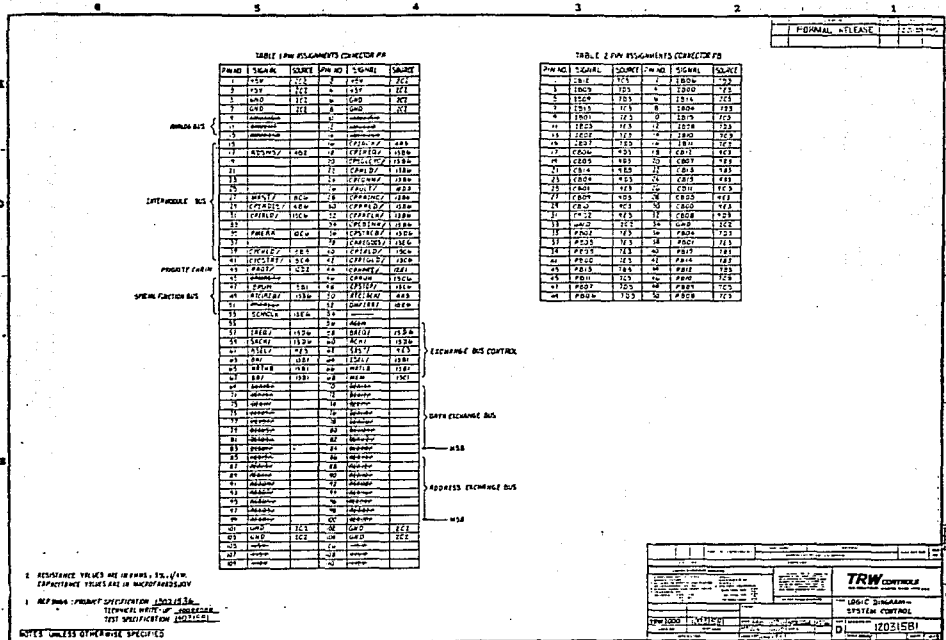

FORMAL RELEASE
TABLE 1 PIN ASSIGNMENTS CONNECTOR P8
TABLE 2 PIN ASSIGNMENTS CONNECTOR P8
ANALOG BUS
INTERMODULE BUS
PRIORITY CHAIN
SPECIAL FUNCTION BUS
EXCHANGE BUS CONTROL
DATA EXCHANGE BUS
ADDRESS EXCHANGE BUS
NOTES UNLESS OTHERWISE SPECIFIED
TRW CONTROLS
LOGIC DIAGRAM
SYSTEM CONTROL
12031581

FIG. 4.5

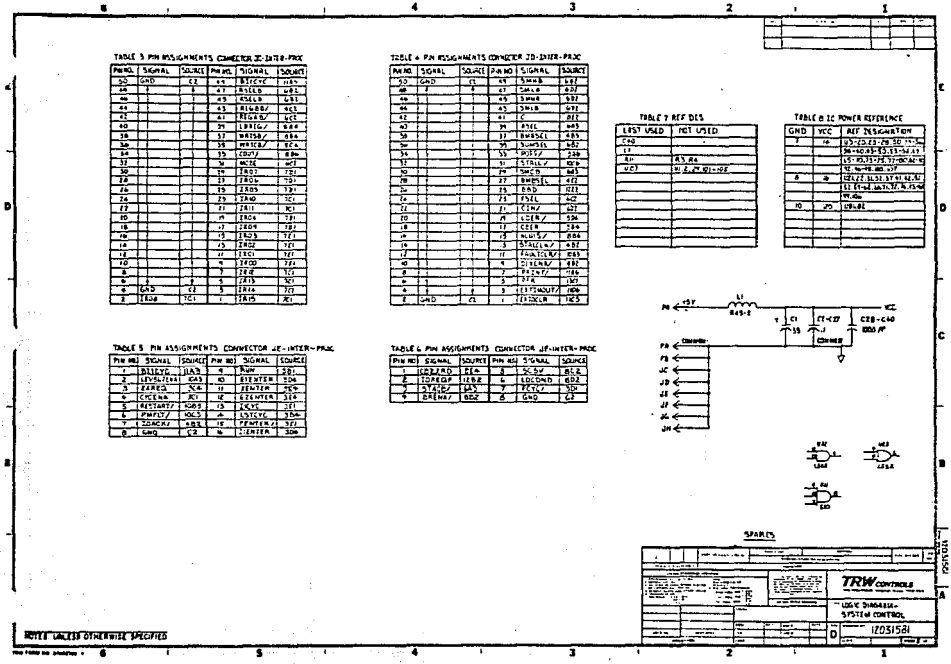


TABLE 3 PIN ASSIGNMENTS CONNECTOR JE-INTER-PROC
TABLE 4 PIN ASSIGNMENTS CONNECTOR JD-INTER-PROC
TABLE 7 REF DES
TABLE 8 IC POWER REFERENCE
TABLE 5 PIN ASSIGNMENTS CONNECTOR JE-INTER-PROC
TABLE 6 PIN ASSIGNMENTS CONNECTOR JF-INTER-PROC
SPARES
TRW CONTROLS
LOGIC DIAGRAM
SYSTEM CONTROL
1705158
NOTES UNLESS OTHERWISE SPECIFIED

FIG. 4.6

Los bloques funcionales con que cuenta la tarjeta son los siguientes:

REGISTRO P.

El Registro de Programas contiene las direcciones de la siguiente localidad del programa en memoria a ser accesada por el procesador. Esta es incrementado simultaneamente con la transferencia de información en el Bus I al Registro I o al Registro E. Para llevar a cabo el direccionamiento, el Registro P hace uso del Bus P.

BUS I.

El Bus I es usado para transferir el contenido de la localida direccionada por el Bus P a el procesador.

MUX P.

El multiplexor Mux P, habilita la rama direccionada del Registro P desde el Bus C o desde el Codificador de Interrupciones Prioritarias (durante un ciclo de interrupción).

REGISTRO I.

El Registro de Instrucciones retiene las instrucciones que están siendo ejecutadas por el procesador.

GENERACION DE MICROCONTROL.

La lógica del Generador de Micro Control decodifica las instrucciones en el Registro I y genera las señales básicas de control requeridas para ejecutar las instrucciones.

CONTROL DE ESTADO.

Esta parte controla la secuencia de estado del procesador a través de sus seis ciclos fundamentales basados en entradas provenientes de la lógica del Micro-Control, la lógica del Control de Interrupciones, la lógica del Bus Central de Control y el Panel de Control.

CONTROL DE INTERRUPCIONES.

La lógica del Control de Interrupciones acepta interrupciones solicitadas en cada uno de los niveles prioritarios de la circuitería del monitor de energía, la lógica del Bus Central de Control, el Reloj de Tiempo Real, Temporizadores del Sistema de Alarmas (Watchdog), Dispositivos de E/S y el Panel de Control.

CONDICIONES Y ESTADOS.

Las condiciones de Acarreo (Carry), Señal y Cero generadas durante la ejecución de algunos tipos de instrucciones, tales como proceso y estado del sistema son almacenadas en la lógica de Condiciones y Estados. Estas condiciones son almacenadas en un Registro de Salvamento de Condiciones cuando una interrupción es procesada durante un ciclo de interrupción.

BUS CENTRAL DE CONTROL.

La lógica del Bus Central de Control monitorea toda la selección de bus y los ciclos del bus de servicio, tales como, el temporizador de alarmas, y genera la interrupción apropiada solicitada por el Control de Interrupciones si un error es detectado (tal como una falla en el bus o una localidad de memoria o dispositivo de E/S no existente).

Toda la circuitería encargada de realizar las funciones anteriormente descritas es mostrada en los diagramas eléctricos de las figuras 4.7 a 4.17, estos corresponden a los diagramas originales de la tarjeta de Control del Sistema.

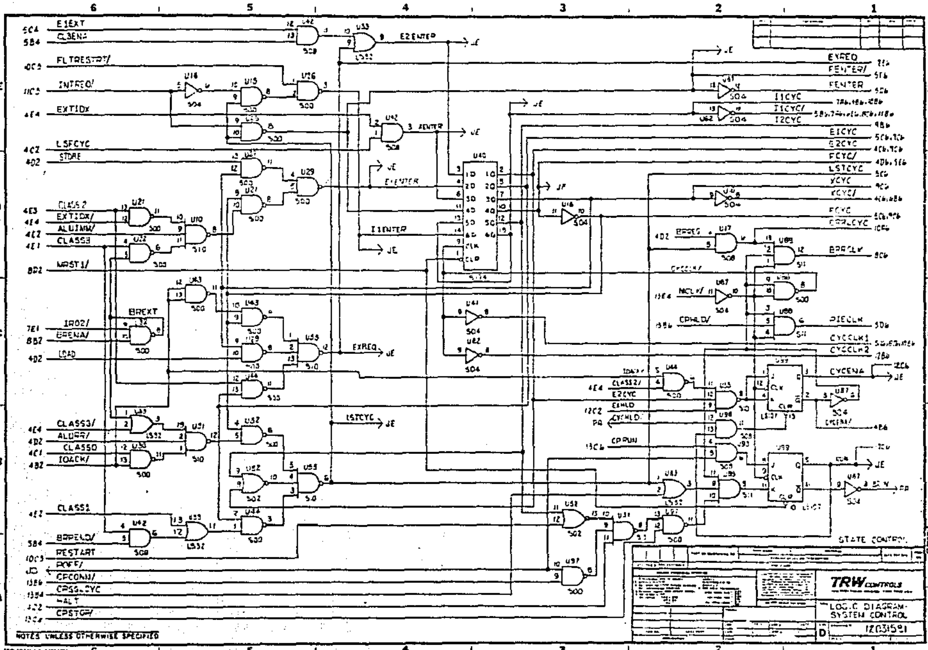

STATE CONTROL
TRW CONTROLS
LOGIC DIAGRAM-SYSTEM CONTROL
NOTES: UNLESS OTHERWISE SPECIFIED

FIG. 4.7

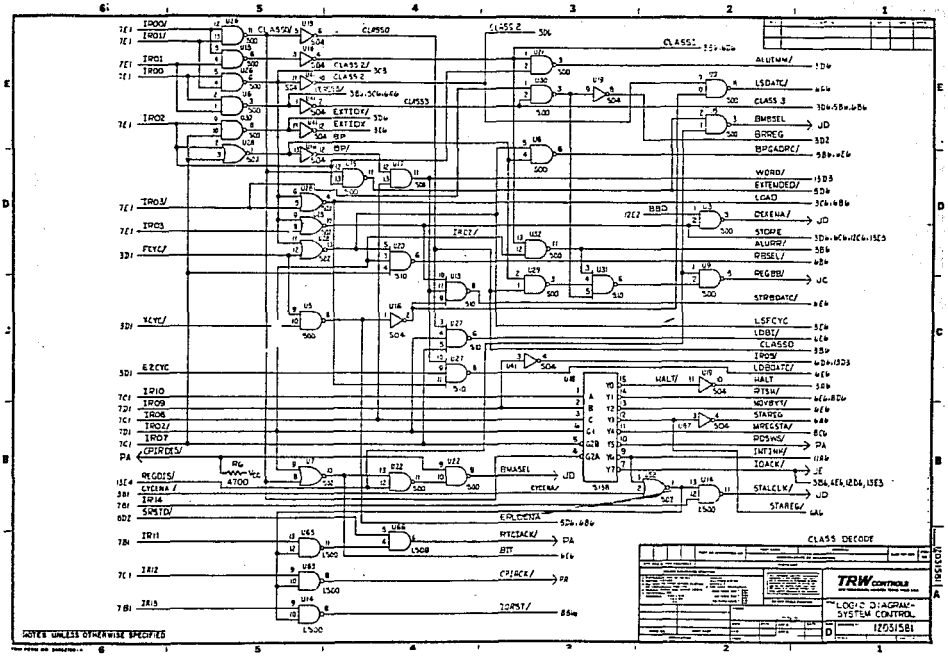

IR00/
IR01/
IR01
IR00
IR02
IR03/
IR03
FEYC/
XCYC/
EZCYC
IR10
IR09
IR08
IR07
IR02/
IR07
CPIRDES/
REGDES/
CYCENA /
IR14
SMSTD/
IR11
IR12
IR13
CLASS2/
CLASS3
EXTEDX/
EXTEDX/
BP/
BP/
CLASS2
CLASS3
CLASS2
CLASS3
IR02/
BMUSEL
JD
EYCENA/
EPLCENA
RTCIACK/
BIT
CPIACK/
IORST/
ALUTEMM/
LSDATE/
CLASS 3
BMBSEL
BRREG
BPCADPC/
WORD/
EXTENDED/
LOAD
DEXENA/
STOPE
ALURR/
RBSEL/
REGBB/
STRBDATE/
LSFCYC
LDBT/
CLASSD
IR09
LDBDATE/
HALT/
HALT
RTSH/
MOVBYT/
STAREG
MREGSTA/
PDSWS/
INTINH/
IOACK/
STALCLK/
STAREG/
CLASS DECODE
TRW CONTROLS
LOGIC DIAGRAM
SYSTEM CONTROL
12705181
NOTES UNLESS OTHERWISE SPECIFIED

FIG. 4.8

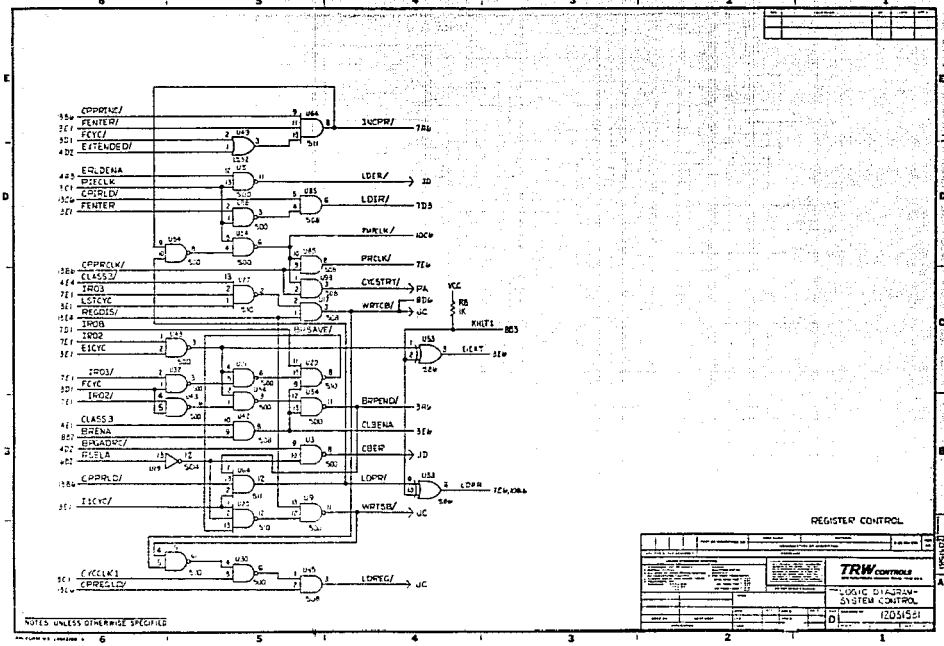

CPPRIME/
FENTER/
FCYC/
EXTENDED/
INCPR/
ERLDENA
PIECLK
CPIRLEV
FENTER
LDER/
LDIR/
PWRLK/
CPPRCLK/
CLASS3/
IRQ3
LSTCYC
REGDIS/
IRQB
IRQ2
EICYC
PRCLK/
CKSTRT/
WRTCB/
RHIS
HISSAVE/
EICNT
IRQ3/
FCYC
IRQ2/
BRPEND/
CLASS3
BRENA
BRADDR/
PIELA
CLBENA
CDER
CPPRLD/
LDPD/
LDIR
IICYC/
WRTSB/
CPCCLK1
CPPRCLD/
LDREG/
VCC
RB
1K
REGISTER CONTROL
TRW CONTROLS
LOGIC DIAGRAM-
SYSTEM CONTROL
1205581
NOTES UNLESS OTHERWISE SPECIFIED

FIG. 4.9

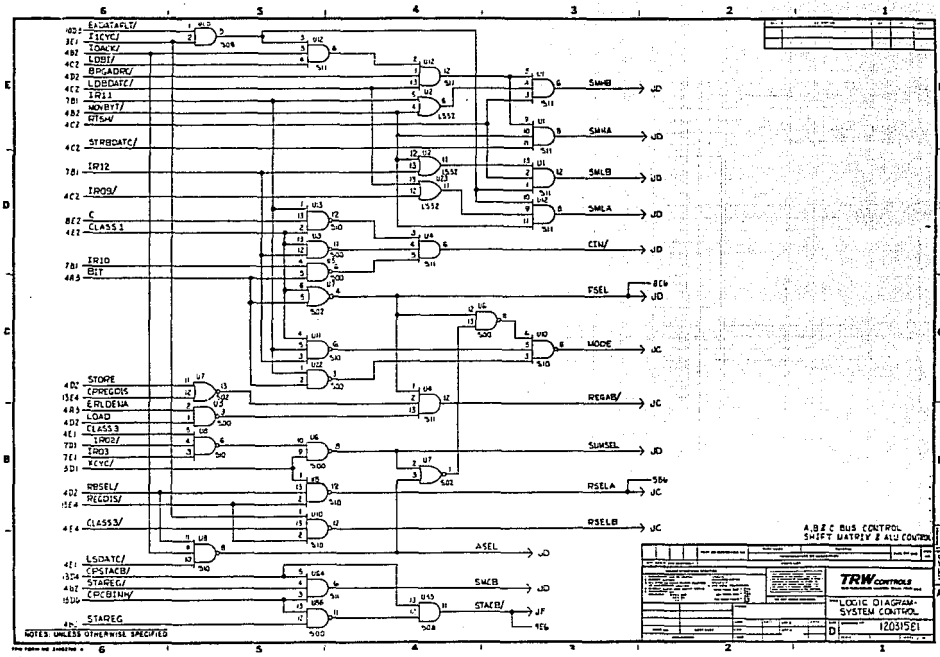

EXCATAFLT/
FICYC/
IOACK/
LDBI/
BRGADRC/
LDBDATC/
IR11
MONBYT/
BTSW/
STRBDATC/
IR12
IR09/
C
CLASS1
IR10
BIT
STORE
CPREGDIS
ERLDSHA
LOAD
CLASS3
IR02/
IR03
MCYC/
RBSEL/
REGDIS/
CLASS3/
LSDATC/
CPSTACB/
STAREG/
CPCBINH/
STAREG
SMHB
SMHA
SMLB
SMLA
CIN/
FSEL
MODE
REGAB/
SUMSEL
RSELA
RSELB
ASEL
SMCB
STACB/
A,B,C BUS CONTROL
SHIFT MATRIX & ALU CONTROL
TRW CONTROLS
LOGIC DIAGRAM-
SYSTEM CONTROL
12031581
NOTES: UNLESS OTHERWISE SPECIFIED

FIG. 4.10

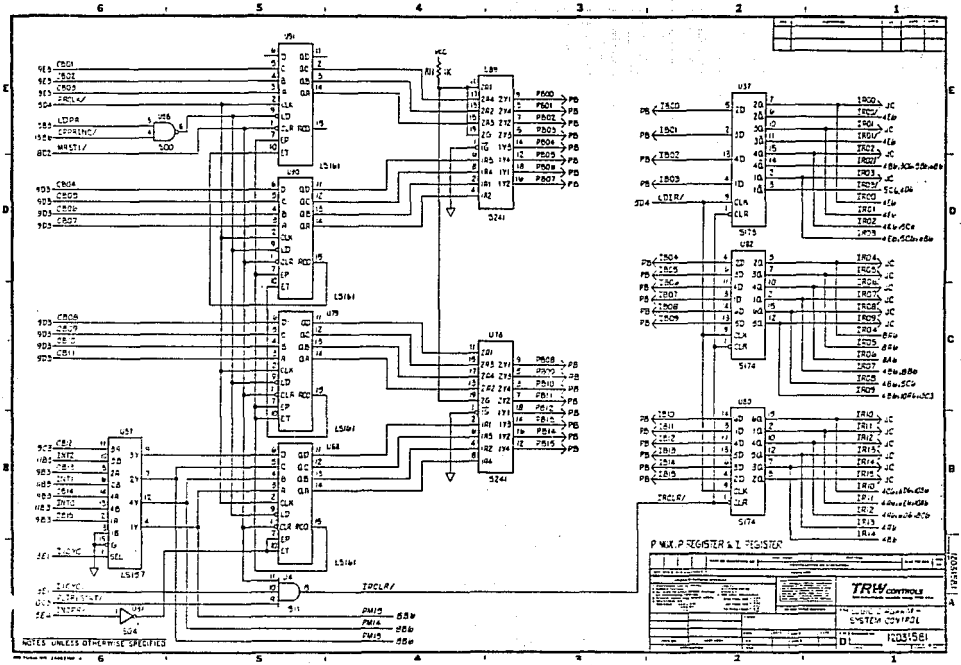

P MUX, P REGISTER & T REGISTER
TRW CONTROLS
SYSTEM 2901B2L
NOTES: UNLESS OTHERWISE SPECIFIED

FIG. 4.11

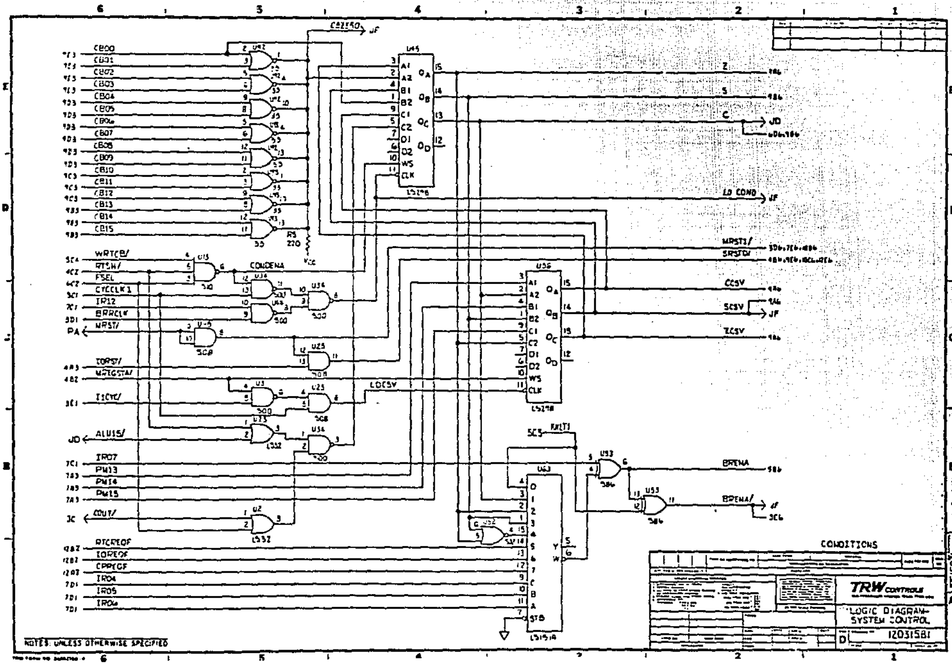

CB00
CB01
CB02
CB03
CB04
CB05
CB06
CB07
CB08
CB09
CB10
CB11
CB12
CB13
CB14
CB15
CAZERO
U45
LS298
Z
S
C
LD COND
MRST1/
SRSTO/
CCSV
SCSV
ZCSV
U56
WRTCB/
RTSH/
FSEL
CYCCLK1
IR12
BRRCLK
MRST/
IORST/
ALU15/
IR07
PM13
PM14
PM15
COUT/
RTCREOF
IOREOF
CPPREOF
IR04
IR05
IR06
CONDENA
LOCSV
BRENA
BRENA/
U63
LS151A
CONDITIONS
TRW CONTROLS
LOGIC DIAGRAM-
SYSTEM CONTROL
12031581
NOTES: UNLESS OTHERWISE SPECIFIED

FIG. 4.12

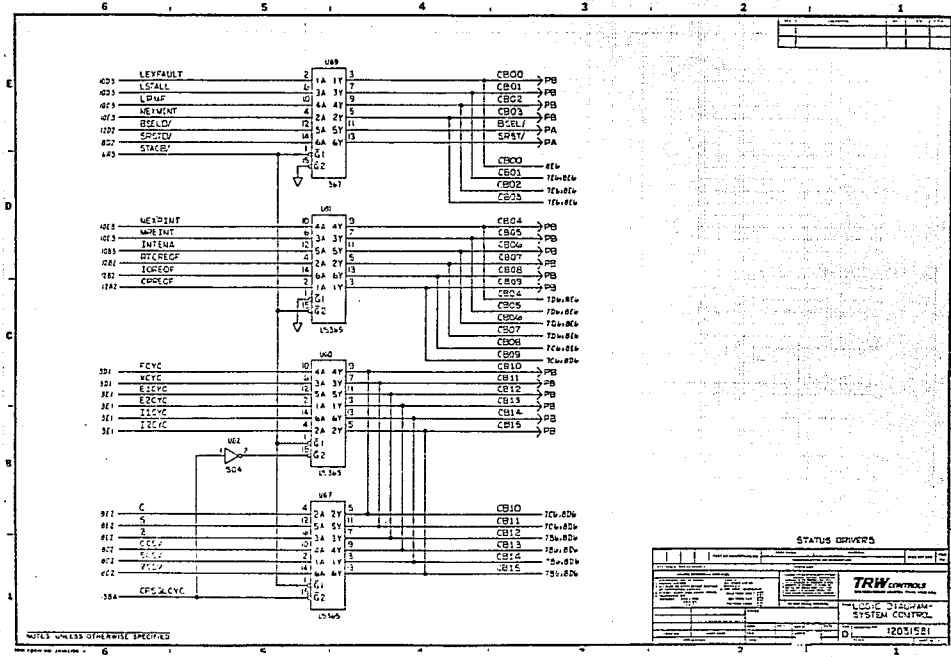

STATUS DRIVERS
TRW CONTROLS
LOGIC SYSTEM-
SYSTEM CONTROL
1205181
NOTES UNLESS OTHERWISE SPECIFIED

FIG. 4.13

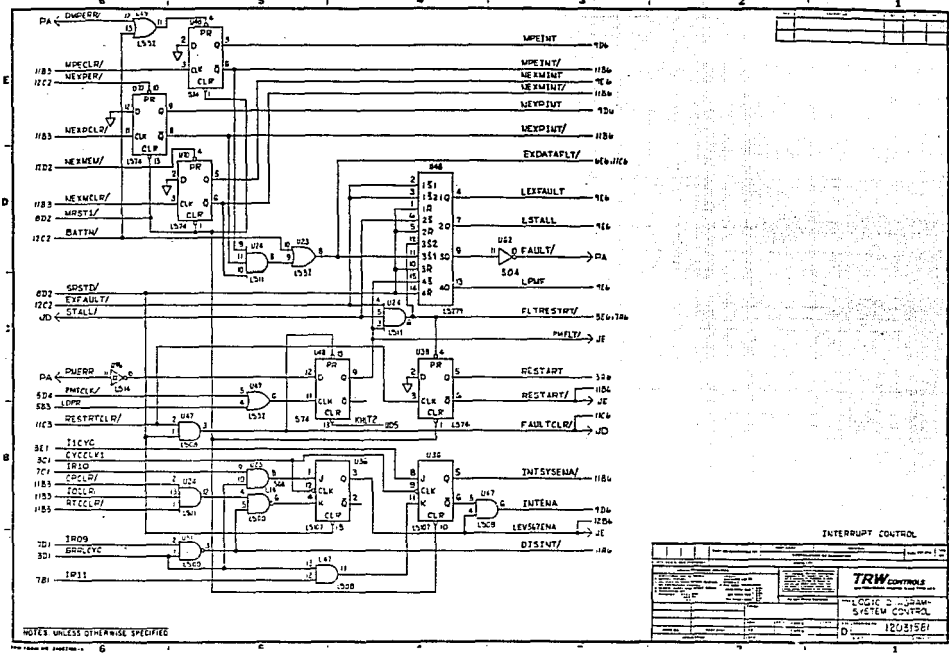

INTERRUPT CONTROL
TRW CONTROLS
LOGIC DIAGRAM
SYSTEM CONTROL
12031561
NOTES UNLESS OTHERWISE SPECIFIED

FIG. 4.14

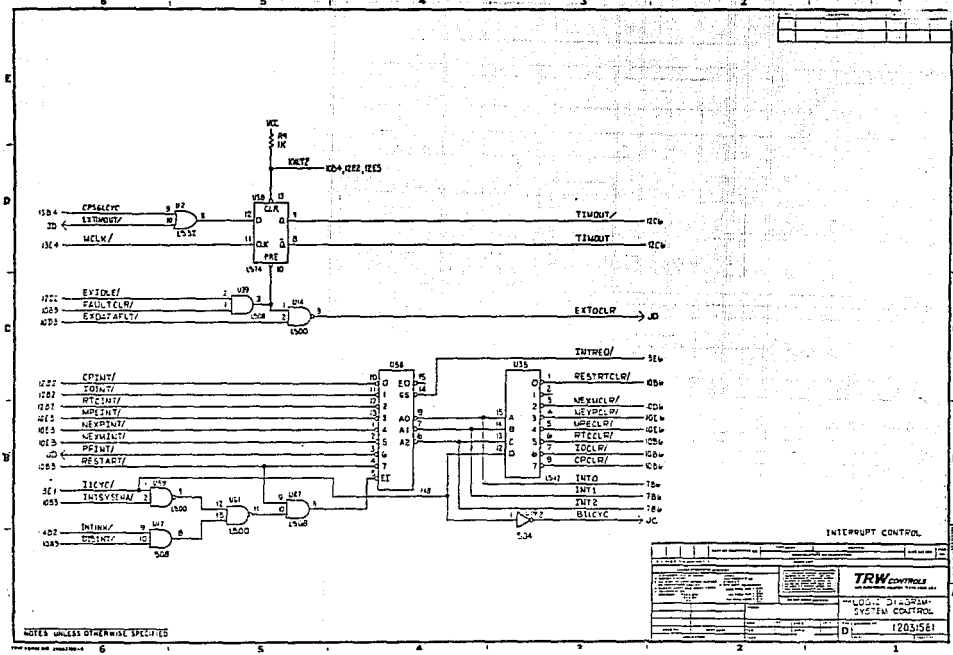

INTERRUPT CONTROL
TRW CONTROLS
LOGIC DIAGRAM
SYSTEM CONTROL
12031581
NOTES: UNLESS OTHERWISE SPECIFIED

FIG. 4.15

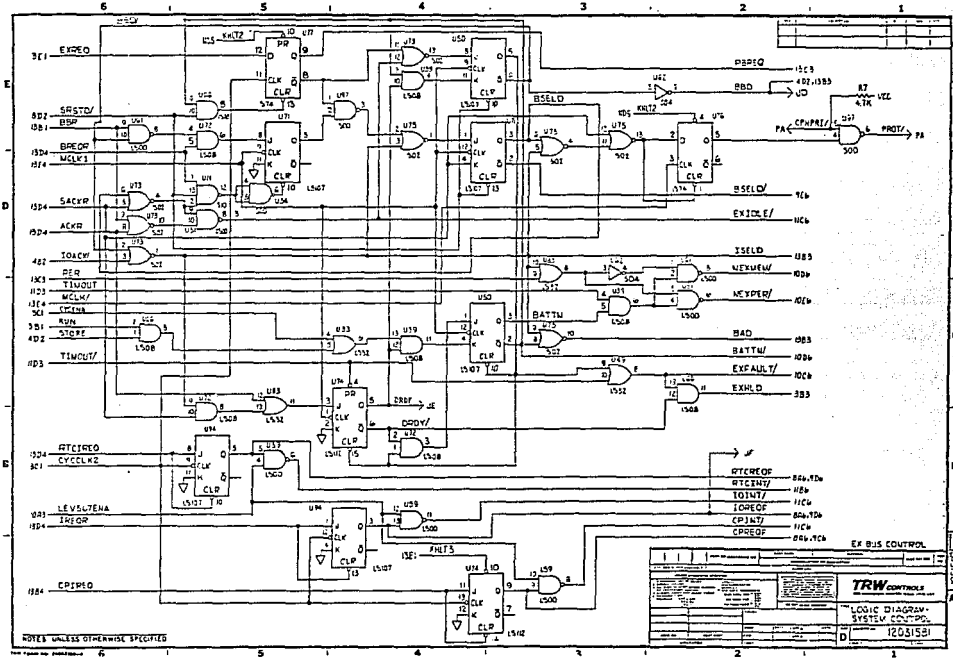

EXREQ
SRSTO/
BSP
BREQR
MCLK1
SACKR
ACKR
IOACW
PER
TIMOUT
MCLK/
CMENA
RUN
STORE
TIMOUT/
RTCREQ
CYCCLK2
LEV5&7ENA
IREQR
CPIREQ
PBREQ
BBD
BSELD
BSELD/
EXICLE/
ISELD
MEMWEN/
MEMPER/
BATTN
BAD
BATTN/
EXFAULT/
EXHLD
RREF
DRDY/
RTCREQF
RTCENT/
TOINT/
IOREQF
CPINT/
CPREQF
EX BUS CONTROL
TRW CONTROLS
LOGIC DIAGRAM-SYSTEM CONTRL
12051581
NOTES: UNLESS OTHERWISE SPECIFIED

FIG. 4.16

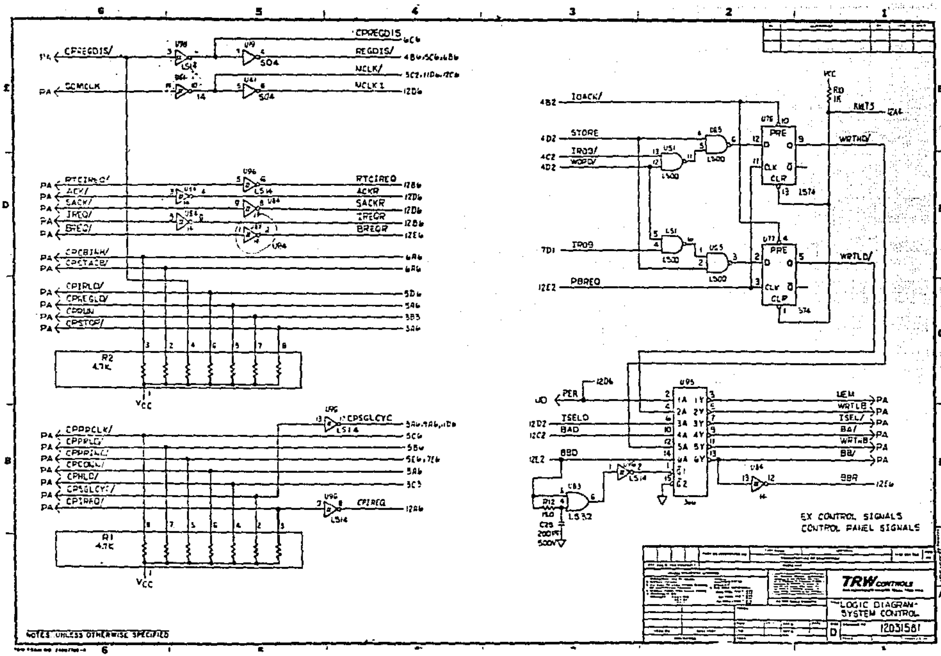

EX CONTROL SIGNALS
CONTROL PANEL SIGNALS
TRW CONTROLS
LOGIC DIAGRAM-
SYSTEM CONTROL
12031581
NOTES: UNLESS OTHERWISE SPECIFIED

FIG. 4.17

CAPITULO V

REDISEÑO DE LA TARJETA DE CONTROL DEL SISTEMA

V.1. INTRODUCCION

En general el diseñar no involucra sólo el seleccionar dispositivos y llevar a cabo una idea con ellos, implica también otros aspectos como son:

- un análisis de los elementos con que se cuenta,
- proyección del diseño a futuro,
- ventajas y desventajas,
- alternativas para desarrollarlo,
- etc.

Debido a que el trabajo de rediseño, esta basado en algo establecido, se tienen que tomar en cuenta otros aspectos, como verificar la compatibilidad de dispositivos, considerando que el rediseñar implica mejorar el diseño original.

El presente capítulo, muestra un enfoque teórico en el diseño con PLD's, abarcando desde metas, pasos y métodos a seguir, así como, cuándo un PLD debe o no debe ser usado, cómo selecionar el dispositivo adecuado, cómo seleccionar las herramientas de diseño apropiadas, cómo crear la lógica que contendrá el diseño, etc.

Posteriormente, en este mismo capítulo, se abordará el rediseño de la tarjeta de Control del Sistema en sí, explicando la secuencia seguida para lograr ésto, por medio del análisis detallado de uno de los diagramas de la tarjeta, lo cual ejemplifica el procedimiento del resto del rediseño.

V.2. DISEÑO CON PLD'S.

El enfoque de esta parte del capítulo no es dar una secuencia de pasos que deban ser seguidos siempre, como una receta de cocina. Sólo son consideraciones que pueden ser tomadas de acuerdo al criterio del diseñador. Incluso, habrá otros aspectos que tengan que ser considerados de acuerdo a las características particulares del diseño.

V.2.1. ALCANCES Y LIMITACIONES.

El primer paso para el establecimiento de un diseño tiene las siguentes consideraciones: definir el área a la cual pertenece el diseño (digital ó analógica), funciones y características a ser implementadas, requerimientos de velocidad, requerimientos de durabilidad, limitaciones en el consumo de energía, limitaciones en espacio, costo, duración del desarrollo del diseño, disponibilidad de tecnología y recursos.

Las consideraciones anteriores dan un marco de referencia para definir que alcances y limitaciones tiene el diseño, así como también seleccionar el camino más adecuado a seguir.

V.2.2. ALTERNATIVAS DE DISEÑO.

Una vez trazados los objetivos, el siguiente paso puede ser decidir que alternativa se va a utilizar, si se trata de PLD's, dispositivos convencionales o ambos. Se deben considerar lo relativo a la arquitectura de los dispositivos, espacio de la tarjeta, tiempo de desarrollo, volumen esperado de la producción y costo.

Cuando se considera el costo efectivo de las diferentes alternativas lógicas, se debe tomar en cuenta la parte de costo absoluto (el costo en el momento del diseño) y el costo a futuro (facilidad de mantenimiento, vida útil, avance tecnológico, etc).

La depuración de un circuito es generalmente más rápida y menos tediosa cuando se utiliza lógica programable. Esto es debido a que los dispositivos pueden ser reprogramados y reutilizados nuevamente, frecuentemente sin adición de un corte de pista y puente por medio de un alambre. Además un PLD tiene mucho más capacidad para implementar una lógica dada en un sólo dispositivo, la cual se construiría con un número mayor de dispositivos convencionales.

Esta etapa del diseño es importante debido a que de aquí en adelante se definen los pasos a seguir. Esta es la razón por la que el diseñador debe conocer las alternativas lógicas con que cuenta. Una vez que se está seguro que los PLD's son la mejor alternativa, el siguiente paso es determinar cual es el mejor PLD a utilizar.

V.2.3. SELECCION DEL DISPOSITIVO ADECUADO.

Las consideraciones para la selección del PLD (o PLD's) a usar se pueden dividir en tres categorías: funcionalidad, soporte de desarrollo y diseño y aspectos de manufactura.

La funcionalidad involucra las especificaciones del dispositivo. Estas incluyen arquitectura, velocidad, consumo de energía, tipos de salida (registrada o combinacional), capacidad de lógica, número de pines de entrada/salida y rangos de operación de temperatura. Otras especificaciones y características también pueden ser importantes, tal como borrabilidad y capacidad de programación.

Dentro de las consideraciones de soporte de desarrollo y diseño se debe tomar en cuenta la disponibilidad de las herramientas que soportarán al diseño, esto es, tener acceso a uno o más compiladores lógicos y programadores de dispositivos, así como también tener acceso a otras ayudas, tales como simulación y minimización. El software a utilizar debe proporcionar diferentes alternativas, es decir, poder aceptar captura esquemática,

ecuaciones booleanas, tablas de verdad o máquinas de estado, según sean los requerimientos del diseño. Otra parte del soporte debe considerar las características del programador a utilizar y las capacidades que éste tenga con respecto al dispositivo.

En cuanto a la manufactura, es importante observar las características del dispositivo a usar en lo que se refiere a costo, rentabilidad, durabilidad, disponibilidad, compatibilidad con los soportes de software y hardware, reprogramabilidad, etc.

La selección del dispositivo adecuado puede variar a lo largo del diseño, esto es, dependiendo de las circunstancias que se presenten.

V.2.4. DETERMINACION DE FORMAS DE DISEÑO.

Para definir la lógica del diseño, en un soporte de software, se utilizan varias formas, las cuales son proporcionadas por el programa, cada forma tiene sus características particulares, así como sus ventajas y desventajas. La manera de diseñar queda determinada por las características del diseño, estas formas se resumen en cinco: Ecuaciones Booleanas, Tablas de Verdad, Máquinas de Estado, Formas de Onda y Captura Esquemática.

La entrada por ecuaciones booleanas es la más básica y universal forma de entrada de definición lógica para compiladores, esta es soportada por casi todos los paquetes de software. En esta

forma es necesario elaborar las ecuaciones de acuerdo al formato establecido por el software a utilizar, debido a esto, las ecuaciones podrían complicarse dependiendo del tamaño de la lógica a implementar.

La entrada por tablas de verdad es conveniente para algunas aplicaciones, es más notable en codificadores (por ejemplo, BCD a 7 segmentos) y en decodificadores direccionadores con requerimientos no muy usuales. Aquí se tienen que especificar las entradas y salidas, con sus valores específicos.

Introducir un diseño por medio de máquinas de estado requiere regularmente de cuatro a ocho estados de diseño, esto es utilizado en algunas aplicaciones de lógica secuencial. El compilador es responsable de convertir las definiciones de estado en ecuaciones. En general, las máquinas de estado pueden ahorrar tiempo con respecto a una entrada por ecuaciones booleanas o una elaboración manual.

Algunos compiladores de PLD's soportan entradas por forma de onda. Ofrecen la posibilidad de observar las entradas y las salidas gráficamente. El compilador convierte las formas de onda en ecuaciones apropiadas. Tiene grandes ventajas en diseños secuenciales.

La entrada por captura esquemática ha crecido como una gran alternativa para el diseño con PLD's. Proporciona la facilidad de crear un diseño por medio de un diagrama eléctrico (en su parte lógica), mediante un adecuado soporte de software. El esquema

incluye la declaración de variables de entrada y salida, este es procesado para crear las ecuaciones correspondientes de la lógica capturada. A pesar de que ofrece gran variedad de facilidades, no siempre resulta ser la mejor alternativa.

V.2.5. OTROS ASPECTOS IMPORTANTES EN EL DISEÑO.

Una vez que se han generado las ecuaciones, algunos soportes de software son capaces de realizar una minimización, con el fin de eliminar redundancias y optimizar recursos.

El diseño puede ser simulado, si el paquete lo permite, generando vectores de prueba. En algunos casos esta es una herramienta útil para comprobar si los resultados son los esperados, pero en muchas ocaciones es muy complicado generar vectores de prueba, cuando la lógica es muy grande, en esos casos es más conveniente programar el dispositivo y probarlo físicamente.

El número de variables de entrada y salida (unidireccional o bidireccional) proporcionan el criterio para particionar el diseño. La partición determina la forma de como queda dividido el diseño en los diferentes PLD's, la lógica contenida en una partición es implementada en un sólo PLD. Es muy importante analizar la forma más óptima de agrupar las variables, de acuerdo a su interdependencia, ya que de esto depende el número de PLD's a utilizar.

Una vez que se asegura que el diseño es el adecuado y que las particiones son las mejores, se procede a generar un archivo estandar capaz de poder ser interpretado por los programadores de PLD's (dependiendo del PLD, será el programador a utilizar). Este archivo estandar es llamado JEDEC, cuya tarea es transformar la información a un mapa de fusibles, el cual proporciona los comandos adecuados para programar al PLD, manteniendo o separando nodos para construir la lógica deseada dependiendo de la arquitectura del dispositivo.

La calidad de la documentación es una parte escencial de un buen diseño de lógica programable. En ésta se deben incluir archivos fuente de cada dispositivo, identificando todas las señales de entrada y salida y describiendo ecuaciones lógicas, tablas de verdad y máquinas de estado, si en alguno de los casos éstas existieran. A la vez, formas de entrada gráfica, tales como forma de onda o captura esquemática, deben de ser complementadas con una descripción de la lógica implementada. Una buena documentación del archivo fuente permite accesar al diseño rápidamente, si es necesario.

V.3. PROCESO DE REDISEÑO.

En esta parte del capítulo se pretende dar una explicación general de los pasos que se siguieron para llevar a cabo todo el rediseño. Es necesario aclarar que no siempre las etapas seguidas en un rediseño tienen la misma secuencia que en un diseño, esto es debido a que cuando se hace un rediseño, se tienen características ya existentes, por lo tanto, algunos pasos son diferentes en la

manera de ser aplicados.

Como se mencinó en el capítulo cuatro, la tarjeta de Control del Sistema forma una parte muy importante en el funcionamiento de la PTU y es debido a las dificultades de mantenimiento y obtención de dispositivos cuando fallaba alguna parte de la tarjeta, por lo que se optó por mejorar la implementación de sus funciones dentro del equipo, es decir, rediseñarla con nueva tecnología.

Tomando en cuenta que las funciones digitales de la tarjeta estaban implementadas en 99 dispositivos lógicos TTL convencionales, el objetivo era manejar esta lógica de una manera más versatil. La alternativa más viable fue la utilización de PLD's, considerando sus características ya mencionadas y la disponibilidad en cuanto a soporte de hardware y software con que se contaba.

Además de los circuitos lógicos, la tarjeta contaba con resistencias y capacitores. La función de estos elementos no puede ser implementada en PLD's, es por esto que no fueron considerados para fines de rediseño, sin embargo, estos son considerados al final, en sus entradas y/o salidas correspondientes, con el objetivo de no afectar a niguna señal.

Tomando en cuenta que se contaba con los diagramas eléctricos y considerando que no se podía basar el rediseño únicamente en estos o en la tarjeta, se procedió a comparar la información que se tenía en papel con lo físicamente construido. La única manera de garantizar la correspondencia entre ambos fue verificando cada conexión en diagrama contra la conexión física. Utilizando un

multímetro digital se revisó la continuidad pin a pin, en todas las conexiones. En algunos puntos se encontró que no había correspondencia, así que la corrección fue considerada en el diagrama correspondiente.

La herramienta más importante que se tenía fueron los diagramas eléctricos, debido a que estos representaban el contenido de la tarjeta, por lo tanto, había que profundizar en ellos, de tal manera que se conocieran todos los dispositivos y conexiones, así como la interrelación entre diagramas. El siguiente paso fue analizar los diagramas, en los cuales no sólo existían compuertas, sino también otros elementos con un funcionamiento más sofisticado, cuya lógica podía ser anexada a la librería del paquete, otros tenían un funcionamiento eléctrico que permitía ser implementado por medio de lógica y por último elementos con una función eléctrica especial, la cual no se lograba soportar ni implementar.

En el primer caso, se analizó la lógica interna del dispositivo, se obtuvieron las ecuaciones y estas fueron introducidas en la librería del paquete, en una forma adecuada. De esta manera, el dispositivo quedó soportado por el software para su utilización posterior.

En el segundo caso, se analizó el funcionamiento del dispositivo y se determinó la forma de implementarlo por medio de lógica (por ejemplo, la función AND - alambrada, que nos proporcionan los dispositivos de colector abierto), que fuera soportada por el paquete. Además, en los dispositivos con lógica interna muy grande, la cual no podía ser interpretada por el

paquete, se optó por particionarlos, separando su lógica.

En el último caso, se determinó dejar los dispositivos originales, debido a que su función es insustituible, tal es el caso de los dispositivos Smmitt-Trigger, los cuales se utilizan como cuadradores de onda.

Por otra parte, se tomó la decisión de renombrar las variables de entrada y salida, debido a que el soporte de software no acepta variables con caracteres especiales ni muy grandes, además de que con esto se facilita su manejo (referirse al Appendice D).

El siguiente paso consistió en determinar la forma de entrada al rediseño. Considerando que se contaba con los diagramas eléctricos, la mejor forma fue la captura esquemática. Obtener las ecuaciones booleanas, tablas de verdad o máquinas de estados que representaran el rediseño, era una tarea demasiado laboriosa y tardada, además que se podía incurrir con mayor facilidad en algún error.

Cada uno de los diagramas fueron capturados con todas las consideraciones antes mencionadas, identificando las variables de entrada, salida y bidireccionales, así como también algunas auxiliares para fines de procesamiento.

Una vez capturados los diagramas, se procedió a su procesamiento, enlazando la parte de captura esquemática con la de procesamiento de archivos GATES (por medio de PLDLinks),

utilizando varias de sus partes, como son: Declaración de Variables, Ecuaciones, Reducción, Partición, Esquemático y Mapa de PLD's(*). Aquí se llevaron a cabo los pasos necesarios para llegar a la programación de los PLD's. En esta parte, se definieron los dispositivos a usar con base en la información proporcionada por el esquema capturado y la forma de partición.

En el paso anterior se obtuvo el archivo con el mapa de fusibles para ser trasladado al programador universal. En este último se llevó a cabo la programación utilizando como enlace la parte de soporte de software UNISITE(*).

Con el dispositivo programado, se realizaron pruebas físicas para comprobar la correcta implementación de la lógica en el PLD. Por una parte, era más confiable la verificación física, y por otra, utilizar la simulación por software resultaba más complicado, considerando la cantidad de variables de entrada y salida. Una vez que todos los PLD's estaban correctamente programados, el rediseño estaba concluido, sólo restaba armar la nueva tarjeta, con su respectiva etapa de prueba. Debido a que el objetivo es plantear el procedimiento para llevar a cabo el rediseño de la tarjeta, la construcción queda fuera de los alcances de esta tesis.

(*) PARA MAYOR INFORMACION ACERCA DEL SOPORTE DE SOFTWARE UTILIZADO, REFERIRSE AL CAPITULO III, PUNTO III.4., CAPITULO V, PUNTO V.3.1.2. Y BIBLIOGRAFIA No. 9, 10 Y 11.

Explicar el proceso de cada uno de los diagramas sería demasiado extenso, no obstante, la explicación detallada de un diagrama ejemplifica el procedimiento seguido para todos los demás, por lo tanto, el resto del capítulo describe los pasos seguidos para el diagrama SYSCON 7.

V.3.1. PROCEDIMIENTO DE REDISEÑO DEL DIAGRAMA SYSCON 7.

Se optó por describir el diagrama SYSCON 7 (mostrado en la figura 4.11), debido a la importancia que este tiene dentro de la tarjeta y lo difícil de su manejo, ya que su lógica está básicamente implementada en contadores. Por otra parte, este diagrama representa una forma demostrativa de visualizar el funcionamiento de los PLD's.

Esta parte de la tarjeta se encarga de direccionar la siguiente localidad del programa en memoria para ser accesada mediante el Registro P. El registro es incrementado simultaneamente con la transferencia de información al registro I, este registro retiene la instrucción que está siendo ejecutada.

El direccionamiento de una localidad fija de memoria, cualquiera que sea, puede ser provocada por una interrupción, la dirección proviene del MUX P, este habilita la ruptura de secuencia del Registro P. Las funciones de todos los registros fueron tratadas en el capítulo IV.

El diagrama SYSCON 7 lo componen 13 dispositivos lógicos, con los cuales se realizan las funciones antes mencionadas. Debido a que el direccionamiento de memoria inicia desde la localidad 8000H, es necesario mantener al bit más significativo en un nivel alto, por lo tanto, este bit permanece sujeto a Vcc mediante una resistencia. Esta resistencia es excluida en la implementación con PLD's, ya que no es un elemento lógico.

V.3.1.1. CAPTURA ESQUEMATICA.

Para llevar a cabo la captura esquemática, se determinó la forma más óptima de implementación lógica en PLD's, ya que en el proceso de captura esquemática se define la cantidad de lógica que contendran las particiones que se generarán posteriormente. En muchas ocaciones, mientras más lógica contenga un PLD, aumenta el número de pines requeridos, sin embargo, en algunos casos, es conveniente introducir más lógica para aprovechar al máximo las propiedades de reducción del software. En otras ocaciones, resulta provechoso eliminar algunas compuertas dentro de un diagrama esquemático para ganar pines , implementando la lógica removida en otros esquemáticos.

En la captura esquemática del diagrama se omiten dos compuertas y se anexa una tercera por las razones anteriormente mencionadas, estas compuertas fueron adicionadas en otra partición fuera de este análisis. Además se excluyeron los dispositivos Smith-Trigger, debido a que su función no puede ser implementada en PLD's (actualmente existen PLD's que soportan las funciones de Smith-Trigger y Colector-Abierto, pero no son muy comunes).

Las consideraciones anteriores, fueron hechas con base en un análisis previo del diagrama y la determinación de los dispositivos más convenientes. La selección del PLD adecuado fue determinada en el transcurso del rediseño.

Mediante la parte del soporte de software DASH(*), se capturó el diagrama, haciendo uso de sus librerías. Puesto que el nombre de las señales manejadas originalmente eran muy grandes y utilizaban caracteres especiales, los cuales no eran aceptados por el paquete, se decidió cambiar su nombre por uno más manejable. Además se icluyeron variables auxiliares que no formaban parte del diagrama original, pero eran necesarias para el procesamiento.

Para poder procesar el diagrama y enlazarlo hacia la parte de GATES, es necesario incluir en el esquemático una caja de declaraciones que contenga la identificación de todas las variables, esto es, asignarlas como entradas, salidas o bidireccionales. Por ejemplo, como se muestra en la figura 5.1, la variable AUX50 identifica la salida de la compuerta U86 (fig. 4.11), la cual fue removida a otra partición. Esta variable es declarada como una entrada. La variable JD13 que proviene de la salida de la compuerta 27 (fig. 5.1), la cual fue anexada al diagrama, es declarada como salida. La variable S01, que proviene

(*) PARA MAYOR INFORMACION ACERCA DEL SOPORTE DE SOFTWARE UTILIZADO, REFERIRSE AL CAPITULO III, PUNTO III.4. Y BIBLIOGRAFIA No. 8.

del pin 4 de U57 y entra al pin 3 de U68 (fig. 5.1) es declarada como bidireccional.

Cabe aclarar que todas las salidas provenientes de un registro deben ser declaradas como bidireccionales, debido a como se implementa la lógica en un flip-flop.

Las señales que esten conectadas físicamente a Vcc o tierra, se les asigna una variable especial, +5V y GND, respectivamente. Estas variables no son incluidas en la caja de declaraciones, el paquete las conecta internamente por software.

La caja de declaraciones tiene también un nombre que identificará las formas de declaración de variables y ecuaciones, que son generadas en GATES. Para nuestro caso se le llamó SC7 (SYSCON 7).

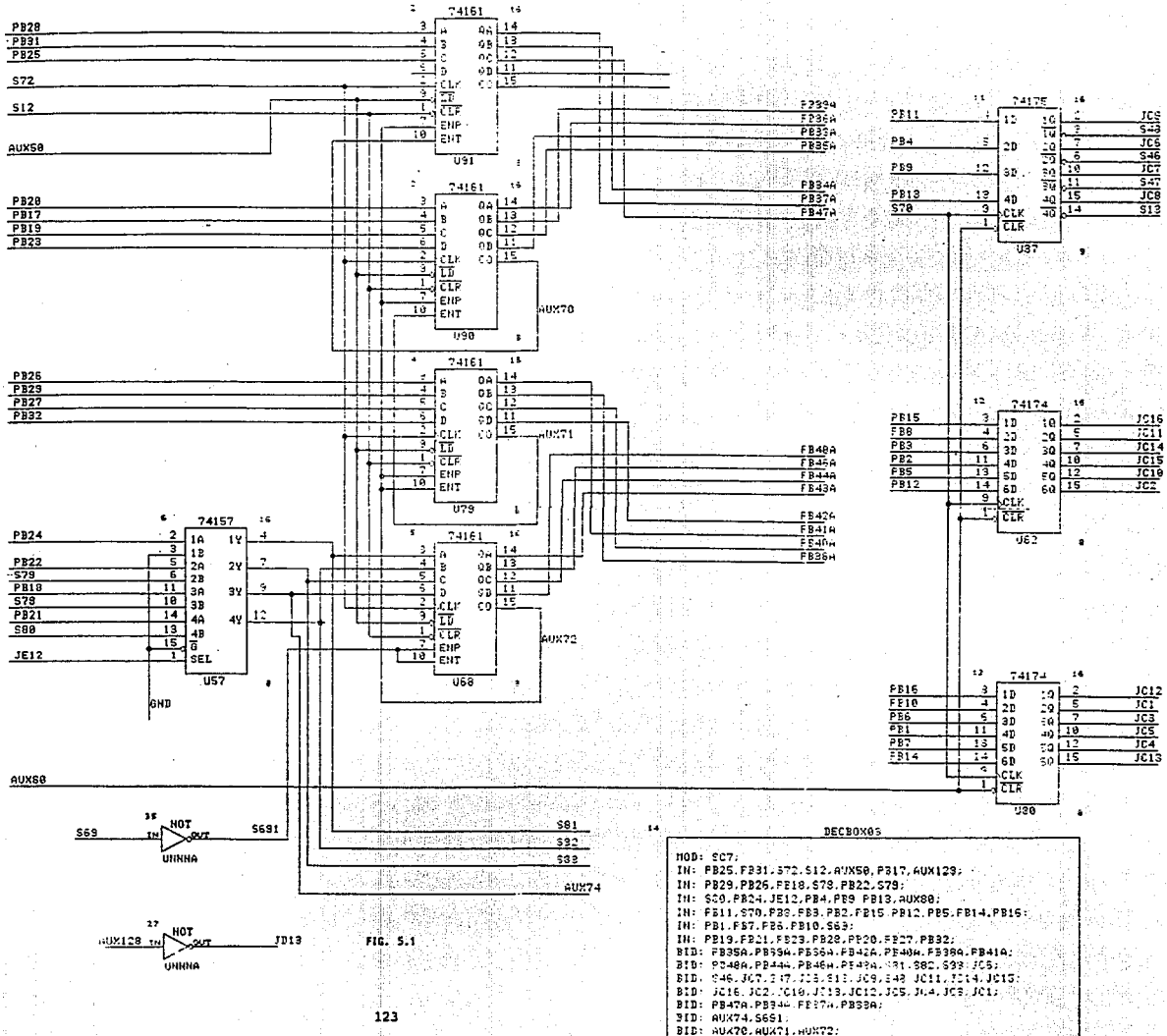

74161
U91
U90
U79
U68
74157
U57
74175
U87
74174
U62
U80
PB28
PB31
PB25
S72
S12
AUX50
PB20
PB17
PB19
PB23
AUX70
PB26
PB29
PB27
PB32
AUX71
PB24
PB22
S79
PB18
S78
PB21
S80
JE12
GND
PB11
PB4
PB9
PB13
S70
PB15
PB8
PB3
PB2
PB5
PB12
PB16
PB10
PB6
PB1
PB7
PB14
AUX72
AUX60
S69
HOT
UNNNA
S691
S81
S82
S83
AUX74
AUX128
JD13
DECBOX05
MOD: SC7;
IN: PB25,PB31,S72,S12,AUX50,PB17,AUX129;
IN: PB29,PB26,PE18,S78,PB22,S79;
IN: S80,PB24,JE12,PB4,PB9 PB13,AUX80;
IN: PB11,S70,PB8,PB3,PB2,PB15 PB12,PB5,PB14,PB16;
IN: PB1,PB7,PB6,PB10,S69;
IN: PB19,PB21,PB23,PB28,PB20,PB27,PB32;
BID: PB35A,PB39A,PB56A,PB42A,PB40A,PB38A,PB41A;
BID: PB48A,PB44A,PB46A,PB43A,S81,S82,S83;JC5;
BID: S46,JC7,JC8,JC13,S13,JC9,S48 JC11,JC14,JC15;
BID: JC16,JC2,JC10,JC12,JC5,JC4,JC3,JC1;
BID: PB47A,PB34A,PB37A,PB33A;
BID: AUX74,S691;
BID: AUX70,AUX71,AUX72;
OUT: JD13;
UNNNA

FIG. 5.1

V.3.1.2. UTILIZACION DE LAS FORMAS DE GATES.

Una vez procesado el diagrama mediante PLDLinks, toda la información es trasladada a GATES, en donde se continúa el proceso de rediseño. Mediante el menú principal (fig. 5.2), accesamos a la primera de las formas de GATES, denominada Indice, para poder visualizar los errores que se pudieron haber producido. Aquí se visualiza la identificación de las formas Declaración y Ecuación, generadas mediante el traslado de la información. Las demás formas permanecen aún vacias.

En la forma Declaración aparecen todas las variables involucradas en el diagrama, así como su tipo y definición, es decir si son registradas o combinacionales. En la forma Ecuación se encuentran las ecuaciones correspondientes a cada una de las salidas, derivadas de la lógica capturada en el diagrama. Las salidas provenientes de un registro generan varias ecuaciones dependiendo del tipo de flip-flop. Para nuestro caso, son generadas tres tipos de ecuaciónes: una dependiente de la entrada "d", una dependiente de su señal de reloj y una dependiente del reset. Para una salida combinacional se genera una sóla ecuación dependiente de las entradas involucradas.

```
F1: HELP                        GATES                        Insert: off
>>_____________
Select:      Edit      File      FOrm      Index      Quit      System
```

FIG. 5.2

Independientemente de la lógica, las ecuaciones que aparecen en la forma Ecuación se muestran en su forma de suma de productos y no han sido minimizadas, esto se hace posteriormente en la forma Reducción. Las ecuaciones fueron utilizadas para verificar la lógica capturada, lo cual representó una muy útil herramienta en el correcto proceso del diseño.

En la forma Reducción (fig. 5.3, 5.4, 5.5, 5.6) aparecen todas las ecuaciones tal como en la forma Ecuación, esta parte fué de gran utilidad, ya que al ser procesada elimina redundancias, con base en una minimización, respetando la estructura de suma de productos.

Form type: reduction Form name: Reduce

```
Reduction Level
 Polarity
  Exclusive-OR
   Display
_____________________________________________________________________
o + n r  AUX70 = AUX71 & PB36A & PB39A & PB35A & PB33A
o + n r  AUX71 = AUX72 & PB41A & PB38A & PB40A & PB42A
o + n r  AUX72 = S691 & PB43A & PB46A & PB44A & PB48A
o + n r  AUX74 = JE12 & S78 # PB18 & !JE12
o + n r  JC1.ap = !AUX80
o + n r  JC1.clk = S70
o - n r  !JC1.d = !PB10
o + n r  JC10.ap = !AUX80
o + n r  JC10.clk = S70
o - n r  !JC10.d = !PB5
o + n r  JC11.ap = !AUX80
o + n r  JC11.clk = S70
o - n r  !JC11.d = !PB8
o + n r  JC12.ap = !AUX80
o + n r  JC12.clk = S70
o - n r  !JC12.d = !PB16
o + n r  JC13.ap = !AUX80
o + n r  JC13.clk = S70
o - n r  !JC13.d = !PB14
o + n r  JC14.ap = !AUX80
o + n r  JC14.clk = S70
o - n r  !JC14.d = !PB3
o + n r  JC15.ap = !AUX80
o + n r  JC15.clk = S70
o - n r  !JC15.d = !PB2
o + n r  JC16.ap = !AUX80
o + n r  JC16.clk = S70
o - n r  !JC16.d = !PB15
o + n r  JC2.ap = !AUX80
o + n r  JC2.clk = S70
o - n r  !JC2.d = !PB12
o + n r  JC3.ap = !AUX80
o + n r  JC3.clk = S70
o - n r  !JC3.d = !PB6
o + n r  JC4.ap = !AUX80
o + n r  JC4.clk = S70
o - n r  !JC4.d = !PB7
o + n r  JC5.ap = !AUX80
o + n r  JC5.clk = S70
o - n r  !JC5.d = !PB1
o + n r  JC6.ap = !AUX80
o + n r  JC6.clk = S70
o - n r  !JC6.d = !PB4
o + n r  JC7.ap = !AUX80
o + n r  JC7.clk = S70
o - n r  !JC7.d = !PB9
o + n r  JC8.ap = !AUX80
o + n r  JC8.clk = S70
```

FIG. 5.3

```
e - n r  !JC8.d = !PB13
e + n r  JC9.ap = !AUX80
e + n r  JC9.clk = S70
e - n r  !JC9.d = !PB11
e + n r  JD13 = !AUX128
e + n r  PB33A.ap = !S12
e + n r  PB33A.clk = S72
e - n r  !PB33A.d =
           AUX50 & AUX71 & PB33A & AUX72 & PB36A & PB39A & PB35A
         # !PB23 & !AUX50
         # AUX50 & !PB33A & !PB35A
         # AUX50 & !PB33A & !PB39A
         # AUX50 & !PB33A & !PB36A
         # AUX50 & !PB33A & !AUX72
         # AUX50 & !AUX71 & !PB33A
e + n r  PB34A.ap = !S12
e + n r  PB34A.clk = S72
e - n r  !PB34A.d =
           AUX50 & AUX70 & PB34A & AUX72 & PB37A
         # !PB31 & !AUX50
         # AUX50 & !PB34A & !PB37A
         # AUX50 & !PB34A & !AUX72
         # AUX50 & !AUX70 & !PB34A
e + n r  PB35A.ap = !S12
e + n r  PB35A.clk = S72
e - n r  !PB35A.d =
           AUX50 & AUX71 & PB35A & AUX72 & PB36A & PB39A
         # !PB19 & !AUX50
         # AUX50 & !PB35A & !PB39A
         # AUX50 & !PB35A & !PB36A
         # AUX50 & !PB35A & !AUX72
         # AUX50 & !AUX71 & !PB35A
e + n r  PB36A.ap = !S12
e + n r  PB36A.clk = S72
e - n r  !PB36A.d =
           AUX50 & AUX71 & PB36A & AUX72
         # AUX50 & !PB36A & !AUX72
         # AUX50 & !AUX71 & !PB36A
         # !PB20 & !AUX50
e + n r  PB37A.ap = !S12
e + n r  PB37A.clk = S72
e - n r  !PB37A.d =
           AUX50 & AUX70 & PB37A & AUX72
         # AUX50 & !PB37A & !AUX72
         # AUX50 & !AUX70 & !PB37A
         # !PB28 & !AUX50
e + n r  PB38A.ap = !S12
e + n r  PB38A.clk = S72
e - n r  !PB38A.d =
           AUX50 & AUX72 & PB38A & PB41A
         # AUX50 & !PB38A & !PB41A
         # AUX50 & !AUX72 & !PB38A
         # !PB29 & !AUX50
e + n r  PB39A.ap = !S12
e + n r  PB39A.clk = S72
```

FIG. 5.4

```
e - n r  !PB39A.d =
           AUX50 & AUX71 & PB39A & AUX72 & PB36A
         # !PB17 & !AUX50
         # AUX50 & !PB39A & !PB36A
         # AUX50 & !PB39A & !AUX72
         # AUX50 & !AUX71 & !PB39A
e + n r  PB40A.ap = !S12
e + n r  PB40A.clk = S72
e - n r  !PB40A.d =
           AUX50 & AUX72 & PB40A & PB41A & PB38A
         # !PB27 & !AUX50
         # AUX50 & !PB40A & !PB38A
         # AUX50 & !PB40A & !PB41A
         # AUX50 & !AUX72 & !PB40A
e + n r  PB41A.ap = !S12
e + n r  PB41A.clk = S72
e - n r  !PB41A.d =
           AUX50 & !AUX72 & !PB41A
         # AUX50 & AUX72 & PB41A
         # !PB26 & !AUX50
e + n r  PB42A.ap = !S12
e + n r  PB42A.clk = S72
e - n r  !PB42A.d =
           AUX50 & AUX72 & PB42A & PB41A & PB38A & PB40A
         # !PB32 & !AUX50
         # AUX50 & !PB42A & !PB40A
         # AUX50 & !PB42A & !PB38A
         # AUX50 & !PB42A & !PB41A
         # AUX50 & !AUX72 & !PB42A
e + n r  PB43A.ap = !S12
e + n r  PB43A.clk = S72
e - n r  !PB43A.d =
           AUX50 & !S691 & !PB43A
         # AUX50 & S691 & PB43A
         # !S81 & !AUX50
e + n r  PB44A.ap = !S12
e + n r  PB44A.clk = S72
e - n r  !PB44A.d =
           AUX50 & S691 & PB44A & PB43A & PB46A
         # !S83 & !AUX50
         # AUX50 & !PB44A & !PB46A
         # AUX50 & !PB44A & !PB43A
         # AUX50 & !S691 & !PB44A
e + n r  PB46A.ap = !S12
e + n r  PB46A.clk = S72
e - n r  !PB46A.d =
           AUX50 & S691 & PB46A & PB43A
         # AUX50 & !PB46A & !PB43A
         # AUX50 & !S691 & !PB46A
         # !S82 & !AUX50
e + n r  PB47A.ap = !S12
e + n r  PB47A.clk = S72
e - n r  !PB47A.d =
           AUX50 & AUX70 & PB47A & AUX72 & PB37A & PB34A
         # !PB25 & !AUX50
```

FIG. 5.5

```
                # AUX50 & !PB47A & !PB34A
                # AUX50 & !PB47A & !PB37A
                # AUX50 & !PB47A & !AUX72
                # AUX50 & !AUX70 & !PB47A
e + n r  PB48A.ap = !S12
e + n r  PB48A.clk = S72
e - n r  !PB48A.d =
           AUX50 & S691 & PB48A & PB43A & PB46A & PB44A
         # !AUX74 & !AUX50
         # AUX50 & !PB48A & !PB44A
         # AUX50 & !PB48A & !PB46A
         # AUX50 & !PB48A & !PB43A
         # AUX50 & !S691 & !PB48A
e + n r  S13.ar = !AUX80
e + n r  S13.clk = S70
e - n r  !S13.d = PB13
e + n r  S46.ar = !AUX80
e + n r  S46.clk = S70
e - n r  !S46.d = PB4
e + n r  S47.ar = !AUX80
e + n r  S47.clk = S70
e - n r  !S47.d = PB9
e + n r  S48.ar = !AUX80
e + n r  S48.clk = S70
e - n r  !S48.d = PB11
e + n r  S691 = !S69
e + n r  S81 = PB24 & !JE12
e + n r  S82 = JE12 & S80 # PB21 & !JE12
e + n r  S83 = JE12 & S79 # PB22 & !JE12
```

Group Definitions

FIG. 5.6

Existe una forma de factorización, la cual es útil cuando una variable de salida excede más términos en una suma de productos, de los soportados por el dispositivo, dicha forma crea unas variables auxiliares que dividen la lógica en diferentes arreglos, haciendo posible la programación del dispositivo. Para nuestro caso no fue necesario utilizar esta forma.

A lo largo de todo el rediseño se fueron encontrando errores, los cuales fueron solucionados a su debido tiempo, revisando y corrigiendo la captura esquemática, verificando las ecuaciones y reprocesando el diagrama y las formas.

Una vez que las ecuaciónes fueron las correctas, el siguiente paso consistió en particionar(*). Para poder determinar que cantidad y tipo de lógica (combinacional o secuencial) contendría cada partición, fue necesario determinar el tipo de dispositivo a utilizar. Dado que el diagrama contenía lógica combinacional y secuencial, además de una gran cantidad de entradas y salidas, se optó por los EPLD's EP 910, para las salidas combinacionales y el CY 7C331 para salidas registradas (para consultar las carcterísticas de los dispositivos, referirse al Apendice B). El primero ofrece la cantidad de pines bidireccionales necesarios, además de una arquitectura versátil. El segundo tiene el tipo de registro y el tamaño más adecuado para implementar la lógica secuencial requerida en las particiones, es decir, su configuración interna ofrece las funciones más adecuadas para la implementación de las salidas registradas con un número de pines acorde a las necesidades de entrada y salida.

(*) PARA MAYOR INFORMACION REFERIRSE AL CAPITULO III, PUNTO III.2 Y AL PUNTO V.2.5 DE ESTE CAPITULO.

En las formas de Partición se fueron agrupando las salidas de acuerdo a su tipo, con el fin de determinar que lógica quedaría programada en los PLD's. Existe una forma con un nombre específico para cada partición. Estas formas nos dan la información de cuantos pines de entrada, salida y totales requerimos, así como las entradas para cada salida y que tipos de salida contienen.

En la primera partición, se incluyeron todas las salidas combinacionales, siendo un total de 9, las cuales requerían 22 entradas, dando un total de 31 pines. Para cada salida se especifican las entradas requeridas, el tipo de señal es bidireccional para todas, excepto para JD13, la cual es una salida. Esta partición abarca los dos inversores, el buffer selector y la parte combinacional de los contadores.

La segunda partición, contiene una parte de las salidas registradas de los flip-flops que componen el registro I, tiene 12 salidas, 10 entradas, dando un total de 22 pines, muestra las entradas correspondientes a cada salida. Podemos observar en el tipo de señal que todas son bidireccionales. En la tercera partición, se abarca la otra parte do las salidas registradas que componen el registro I, contiene 8 salidas registradas y 10 entradas que hacen un total de 18 pines.

En las particiones cuarta y quinta, se incluyen las señales del registro P formada por los cuatro contadores. La cuarta partición tiene 7 salidas y 13 entradas, para un total de 20 pines. La quinta partición tiene 8 salidas y 13 entradas, que dan un total de 21 pines.

La división de las particiones se hizo tratando de agrupar las señales que se interrelacionaban, en cuanto a tipo y función, considerando la optimización de pines a utilizar.

Tomando como base las formas de Partición, el siguiente paso fue generar sus respectivos archivos Jedec, mediante las formas de PLD-map. En estas se proporciona el tipo de PLD y el nombre de la partición, por lo tanto, existe una forma de PLD-map por cada forma de Partición. En el momento de asignar el nombre de la partición, el paquete automáticamente despliega todas las señales involucradas en esta, así como sus atributos y características: tipo de señal (entrada, salida o bidireccional), número de términos producto (de cada salida y bidireccional) y atributos para cada pin (nivel activo, puenteo del registro, tipo de registro y realimentación). En estas formas se especifica el número de pin que se quiere asignar a cada señal, esto se hace considerando la arquitectura propia del PLD y la manera más óptima de acomodar las señales, para su futura conexión con los otros PLD's.

En las figuras 5.7 a 5.11 se muestra cada una de las formas de PLD-map para cada partición, el nombre de cada forma es el mismo que el de la correspondiente forma de Partición. Al momento de procesar cada forma de PLD-map se genera un archivo Jedec (fig. 5.12 a 5.16), los cuales tienen también el mismo nombre. Estos archivos representan el resultado final de la parte de soporte de software referente a GATES, y contienen toda la información necesaria para poder programar los PLD's mediante un mapa de fusión interpretable por cualquier programador universal.

De manera adicional, GATES proporciona la facilidad de enlazarce con DASH para generar un diagrama esquemático de los dispositivos programados con sus respectivas señales, mediante la forma de Esquemático, este diagrama es representado en la figura 5.17.

Form type: pld-map Form name: sc7pa2

Partition: SC7PA2
Target Device Type: E0900 Output file format: jedec
Output File Name: sc7pa2
File Title Line:
Checksum Format: full Fast Flag: no
Simulation Form(s):
PLD-Sim? _ Trace: 0 X-level: 1 X-level: h BreakPoints:
Display Pins:

Special Fuse Number: Value:

Pin Name	Used As	Prod Terms	Pin Num	---Pin Attributes-- Act Lev	Reg Byp	Reg Typ	Feed Back
AUX128	Input		2	high	yes	com	none
AUX70	Output	1	36	high	yes	com	none
AUX71	Bidir	1	35	high	yes	com	none
AUX72	Bidir	1	34	high	yes	com	none
AUX74	Output	2	8	high	yes	com	none
JD13	Output	1	9	high	yes	com	none
JE12	Input		3	high	yes	com	none
PB18	Input		4	high	yes	com	none
PB21	Input		5	high	yes	com	none
PB22	Input		6	high	yes	com	none
PB24	Input		7	high	yes	com	none
PB33A	Input		12	high	yes	com	none
PB35A	Input		13	high	yes	com	none
PB36A	Input		14	high	yes	com	none
PB38A	Input		33	high	yes	com	none
PB39A	Input		17	high	yes	com	none
PB40A	Input		32	high	yes	com	none
PB41A	Input		31	high	yes	com	none
PB42A	Input		30	high	yes	com	none
PB43A	Input		29	high	yes	com	none
PB44A	Input		28	high	yes	com	none
PB46A	Input		27	high	yes	com	none
PB48A	Input		26	high	yes	com	none
S69	Input		19	high	yes	com	none
S691	Bidir	1	10	high	yes	com	none
S78	Input		23	high	yes	com	none
S79	Input		24	high	yes	com	none
S80	Input		37	high	yes	com	none
S81	Output	1	25	high	yes	com	none
S82	Output	2	16	high	yes	com	none
S83	Output	2	15	high	yes	com	none

FIG. 5.7

Form type: pld-map Form name: sc7pa3

Partition: SC7PA3
Target Device Type: P331 Output file format: jedec
Output File Name: sc7pa3
File Title Line:
Checksum Format: full Fast Flag: no
Simulation Form(s):
PLD-Sim? _ Trace: 0 X-level: 1 Z-level: h Breakpoints:
Display Pins:

Special Fuse Number: Value:

Pin Name	Used As	Prod Terms	Pin Num	---Pin Attributes-- Act Lev	Reg Byp	Reg Typ	Feed Back
AUX80	Input		1	high	yes	com	none
JC11	Output	1	28	high	no	d	reg
JC14	Output	1	27	high	no	d	reg
JC15	Output	1	26	high	no	d	reg
JC16	Output	1	25	high	no	d	reg
JC6	Output	1	24	high	no	d	reg
JC7	Output	1	23	high	no	d	reg
JC8	Output	1	20	high	no	d	reg
JC9	Output	1	19	high	no	d	reg
PB11	Input		2	high	yes	com	none
PB13	Input		3	high	yes	com	none
PB15	Input		4	high	yes	com	none
PB2	Input		5	high	yes	com	none
PB3	Input		6	high	yes	com	none
PB4	Input		7	high	yes	com	none
PB8	Input		9	high	yes	com	none
PB9	Input		10	high	yes	com	none
S13	Output	1	18	high	no	d	reg
S46	Output	1	17	high	no	d	reg
S47	Output	1	16	high	no	d	reg
S48	Output	1	15	high	no	d	reg
S70	Input		11	high	yes	com	none

FIG. 5.8

```
      Form type:  pld-map          Form name:  sc7pa4

Partition: SC7PA4
Target Device Type: P331          Output file format: jedec
Output File Name: sc7pa4
File Title Line:
Checksum Format: Full             Fast Flag: no
Simulation Form(s):
PLD-Sim? _   Trace: 0  X-level: 1  Z-level: h  Breakpoints:
             Display Pins:

          Special Fuse Number:          Value:
```

Pin Name	Used As	Prod Terms	Pin Num	Act Lev	Reg Byp	Reg Typ	Feed Back
				---Pin Attributes---			
AUX80	Input		1	high	yes	com	none
JC1	Output	1	28	high	no	d	reg
JC10	Output	1	27	high	no	d	reg
JC12	Output	1	26	high	no	d	reg
JC13	Output	1	25	high	no	d	reg
JC2	Output	1	24	high	no	d	reg
JC3	Output	1	23	high	no	d	reg
JC4	Output	1	20	high	no	d	reg
JC5	Output	1	19	high	no	d	reg
PB1	Input		2	high	yes	com	none
PB10	Input		3	high	yes	com	none
PB12	Input		4	high	yes	com	none
PB14	Input		5	high	yes	com	none
PB16	Input		6	high	yes	com	none
PB5	Input		7	high	yes	com	none
PB6	Input		9	high	yes	com	none
PB7	Input		10	high	yes	com	none
S70	Input		11	high	yes	com	none

FIG. 5.9

Form type: pld-map Form name: sc7pa5

Partition: SC7PA5
Target Device Type: PJ31 Output file format: jedec
Output File Name: sc7pa5
File Title Line:
Checksum Format: full Fast Flag: no
Simulation Form(s):
PLD-Sim? _ Trace: 0 X-Level: 1 Z-Level: h BreakPoints:
Display Pins:

Special Fuse Number: Value:

Pin Name	Used As	Prod Terms	Pin Num	---Pin Attributes-- Act Lev	Reg Byp	Reg Typ	Feed Back
AUX50	Input		1	high	yes	com	none
AUX70	Input		2	high	yes	com	none
AUX71	Input		3	high	yes	com	none
AUX72	Input		4	high	yes	com	none
PB17	Input		5	high	yes	com	none
PB19	Input		6	high	yes	com	none
PB20	Input		7	high	yes	com	none
PB23	Input		9	high	yes	com	none
PB25	Input		10	high	yes	com	none
PB28	Input		11	high	yes	com	none
PB31	Input		12	high	yes	com	none
PB33A	Bidir	7	27	high	no	d	reg
PB34A	Bidir	5	26	high	no	d	reg
PB35A	Bidir	6	25	high	no	d	reg
PB36A	Bidir	4	24	high	no	d	reg
PB37A	Bidir	4	23	high	no	d	reg
PB39A	Bidir	5	20	high	no	d	reg
PB47A	Bidir	6	19	high	no	d	reg
S12	Input		13	high	yes	com	none
S72	Input		14	high	yes	com	none

FIG. 5.10

```
Form type: pld-map          Form name: sc7pa6

Partition: SC7PA6
Target Device Type: P331        Output file format: jedec
Output File Name: sc7pa6
File Title Line:
Checksum Format: full           Fast Flag: no
Simulation Form(s):
PLD-Sim? _  Trace: 0  X-level: 1  Z-level: h  Breakpoints:
            Display Pins:

        Special Fuse Number:          Value:
                ____                  _____
```

Pin Name	Used As	Prod Terms	Pin Num	---Pin Attributes--- Act Lev	Reg Byp	Reg Typ	Feed Back
AUX50	Input	___	1	high	yes	com	none
AUX72	Input	___	2	high	yes	com	none
AUX74	Input	___	3	high	yes	com	none
PB26	Input	___	4	high	yes	com	none
PB27	Input	___	5	high	yes	com	none
PB29	Input	___	6	high	yes	com	none
PB32	Input	___	7	high	yes	com	none
PB38A	Bidir	4	28	high	no	d	reg
PB40A	Bidir	5	27	high	no	d	reg
PB41A	Bidir	3	26	high	no	d	reg
PB42A	Bidir	6	25	high	no	d	reg
PB43A	Bidir	3	24	high	no	d	reg
PB44A	Bidir	5	23	high	no	d	reg
PB46A	Bidir	4	20	high	no	d	reg
PB48A	Bidir	6	19	high	no	d	reg
S12	Input	___	9	high	yes	com	none
S691	Input	___	10	high	yes	com	none
S72	Input	___	11	high	yes	com	none
S81	Input	___	12	high	yes	com	none
S82	Input	___	13	high	yes	com	none
S83	Input	___	14	high	yes	com	none

FIG. 5.11

```
*GATES(tm) Version 5.01 JEDEC file for: E0900
Created on: 06-Feb-92 01:24 PM*
QP40* QF17402* QV0* F0*
NOTE Table of pin names and numbers*
NOTE PINS AUX128:2 AUX70:36 AUX71:35 AUX72:34 AUX74:8 JD13:9 JE12:3 PB18:4*
NOTE PINS PB21:5 PB22:6 PB24:7 PB33A:12 PB35A:13 PB36A:14 PB38A:33 PB39A:17*
NOTE PINS PB40A:32 PB41A:31 PB42A:30 PB43A:29 PB44A:28 PB46A:27 PB48A:26 S69:19*
NOTE PINS S691:10 S76:23 S79:24 S80:57 S81:25 S82:16 S83:15*
L0000 11111111111111111111101010111101111111111111111111111111111101111111111*
L0648 11111111111111111111111111111111111111111111111111111111111111111111111*
L0720 11111111111111111111111111111111111111111111111111101010101011111111111*
L1368 11111111111111111111111111111111111111111111111111111111111111111111111*
L1440 11111111111111111101111111111111111111111111111101010101011111111111111*
L2088 11111111111111111111111111111111111111111111111111111111111111111111111*
L7920 11011111111101111111111111111111111111111111111111111111111111111111111*
L8568 11111111111111111111111111111111111111111111111111111111111111111111111*
L10800 11101111111111111111111111111111111111111101111111111111111111111111111*
L10872 11011011111111111111111111111111111111111111111111111111111111111111111*
L11448 11111111111111111111111111111111111111111111111111111111111111111111111*
L11520 01111111111111111111111111111111111111111111111111111111111111111111111*
L12168 11111111111111111111111111111111111111111111111111111111111111111111111*
L12240 11111111111111111111111111111111111101111111111111111111111111111111111*
L12888 11111111111111111111111111111111111111111111111111111111111111111111111*
L15840 11101111111111111111111111111111111111111101111111111111111111111111111*
L15912 11011111101111111111111111111111111111111111111111111111111111111111111*
L16488 11111111111111111111111111111111111111111111111111111111111111111111111*
L16560 11101111111111111111111111111111111111111111111111111111111111111111101*
L16632 11011101111111111111111111111111111111111111111111111111111111111111111*
L17208 11111111111111111111111111111111111111111111111111111111111111111111111*
L17280 10000*
L17285 10000*
L17290 10000*
L17335 10000*
L17355 10000*
L17360 10000*
L17365 10000*
L17390 10000*
L17395 10000*
L17400 11*
C88E3*
V77D
```

FIG. 5.12

```
*GATES(tm) Version 5.01 JEDEC file for: P331
Created on: 18-Jan-92 11:32 AM*
QP20* QF11934* QV0* F0*
NOTE Table of pin names and numbers*
NOTE PINS AUX60:1 JC11:20 JC14:27 JC15:26 JC16:25 JC6:24 JC7:23 JC8:20*
NOTE PINS JC9:19 PB11:2 PB13:3 PB15:4 PB2:5 PB3:6 PB4:7 PB8:9*
NOTE PINS PB9:10 S13:18 S46:17 S47:16 S48:15 S70:11*
L0000 111111111111111111111111111111111111111111111111111111111111111111111111*
L0062 101111111111111111111111111111111111111111111111111111111111111111111111*
L0124 111111111111111111111111111111111111111111111101111111111111111111111111*
L0434 111111111111111111111111111111111111101111111111111111111111111111111111*
L0744 111111111111111111111111111111111111111111111111111111111111111111111111*
L0806 101111111111111111111111111111111111111111111111111111111111111111111111*
L0868 111111111111111111111111111111111111111111111101111111111111111111111111*
L1178 111111111111111111111101111111111111111111111111111111111111111111111111*
L1984 111111111111111111111111111111111111111111111111111111111111111111111111*
L2046 101111111111111111111111111111111111111111111111111111111111111111111111*
L2108 111111111111111111111111111111111111111111111101111111111111111111111111*
L2418 111111111111111111111101111111111111111111111111111111111111111111111111*
L2852 111111111111111111111111111111111111111111111111111111111111111111111111*
L2914 101111111111111111111111111111111111111111111111111111111111111111111111*
L2976 111111111111111111111111111111111111111111111101111111111111111111111111*
L3286 111111111111101111111111111111111111111111111111111111111111111111111111*
L3968 111111111111111111111111111111111111111111111111111111111111111111111111*
L4030 101111111111111111111111111111111111111111111111111111111111111111111111*
L4092 111111111111111111111111111111111111111111111101111111111111111111111111*
L4402 111111111111111111111111111111111101111111111111111111111111111111111111*
L4960 111111111111111111111111111111111111111111111111111111111111111111111111*
L5022 101111111111111111111111111111111111111111111111111111111111111111111111*
L5084 111111111111111111111111111111111111111111111101111111111111111111111111*
L5394 111111111111111111111111111111111111111111111101111111111111111111111111*
L5952 111111111111111111111111111111111111111111111111111111111111111111111111*
L6014 101111111111111111111111111111111111111111111111111111111111111111111111*
L6076 111111111111111111111111111111111111111111111102111111111111111111111111*
L6386 111111111101111111111111111111111111111111111111111111111111111111111111*
L5944 111111111111111111111111111111111111111111111111111111111111111111111111*
L7006 101111111111111111111111111111111111111111111111111111111111111111111111*
L7068 111111111111111111111111111111111111111111111101111111111111111111111111*
L7378 111101111111111111111111111111111111111111111111111111111111111111111111*
L7936 111111111111111111111111111111111111111111111111111111111111111111111111*
L8060 111111111111111111111111111111111111111111111101111111111111111111111111*
L8122 101111111111111111111111111111111111111111111111111111111111111111111111*
L8370 111111111101111111111111111111111111111111111111111111111111111111111111*
L9052 111111111111111111111111111111111111111111111111111111111111111111111111*
L9176 111111111111111111111111111111111111111111111101111111111111111111111111*
L9238 101111111111111111111111111111111111111111111111111111111111111111111111*
L9486 111111111111111111111111111111111101111111111111111111111111111111111111*
L9920 111111111111111111111111111111111111111111111111111111111111111111111111*
L10044 111111111111111111111111111111111111111111111101111111111111111111111111*
L10106 101111111111111111111111111111111111111111111111111111111111111111111111*
L10354 111111111111111111111111111111111111111111111101111111111111111111111111*
L11160 111111111111111111111111111111111111111111111111111111111111111111111111*
L11284 111111111111111111111111111111111111111111111101111111111111111111111111*
L11346 101111111111111111111111111111111111111111111111111111111111111111111111*
L11594 111011111111111111111111111111111111111111111111111111111111111111111111*
C6B2E*
*D44B
```

FIG. 5.13

```
*GATES(tm) Version 5.01 JEDEC file for: P331
Created on: 18-Jan-92 11:33 AM*
QP28* QF11934* QV0* F0*
NOTE Table of pin names and numbers*
NOTE PINS AUX80:1 JC1:28 JC10:27 JC12:26 JC13:25 JC2:24 JC3:23 JC4:20*
NOTE PINS JC5:19 PB1:2 PB10:3 PB12:4 PB14:5 PB16:6 PB5:7 PB6:9*
NOTE PINS PB7:10 S70:1)*
L0000 111111111111111111111111111111111111111111111111111111111111111111*
L0062 101111111111111111111111111111111111111111111111111111111111111111*
L0124 111111111111111111111111111111111111111111101111111111111111111111*
L0434 111111111110111111111111111111111111111111111111111111111111111111*
L0744 111111111111111111111111111111111111111111111111111111111111111111*
L0806 101111111111111111111111111111111111111111111111111111111111111111*
L0868 111111111111111111111111111111111111111111101111111111111111111111*
L1178 111111111111111111111111111111111101111111111111111111111111111111*
L1984 111111111111111111111111111111111111111111111111111111111111111111*
L2046 101111111111111111111111111111111111111111111111111111111111111111*
L2108 111111111111111111111111111111111111111111101111111111111111111111*
L2418 111111111111111111111111101111111111111111111111111111111111111111*
L2852 111111111111111111111111111111111111111111111111111111111111111111*
L2914 101111111111111111111111111111111111111111111111111111111111111111*
L2976 111111111111111111111111111111111111111111101111111111111111111111*
L3286 111111111111111111111101111111111111111111111111111111111111111111*
L3968 111111111111111111111111111111111111111111111111111111111111111111*
L4030 101111111111111111111111111111111111111111111111111111111111111111*
L4092 111111111111111111111111111111111111111111101111111111111111111111*
L4402 111111111111101111111111111111111111111111111111111111111111111111*
L4960 111111111111111111111111111111111111111111111111111111111111111111*
L5022 101111111111111111111111111111111111111111111111111111111111111111*
L5084 111111111111111111111111111111111111111111101111111111111111111111*
L5394 111111111111111111111111111111111101111111111111111111111111111111*
L5952 111111111111111111111111111111111111111111111111111111111111111111*
L6014 101111111111111111111111111111111111111111111111111111111111111111*
L6076 111111111111111111111111111111111111111111101111111111111111111111*
L6386 111111111111111111111111111111111111111111101111111111111111111111*
L6944 111111111111111111111111111111111111111111111111111111111111111111*
L7006 101111111111111111111111111111111111111111111111111111111111111111*
L7068 111111111111111111111111111111111111111111101111111111111111111111*
L7378 111101111111111111111111111111111111111111111111111111111111111111*
CF0E0*
♥F733*
```

FIG. 5.14

```
@GATES(tm) Version 5.01 JEDEC file for: P331
Created on: 18-Jan-92 11:34 AM*
QP28* QF11934* QV0* F0*
NOTE Table of pin names and numbers*
NOTE PINS AUX50:1 AUX70:2 AUX71:3 AUX72:4 PB17:5 PB19:6 PB20:7 PB23:9*
NOTE PINS PB25:10 PB28:11 PB31:12 PB33A:27 PB34A:26 PB35A:25 PB36A:24 PB37A:23*
NOTE PINS PB39A:20 PB47A:19 S12:13 S72:14*
L0744 1111111111111111111111111111111111111111111111111111111111111111*
L0806 1111111111111111111111111111111111111111111111111111111111111101*
L0868 1111111111111111111111111111111111111111111111111111111111111101*
L1054 1111111111111111111111111111111111111111111111111111111111111111*
L1116 1111111111111111111111111111111111111111111111111111111111111111*
L1240 0111111101010111111011111101111111101111111111111111111111111111*
L1302 1011111111111111111111111111111111111011111111111111111111111111*
L1364 0111111110111111111011111111111111111111111111111111111111111111*
L1426 0111111101111111111111111111111111101111111111111111111111111111*
L1488 0111111101111111111111101111111111111111111111111111111111111111*
L1550 0111111110111101111111111111111111111111111111111111111111111111*
L1612 0111111101010111111111111111111111111111111111111111111111111111*
L1984 1111111111111111111111111111111111111111111111111111111111111111*
L2046 1111111111111111111111111111111111111111111111111111111011111111*
L2108 1111111111111111111111111111111111111111111111111111111111111101*
L2294 1111111111111111111111111111111111111111111111111111111111111111*
L2356 1111111111111111111111111111111111111111111111111111111111111111*
L2480 0101111111101010111111111111011111111111111111111111111111111111*
L2542 1011111111111111111111111111111111111111111111111101111111111111*
L2604 0111111111111101111111111101111111111111111111111111111111111111*
L2666 0111111111111010111111111111111111111111111111111111111111111111*
L2728 0110111111111111111111111111111111111111111111111111111111111111*
L2852 1111111111111111111111111111111111111111111111111111111111111111*
L2914 1111111111111111111111111111111111111111111111111111011111111111*
L2976 1111111111111111111111111111111111111111111111111111111111111101*
L3162 1111111111111111111111111111111111111111111111111111111111111111*
L3224 1111111111111111111111111111111111111111111111111111111111111111*
L3348 0111111111101011111011111011111111111111111111111111111111111111*
L3410 1011111111111111111111101111111111111111111111111111111111111111*
L3472 0111111111111111111101111111111111101111111111111111111111111111*
L3534 0111111111111111110111101111111111111111111111111111111111111111*
L3596 0111111111111011111011111111111111111111111111111111111111111111*
L3658 0111111111101111110111111111111111111111111111111111111111111111*
L3968 1111111111111111111111111111111111111111111111111111111111111111*
L4030 1111111111111111111111111111111111111111111111111111111011111111*
L4092 1111111111111111111111111111111111111111111111111111111111111101*
L4278 1111111111111111111111111111111111111111111111111111111111111111*
L4340 1111111111111111111111111111111111111111111111111111111111111111*
L4464 0111111111010121111111101111111111111111111111111111111111111111*
L4526 0111111111111101111111111011111111111111111111111111111111111111*
L4588 0111111111011111111111101111111111111111111111111111111111111111*
L4650 1011111111111111111111111111111111101111111111111111111111111111*
L4960 1111111111111111111111111111111111111111111111111111111111111111*
L5022 1111111111111111111111111111111111111111111111111111111011111111*
L5084 1111111111111111111111111111111111111111111111111111111111111101*
L5270 1111111111111111111111111111111111111111111111111111111111111111*
L5332 1111111111111111111111111111111111111111111111111111111111111111*
L5456 0101111111101111111111111101111111111111111111111111111111111111*
L5518 0111111111111011111111111111101111111111111111111111111111111111*
L5580 0110111111111111111111111111111011111111111111111111111111111111*
L5642 1011111111111111111111111111111111111111111111111011111111111111*
L5952 1111111111111111111111111111111111111111111111111111111111111111*
L6014 1111111111111111111111111111111111111111111111111111111011111111*
L6076 1111111111111111111111111111111111111111111111111111111111111101*
L6262 1111111111111111111111111111111111111111111111111111111111111111*
L6324 1111111111111111111111111111111111111111111111111111111111111111*
L6448 0111111111010121111111101111111101111111111111111111111111111111*
L6510 1011111111111111111111101111111111111111111111111111111111111111*
L6572 0111111111111111111111111101111111101111111111111111111111111111*
```

FIG. 5.15

```
L6634 01111111111110111111111111111111111101111111111111111111111111*
L6696 01111111111011111111111111111111111011111111111111111111111111*
L6944 11111111111111111111111111111111111111111111111111111111111111*
L7006 11111111111111111111111111111111111111111111111111111111111111*
L7068 11111111111111111111111111111111111111111111111111111111111101*
L7254 11111111111111111111111111111111111111111111111111111111111111*
L7316 11111111111111111111111111111111111111111111111111111111111111*
L7440 01011111111110101111111111111111101111111101111111111111111111*
L7502 10111111111111111111111111111111111111111111011111111111111111*
L7564 01111111111111101111111111111111111111111101111111111111111111*
L7626 01111111111111111111111111111101111111110111111111111111111111*
L7688 01111111111101111111111111111111111111110111111111111111111111*
L7750 01101111111111111111111111111111111111110111111111111111111111*
C196C*
▼1534
```

FIG. 5.15 (CONTINUACION)

```
*GATES(tm) Version 5.01 JEDEC file for: P331
Created on: 24-Feb-92 11:58 AM*
QP28* QF11934* QV0* F0*
NOTE Table of pin names and numbers*
NOTE PINS AUX50:1 AUX72:2 AUX74:3 PB26:4 PB27:5 PB29:6 PB32:7 PB38A:28*
NOTE PINS PB40A:27 PB41A:26 PB42A:25 PB43A:24 PB44A:23 PB46A:20 PB48A:19 S12:9*
NOTE PINS S691:10 S72:11 S81:12 S82:13 S83:14*
L0000 1111111111111111111111111111111111111111111111111111111111111111*
L0062 1111111111111111111111111111111111101111111111111111111111111111*
L0124 1111111111111111111111111111111111111101111111111111111111111111*
L0310 1111111111111111111111111111111111111111111111111111111111111111*
L0372 1111111111111111111111111111111111111111111111111111111111111111*
L0496 0101010111111111011111111111111111111111111111111111111111111111*
L0558 0111101111111110111111111111111111111111111111111111111111111111*
L0620 0110101111111111111111111111111111111111111111111111111111111111*
L0682 1011111111111111111111111011111111111111111111111111111111111111*
L0744 1111111111111111111111111111111111111111111111111111111111111111*
L0806 1111111111111111111111111111111111102111111111111111111111111111*
L0868 1111111111111111111111111111111111111111101111111111111111111111*
L1054 1111111111111111111111111111111111111111111111111111111111111111*
L1116 1111111111111111111111111111111111111111111111111111111111111111*
L1240 0101011101111111111111111111111111111111111111111111111111111111*
L1302 1011111111111111111101111111111111111111111111111111111111111111*
L1364 0111101110111111111111111111111111111111111111111111111111111111*
L1426 0111111110111110111111111111111111111111111111111111111111111111*
L1488 0110111110111111111111111111111111111111111111111111111111111111*
L1984 1111111111111111111111111111111111111111111111111111111111111111*
L2046 1111111111111111111111111111111111101111111111111111111111111111*
L2108 1111111111111111111111111111111111111111111101111111111111111111*
L2294 1111111111111111111111111111111111111111111111111111111111111111*
L2356 1111111111111111111111111111111111111111111111111111111111111111*
L2480 0110111111111110111111111111111111111111111111111111111111111111*
L2542 0101111111111011111111111111111111111111111111111111111111111111*
L2604 1011111111111011111111111111111111111111111111111111111111111111*
L2852 1111111111111111111111111111111111111111111111111111111111111111*
L2914 1111111111111111111111111111111111011111111111111111111111111111*
L2976 1111111111111111111111111111111111111111101111111111111111111111*
L3162 1111111111111111111111111111111111111111111111111111111111111111*
L3224 1111111111111111111111111111111111111111111111111111111111111111*
L3348 0101011101111011011111111111111111111111111111111111111111111111*
L3410 1011111111111111111111111101111111111111111111111111111111111111*
L3472 0111111110111111111011111111111111111111111111111111111111111111*
L3534 0111101111111111111011111111111111111111111111111111111111111111*
L3596 0111111111111111011101111111111111111111111111111111111111111111*
L3658 0110111111111111111101111111111111111111111111111111111111111111*
L3968 1111111111111111111111111111111111111111111111111111111111111111*
L4030 1111111111111111111111111111111111101111111111111111111111111111*
L4092 1111111111111111111111111111111111111111111101111111111111111111*
L4278 1111111111111111111111111111111111111111111111111111111111111111*
L4340 1111111111111111111111111111111111111111111111111111111111111111*
L4464 0111111111111111111111111011111111111111111011111111111111111111*
L4526 0111111111111111111111110111111111111111101111111111111111111111*
L4588 1011111111111111111111111111111111111111111111111111111111111111*
L4960 1111111111111111111111111111111111111111111111111111111111111111*
L5022 1111111111111111111111111111111111011111111111111111111111111111*
L5084 1111111111111111111111111111111111111111111011111111111111111111*
L5270 1111111111111111111111111111111111111111111111111111111111111111*
L5332 1111111111111111111111111111111111111111111111111111111111111111*
L5456 0111111111111111111111111111101110111101111011111111111111111111*
L5518 1011111111111111111111111111111111111111111111111111111111111110*
L5580 0111111111111111111111111111111111011111101111111111111111111111*
L5642 0111111111111111111111111101110111111111111111111111111111111111*
L5704 0111111111111111111111111111111111011111111111011111111111111111*
L5952 1111111111111111111111111111111111111111111111111111111111111111*
L6014 1111111111111111111111111111111111011111111111111111111111111111*
L6076 1111111111111111111111111111111111111111111101111111111111111111*
```

FIG. 5.16

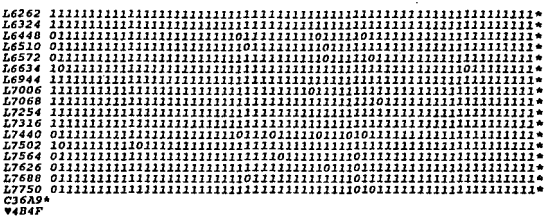
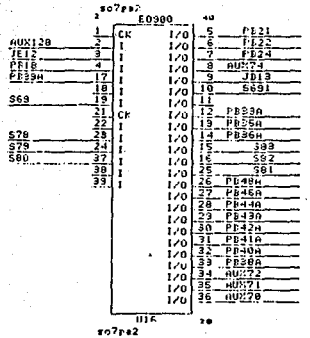
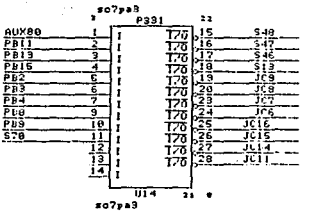
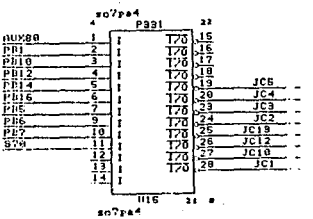
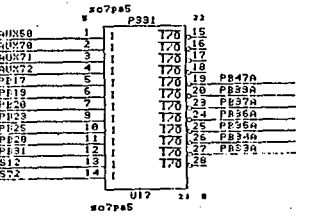


FIG. 5.16 (CONTINUACION)


so7pa2
E0380
AUX128
PB39H
S69
S78
S79
S80
PB21
PB22
PB24
AUX74
JB13
S691
PB38A
PB35A
PB36H
S83
S82
S81
PB48H
PB46A
PB44A
PB43A
PB42A
PB41A
PB40A
PB38A
AUX72
AUX71
AUX70
U16
so7pa2


so7pa3
P331
AUX80
PB11
PB13
PB15
PB2
PB3
PB4
PB8
PB9
S78
S48
S47
S46
S13
JC9
JC8
JC7
JC6
JC16
JC15
JC14
JC11
U14
so7pa3


so7pa4
P331
AUX80
PB1
PB10
PB12
PB14
PB16
PB6
PB7
S78
JC5
JC4
JC3
JC2
JC19
JC12
JC10
JC1
U16
so7pa4


so7pa5
P331
AUX60
AUX70
AUX71
AUX72
PE17
PE19
PE20
PE23
PE26
PE28
PE31
S12
S22
PB47A
PB38A
PB37H
PB36A
PB35H
PB34A
PB33A
U17
so7pa5

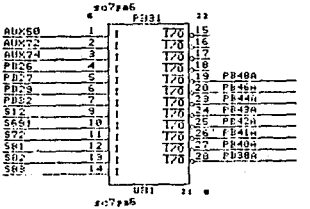

so7pa6
P331
AUX60
AUX72
AUX74
PB26
PB27
PB29
PB32
S12
S691
S22
S81
S82
S83
PB48A
PB45H
PB44H
PB43A
PB42A
PB41H
PB40H
PB38A
U11
so7pa6

FIG. 5.17

V.3.1.3. PROGRAMACION Y VERIFICACION.

Una vez teniendo los archivos Jedec, se procedió a la programación de los dispositivos. Esto se hizo mediante la parte de soporte de software UNISITE, para poder comunicar a la PC con un programador y transferir el archivo.

Mediante el menú de UNISITE, se seleccionó el dispositivo a programar, dependiendo del respectivo Jedec, con el fin de que el paquete reconociera el tipo de dispositivo. Una vez seleccionado el dispositivo, se transfirió el archivo Jedec de PC a la RAM del programador. De este modo, la tarea final en cuanto a software, fué activar la programación por medio de los comandos del paquete, quedando así el PLD programado.

Teniendo todos los PLD's programados, la siguiente tarea consistió en verificar que la programación fuera correcta, armando el circuito y suministrandole señales de prueba, para poder comparar los resultados con el análisis de la lógica del diagrama. Con esta última parte, el proceso de rediseño quedó concluido, dejando fuera de los alcances de esta tesis sólo la construcción de la tarjeta y los problemas que esto involucre.

En esta última parte del capítulo, se planteó el proceso de rediseño de sólo una parte de la tarjeta y se mostró su respectiva documentación, para el resto de los diagramas el proceso fué de manera similar y la documentación más importante de cada diagrama es incluida,en el Apendice A.

CAPITULO VI

CONCLUSIONES

VI.1. CONCLUSIONES.

El diseño en general, presenta siempre problemas a lo largo de todo el proceso, tal fue el caso del rediseño de la tarjeta de Control del Sistema, que algunas veces no dió resultados correctos, debido a errores que fueron, ya sea reportados por el paquete o detectados por la verificación realizada en cada etapa. Esto ocacionó el tener que realizar nuevamente una secuencia de pasos, con sus correcciones pertinentes, por medio del análisis del diagrama correspondiente, verificación de librerías, revisión de arquitecturas de dispositivos, etc., para lograr el correcto rediseño, tomando en cuenta los propósitos fijados con base en la estructura de la tarjeta de Control del Sistema.

El total de dispositivos EPLD's programados fue de 30, de los cuales 10 fueron EP 910, 9 EP 610 y 11 CY 7C331 (referise al Apendice B). La demás lógica que no pudo ser soportada, fue incluida en 5 dispositivos TTL, siendo un total de 35 circuitos integrados. Esto representa la reducción de la cantidad de dispositivos en aproximadamente una tercera parte, lo cual implica menor cantidad de pistas impresas en la tarjeta, así como la reducción de su tamaño, dejando la posibilidad de hacer más pequeñas las dimensiones del equipo, además de disminuir el riesgo

de fallas y facilitando el mantenimiento de la tarjeta.

Tomando en cuenta la capacidad de reprogramación de los PLD's, el remplazamiento y reutilización de dispositivos se hace más sencillo y útil, ya que, si un CI se detecta en falla, sólo se tiene que retirar de la tarjeta, reprogramarse y ser puesto en operación nuevamente. Por otra parte, se puede modificar la lógica del dispositivo para diferentes fines.

En lo referente al costo total de la tarjeta, se deben apreciar varios aspectos, uno es el valor de los PLD's en comparación con los dispositivos convencionales, otro es el costo que implica la construcción de la tarjeta con PLD's en comparación con dispositivos TTL convencionales. Un último aspecto a considerar es el costo de la tarjeta a largo plazo.

Actualmente un PLD es más caro que un dispositivo TTL desde el punto de vista precio por unidad de dispositivo (como se puede observar en las cotizaciones presentadas en el Apendice C), sin embargo, considerando la capacidad de lógica contenida en cada CI, es claro que mientras un dispositivo convencional sólo cuenta con una cierta lógica, un PLD tiene la capacidad de absorber mucho mayor lógica que tendría que ser implementada con varios CI TTL, aún más, el mismo PLD puede variar su función dependiendo de su aplicación, para lo cual se necesitaría agregar más dispositivos convencionales, para implementar dicha lógica, por lo tanto, mientras los dispositivos TTL tienen una lógica interna rígida, los PLD's presentan una lógica interna flexible.

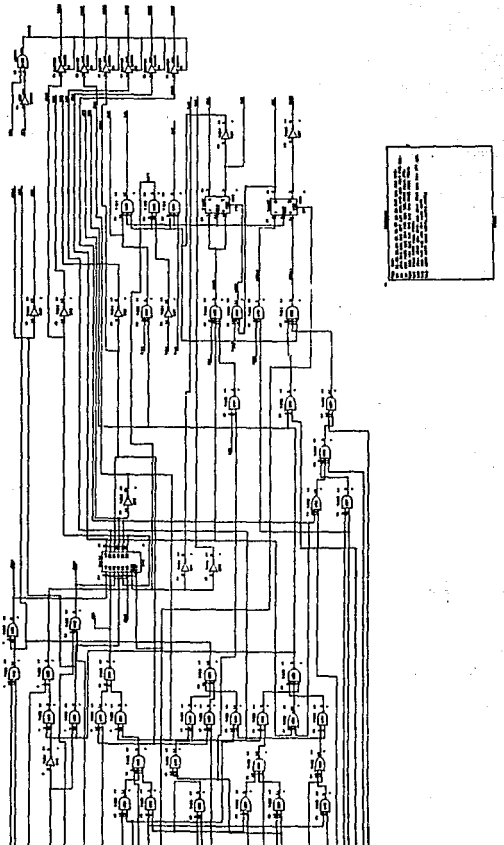
El costo de una tarjeta impresa está en función de las características requeridas, de las cuales, las más importantes son el número de dispositivos y la cantidad de pistas; con base en este argumento, el precio de la tarjeta que requiere el rediseño es menor, tomando en cuenta el resultado obtenido.

Con el avance de la tecnología, los dispositivos existentes se vuelven obsoletos y por lo tanto caros. Los equipos tienen que irse actualizando para no devaluarse, el rediseño hace que la tarjeta de Control del Sistema no pierda su valor a largo plazo, considerando que los Dispositivos de Lógica Programable son lo más actual en cuanto a diseño de lógica se refiere.

En general, considerando los aspectos anteriormente mencionados, podemos concluir que el rediseño de la tarjeta de Control del Sistema presenta grandes ventajas con respecto a la tarjeta original, dejando un precedente para aplicación y metodología en cuanto a diseño de lógica digital se refiere, además de sentar las bases para la utilización de PLD's en el Instituto Mexicano del Petróleo y tratando de motivar a todo el que pretenda incursionar en el campo de la lógica programable.

APENDICE A

DOCUMENTACION DE FORMAS Y DIAGRAMAS



Form type: reduction Form name: Reduce

```
Reduction Level
 Polarity
  Exclusive-OR
   Display
-------------------------------------------------------------------------------
e + n r  AUX1.ap = !S12
e + n r  AUX1.clk = AUX7
e - n r  !AUX1.d = !JE10
e + n r  AUX10 = JD17 & PA24
e + n r  AUX11 =
           !S45 & !AUX2 & !S20 & S22 & !JD17 & S21
         # !S45 & !PA6 & !AUX2 & !S20 & S22 & S21
         # S36 & !S45 & !AUX2 & !S20 & S22 & !JD17
         # S36 & !S45 & !PA6 & !AUX2 & !S20 & S22
         # !PA26 & !S45 & S21
         # S36 & !PA26 & !S45
e + n r  AUX12 = S21 & !JF3
e + n r  AUX2.ap = !S12
e + n r  AUX2.clk = AUX7
e - n r  !AUX2.d = !JE12
e + n r  AUX7 = S24 # S45
e + n r  AUX8 = JE7 & S25 # !S34 # !S26
e + n r  AUX9 = PA18 & JE8
e + n r  JE10 = S6 & S5
e + n r  JE11 = S1 & S2 # JE3
e + n r  JE12.ap = !S12
e + n r  JE12.clk = AUX7
e - n r  !JE12.d = !JE15
e + n r  JE14 = !S36 # !S4
e + n r  JE15 = S36 & !S4 # !S3
e + n r  JE3 = AUX1 & S14 # S39 & !JE7 # S8 & S33
e + n r  JE4.ap = !AUX9
e + n r  JE4.clk = S24
e - n r  !JE4.d = !JE4 & !AUX8 # JE4 & AUX8
e + n r  JE8.ap = !S12
e + n r  JE8.clk = S24
e - n r  !JE8.d = JE8 & AUX11 # !JE8 & !AUX10
e + n r  JE9 =
           S39 & S9 & S8
         # AUX1 & S7
         # S39 & S11 & !JF4
         # S39 & !S13 & S11
         # S39 & !S10
o + n r  JF7.ap = !S12
e + n r  JF7.clk = AUX7
e - n r  !JF7.d = !JE14
e + n r  PA25.ar = !S12
e + n r  PA25.clk = S24
e - n r  !PA25.d = !JE8.q
e + n r  PB18.ap = !S12
e + n r  PB18.clk = AUX7
e - n r  !PB18.d = !S33
e + n r  PB18.oe = AUX12
```

```
e + n r  PB21.ap = !S12
e + n r  PB21.clk = AUX7
e - n r  !PB21.d = !JE12
e + n r  PB21.oe = AUX12
e + n r  PB22.ap = !S12
e + n r  PB22.clk = AUX7
e - n r  !PB22.d = !S34
e + n r  PB22.oe = AUX12
e + n r  PB24.ap = !S12
e + n r  PB24.clk = AUX7
e - n r  !PB24.d = !AUX2
e + n r  PB24.oe = AUX12
e + n r  PB26.ap = !S12
e + n r  PB26.clk = AUX7
e - n r  !PB26.d = !AUX1
e + n r  PB26.oe = AUX12
e + n r  PB29 = S39
e + n r  PB29.oe = AUX12
e + n r  S29 = !JE16
e + n r  S31.ap = !S12
e + n r  S31.clk = AUX7
e - n r  !S31.d = JE12
e + n r  S33.ap = !S12
e + n r  S33.clk = AUX7
e - n r  !S33.d = !JE9
e + n r  S34.ap = !S12
e + n r  S34.clk = AUX7
e - n r  !S34.d = !JE11
e + n r  S16 =
             S13 & JF4 & !S15 & S39
           # S19 & S11 & S33
           # S39 & S17 & JE7
           # S33 & S18
           # S19 & !S16
           # AUX2
           # S34
e + n r  S38.ap = !S12
e + n r  S38.clk = AUX7
e - n r  !S38.d = AUX1
e + n r  S39.ap = !S12
e + n r  S39.clk = AUX7
e - n r  !S39.d = JF7
e + n r  S40 = S16 & S23
e + n r  S41 = S40 & !S24 & !S45
e + n r  S42 = !S45 & !S24 & PA5
e + n r  S43 = !AUX7
e + n r  S44 = !AUX7
e + n r  S45.ar = !AUX9
e + n r  S45.clk = S24
e - n r  !S45.d = JE4 & !AUX8 # !JE4 & AUX8
```

Group Definitions

Form type: pld-map Form name: sc3pa1

Partition: SC3PA1
Target Device Type: E0900 Output file format: jedec
Output File Name: sc3pa1
File Title Line:
Checksum Format: full Fast Flag: no
Simulation Form(s):
PLD-Sim? _ Trace: o X-level: 1 Z-level: h Breakpoint:
Display Pins:

Special Fuse Number: Value:

Pin Name	Used As	Prod Terms	Pin Num	---Pin Attributes-- Act Lev	Reg Byp	Reg Typ	Feed Back
AUX1	Input		2	high	yes	com	none
AUX2	Input		30	high	yes	com	none
JE10	Output	1	5	high	yes	com	none
JE11	Output	2	6	high	yes	com	none
JE14	Bidir	2	7	high	yes	com	none
JE15	Output	2	8	high	yes	com	none
JE3	Bidir	3	9	high	yes	com	none
JE7	Input		3	high	yes	com	none
JE9	Output	5	10	high	yes	com	none
JP4	Input		4	high	yes	com	none
S1	Input		11	high	yes	com	none
S10	Input		12	high	yes	com	none
S11	Input		13	high	yes	com	none
S13	Input		14	high	yes	com	none
S14	Input		15	high	yes	com	none
S15	Input		37	high	yes	com	none
S16	Input		38	high	yes	com	none
S17	Input		39	high	yes	com	none
S18	Input		31	high	yes	com	none
S19	Input		32	high	yes	com	none
S2	Input		16	high	yes	com	none
S29	Output	1	25	high	yes	com	none
S3	Input		17	high	yes	com	none
S33	Input		18	high	yes	com	none
S34	Input		33	high	yes	com	none
S36	Bidir	7	19	high	yes	com	none
S39	Input		22	high	yes	com	none
S4	Input		23	high	yes	com	none
S5	Input		24	high	yes	com	none
S6	Input		26	high	yes	com	none
S7	Input		27	high	yes	com	none
S8	Input		28	high	yes	com	none
S9	Input		29	high	yes	com	none

Form type: pld-map Form name: sc3pa2

Partition: SC3PA2
Target Device Type: E0900 Output file format: jedec
Output File Name: sc3pa2
File Title Line:
Checksum Format: full Fast Flag: no
Simulation Form(s):
PLD-Sim? _ Trace: 0 X-level: 1 Z-level: h Breakpoints:
Display Pins:

Special Fuse Number: Value:

Pin Name	Used As	Prod Terms	Pin Num	Act Lev	Reg Byp	Reg Typ	Feed Back
AUX10	Output	1	5	high	yes	com	none
AUX11	Output	6	6	high	yes	com	none
AUX12	Bidir	1	33	high	yes	com	none
AUX2	Input		2	high	yes	com	none
AUX7	Bidir	2	7	high	yes	com	none
AUX8	Output	3	8	high	yes	com	none
AUX9	Output	1	9	high	yes	com	none
JD17	Input		3	high	yes	com	none
JE7	Input		4	high	yes	com	none
JE8	Input		10	high	yes	com	none
JF3	Input		34	high	yes	com	none
PA18	Input		11	high	yes	com	none
PA24	Input		12	high	yes	com	none
PA26	Input		13	high	yes	com	none
PA5	Input		14	high	yes	com	none
PA6	Input		15	high	yes	com	none
PB29	Output	1	35	high	yes	com	none
S20	Input		16	high	yes	com	none
S21	Input		17	high	yes	com	none
S22	Input		18	high	yes	com	none
S23	Input		19	high	yes	com	none
S24	Input		22	high	yes	com	none
S25	Input		23	high	yes	com	none
S26	Input		24	high	yes	com	none
S34	Input		25	high	yes	com	none
S36	Input		26	high	yes	com	none
S39	Input		36	high	yes	com	none
S40	Bidir	1	27	high	yes	com	none
S41	Output	1	28	high	yes	com	none
S42	Output	1	29	high	yes	com	none
S43	Output	1	30	high	yes	com	none
S44	Output	1	31	high	yes	com	none
S45	Input		32	high	yes	com	none

Form type: pld-map Form name: sc3pa3

Partition: SC3PA3
Target Device Type: P331 Output file format: jedec
Output File Name: sc3pa3
File Title Line:
Checksum Format: full Fast Flag: no
Simulation Form(s):
PLD-Sim? _ Trace: 0 X-level: 1 Z-level: h Breakpoints:
Display Pins:

Special Fuse Number: Value:

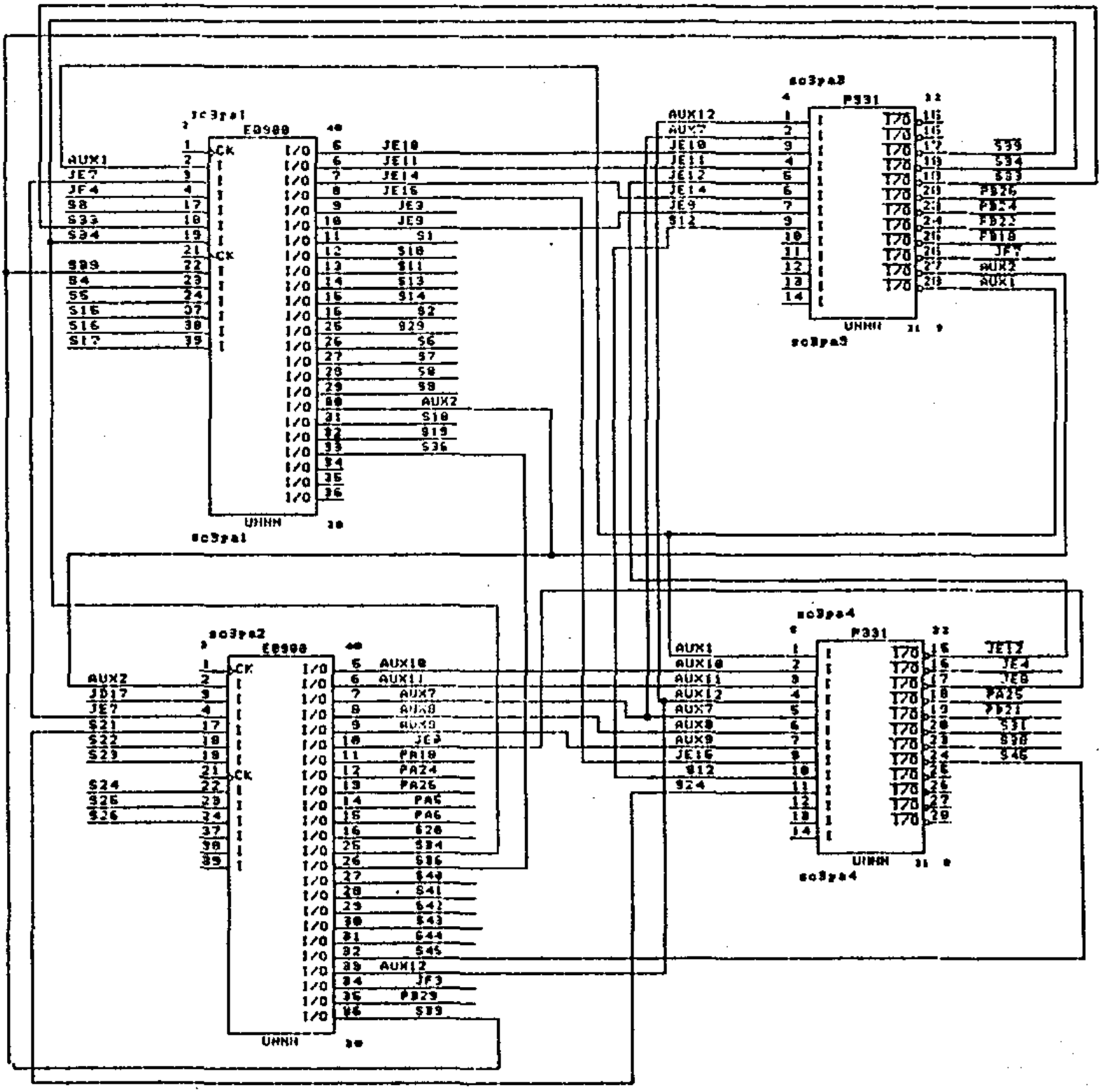
Pin Name	Used As	Prod Terms	Pin Num	---Pin Attributes-- Act Lev	Reg Byp	Reg Typ	Feed Back
AUX1	Bidir	1	28	high	no	d	reg
AUX12	Input		1	high	yes	com	none
AUX2	Bidir	1	27	high	no	d	reg
AUX7	Input		2	high	yes	com	none
JE10	Input		3	high	yes	com	none
JE11	Input		4	high	yes	com	none
JE12	Input		5	high	yes	com	none
JE14	Input		6	high	yes	com	none
JE9	Input		7	high	yes	com	none
JF7	Bidir	1	26	high	no	d	reg
PB18	Output	1	25	high	no	d	reg
PB22	Output	1	24	high	no	d	reg
PB24	Output	1	23	high	no	d	reg
PB26	Output	1	20	high	no	d	reg
S12	Input		9	high	yes	com	none
S33	Bidir	1	19	high	no	d	reg
S34	Bidir	1	18	high	no	d	reg
S39	Output	1	17	high	no	d	reg

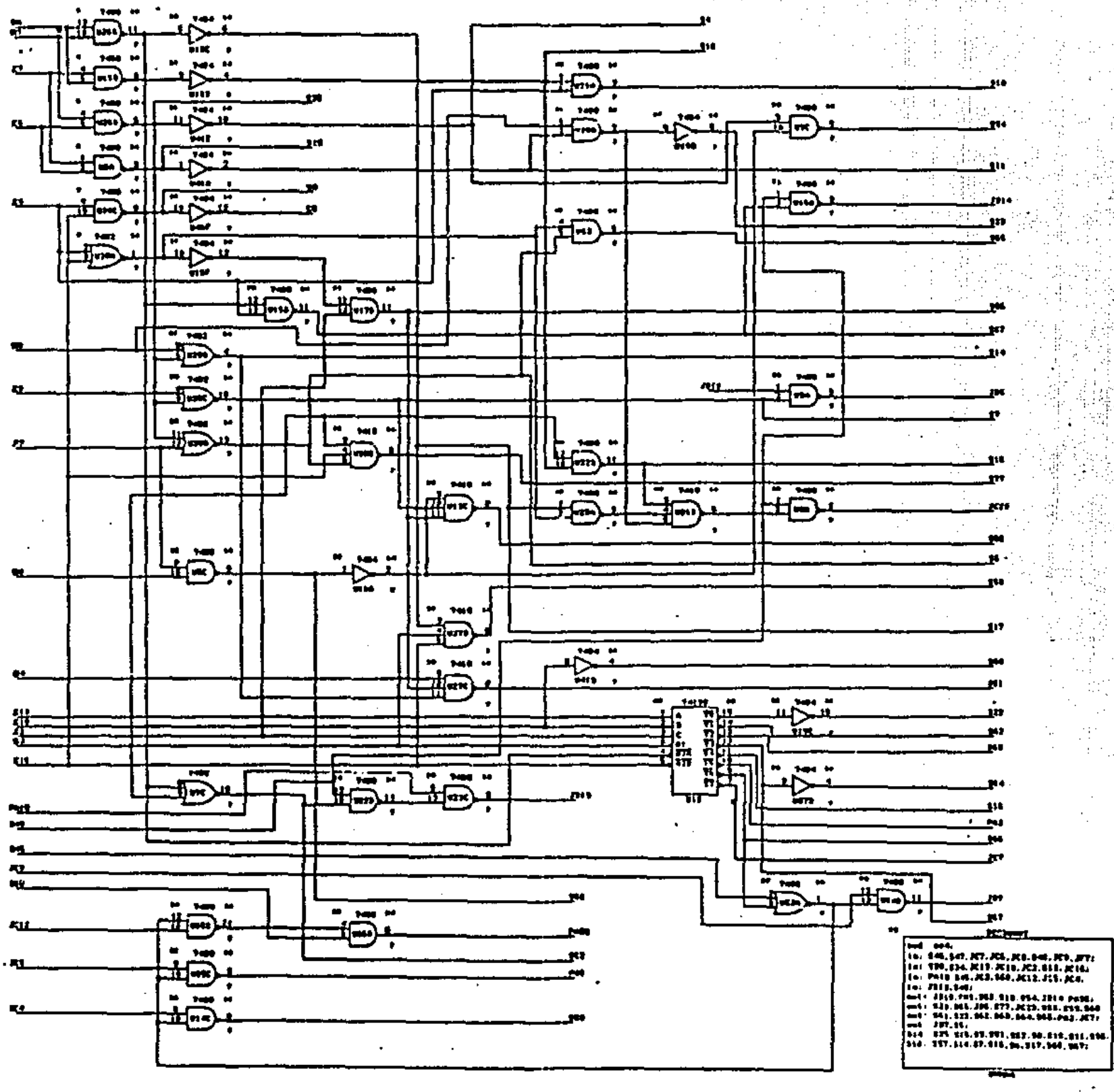
Form type: pld-map Form name: sc3pa4

Partition: SC3PA4
Target Device Type: PJ31 Output file format: jedec
Output File Name: sc3pa4
File Title Line:
Checksum Format: full Fast Flag: no
Simulation Form(s):
PLD-Sim? _ Trace: 0 X-level: 1 Z-level: h Breakpoints:
Display Pins:

Special Fuse Number: Value:

Pin Name	Used As	Prod Terms	Pin Num	---Pin Attributes-- Act Lev	Reg Byp	Reg Typ	Feed Back
AUX1	Input		1	high	yes	com	none
AUX10	Input		2	high	yes	com	none
AUX11	Input		3	high	yes	com	none
AUX12	Input		4	high	yes	com	none
AUX7	Input		5	high	yes	com	none
AUX8	Input		6	high	yes	com	none
AUX9	Input		7	high	yes	com	none
JE12	Bidir	1	15	high	no	d	reg
JE15	Input		9	high	yes	com	none
JE4	Bidir	2	16	high	no	d	reg
JE8	Bidir	2	17	high	no	d	reg
PA25	Output	1	18	high	no	d	reg
PB21	Output	1	19	high	no	d	reg
S12	Input		10	high	yes	com	none
S24	Input		11	high	yes	com	none
S31	Output	1	20	high	no	d	reg
S36	Output	1	23	high	no	d	reg
S45	Output	2	24	high	no	d	reg


sc3pa1
E0900
UHHH
sc3pa2
E0900
UHHH
sc3pa3
P331
UHHH
sc3pa4
P331
UHHH



Form type: reduction Form name: Reduce

Reduction Level
Polarity
Exclusive-OR
Display

```
e + n r  JC23 =
           S16 & JC16 & !S11
         # S16 & JC8 & !S11
         # !S17 & S16 & !S11
         # !S48 & S16 & JC16
         # !S48 & S16 & JC8
         # !S48 & !S17 & S16
         # !S49
e + n r  JD14 = !S57 # !S49
e + n r  JD19 = S49 & S52 # !PA10
e + n r  JD5 = !S7 # !JD13
e + n r  JD7 = S46 # S45 # !JC3
e + n r  JE7 = JC16 # !S47 # !S46 # !S13 # !JC2 # !JC10 # !JC13
e + n r  PA1 = S66 # S45 # !JC5
e + n r  PA2 = JC16 # !S47 # !S46 # !S13 # !JC2 # JC10 # !JC13
e + n r  PA28 = S50 & !JC12 # S50 & S66 # S50 & S45
e + n r  S10 = !JC8 # !S18
e + n r  S11 = !S15
e + n r  S14 = !S48 & !S25
e + n r  S15 = !JC6 # !JC7
e + n r  S16 = !S18 # !S13
e + n r  S17 = S46 & S47
e + n r  S18 = S46 & JC7
e + n r  S22 = !JC13 & !JC10 & !JC2 & S13 & S46 & S47 & !JC16
e + n r  S23 = S48 & S11
e + n r  S25 = !S47 # !JC6
e + n r  S5 = !S9
e + n r  S51 = !S38 # !JF7
e + n r  S52 = S46 & S47 & !S13
e + n r  S53 = S66 # S45 # !JC4
e + n r  S54 = S51 # !S8
e + n r  S55 = JC16 # JC8 # !S6
e + n r  S56 = JC2 & JC16 # JC8 & JC2
e + n r  S57 = S46 & S47 # !JC8
e + n r  S58 = !S56 # !S7 # S51
e + n r  S59 = !S17 # !S13 # !JC16
e + n r  S6 = !S25 & !JF7
e + n r  S60 = !JC10
e + n r  S61 = !S14 # !S56 # !S34
e + n r  S62 = JC16 # !S47 # !S46 # !S13 # JC2 # JC10 # !JC13
e + n r  S63 = JC16 # !S47 # !S46 # !S13 # JC2 # !JC10 # JC13
e + n r  S64 = !S67
e + n r  S65 = JC16 # !S47 # !S46 # !S13 # !JC2 # JC10 # JC13
e + n r  S66 = JC16 # !S47 # !S46 # !S13 # !JC2 # !JC10 # JC13
e + n r  S67 = JC16 # !S47 # !S46 # !S13 # JC2 # !JC10 # !JC13
e + n r  S7 = !JC9 & !S25
e + n r  S77 = !S13 # !JC16 # !S6
e + n r  S8 = !S25
```

e + n r S9 = !JC16 # !JC8

Group Definitions

Form type: pld-map Form name: sc4pa1

Partition: SC4PA1
Target Device Type: E0900 Output file format: jedec
Output File Name: sc4pa1
File Title Line:
Checksum Format: full Fast Flag: no
Simulation Form(s):
PLD-Sim? _ Trace: 0 X-level: 1 Z-level: h Breakpoints:
Display Pins:

Special Fuse Number: Value:

Pin Name	Used As	Prod Terms	Pin Num	---Pin Attributes-- Act Lev	Reg Byp	Reg Typ	Feed Back
JC16	Input		2	high	yes	com	none
JC2	Input		3	high	yes	com	none
JC23	Output	7	5	high	yes	com	none
JC6	Input		4	high	yes	com	none
JC7	Input		17	high	yes	com	none
JC8	Input		18	high	yes	com	none
JC9	Input		19	high	yes	com	none
JD13	Input		22	high	yes	com	none
JD14	Output	2	6	high	yes	com	none
JD5	Output	2	7	high	yes	com	none
S10	Output	2	8	high	yes	com	none
S11	Bidir	1	9	high	yes	com	none
S13	Input		23	high	yes	com	none
S14	Output	1	10	high	yes	com	none
S15	Bidir	2	11	high	yes	com	none
S16	Bidir	2	12	high	yes	com	none
S17	Input		24	high	yes	com	none
S18	Bidir	1	13	high	yes	com	none
S23	Output	1	14	high	yes	com	none
S25	Bidir	2	15	high	yes	com	none
S46	Input		37	high	yes	com	none
S47	Input		38	high	yes	com	none
S48	Input		39	high	yes	com	none
S49	Input		36	high	yes	com	none
S5	Output	1	16	high	yes	com	none
S51	Input		35	high	yes	com	none
S54	Output	2	25	high	yes	com	none
S55	Output	3	26	high	yes	com	none
S56	Bidir	2	27	high	yes	com	none
S57	Bidir	2	28	high	yes	com	none
S58	Output	3	29	high	yes	com	none
S6	Input		34	high	yes	com	none
S7	Bidir	1	30	high	yes	com	none
S77	Output	3	31	high	yes	com	none
S8	Bidir	1	32	high	yes	com	none
S9	Bidir	2	33	high	yes	com	none

Form type: pld-asp Form name: sc4pa2

Partition: SC4PA2
Target Device Type: E0900 Output file format: jedec
Output File Name: sc4pa2
File Title Line:
Checksum Format: full Fast Flag: no
Simulation Form(s):
PLD-Sim? _ Trace: 0 X-level: 1 Z-level: h Breakpoints:
Display Pins:

Special Fuse Number: Value:

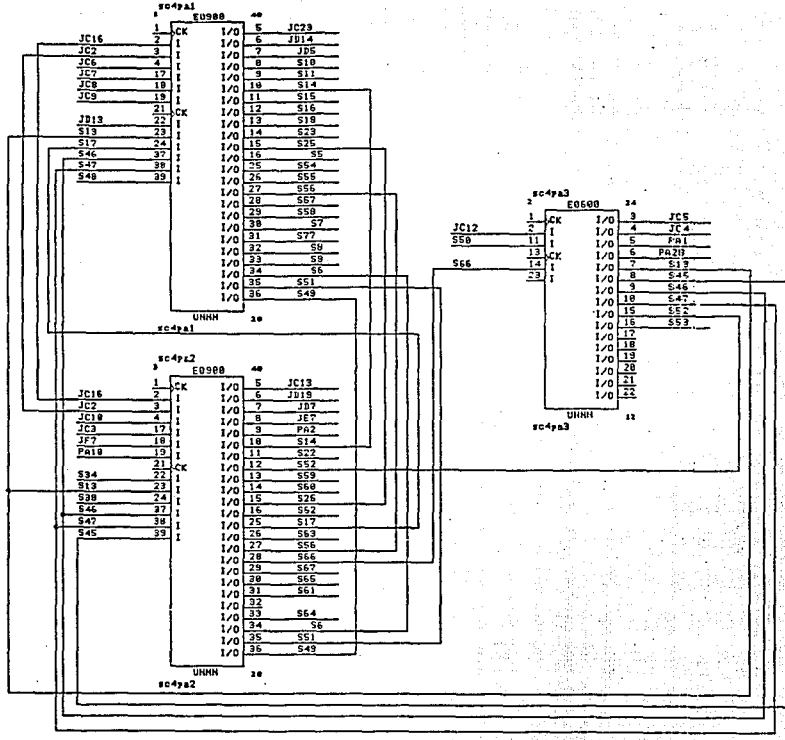
Pin Name	Used As	Prod Terms	Pin Num	---Pin Attributes-- Act Lev	Reg Byp	Reg Typ	Feed Back
JC10	Input		4	high	yes	com	none
JC13	Input		5	high	yes	com	none
JC16	Input		2	high	yes	com	none
JC2	Input		3	high	yes	com	none
JC3	Input		17	high	yes	com	none
JD19	Output	2	6	high	yes	com	none
JD7	Output	3	7	high	yes	com	none
JE7	Output	7	8	high	yes	com	none
JF7	Input		18	high	yes	com	none
PA10	Input		19	high	yes	com	none
PA2	Output	7	9	high	yes	com	none
S13	Input		23	high	yes	com	none
S14	Input		10	high	yes	com	none
S17	Bidir	1	25	high	yes	com	none
S22	Output	1	11	high	yes	com	none
S25	Input		15	high	yes	com	none
S34	Input		22	high	yes	com	none
S38	Input		24	high	yes	com	none
S45	Input		39	high	yes	com	none
S46	Input		37	high	yes	com	none
S47	Input		38	high	yes	com	none
S49	Input		36	high	yes	com	none
S51	Output	2	35	high	yes	com	none
S52	Input		12	high	yes	com	none
S56	Input		27	high	yes	com	none
S59	Output	3	13	high	yes	com	none
S6	Output	1	14	high	yes	com	none
S60	Output	1	14	high	yes	com	none
S61	Output	3	31	high	yes	com	none
S62	Output	7	16	high	yes	com	none
S63	Output	7	26	high	yes	com	none
S64	Output	1	33	high	yes	com	none
S65	Output	7	30	high	yes	com	none
S66	Bidir	7	28	high	yes	com	none
S67	Bidir	7	29	high	yes	com	none

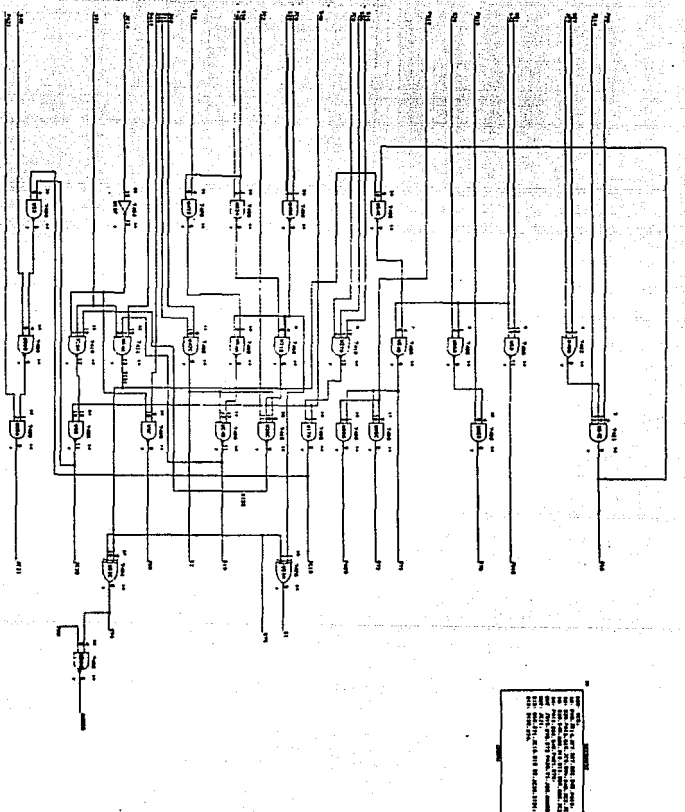
Form type: pld-map Form name: sc4pa3

Partition: SC4PA3
Target Device Type: E0600 Output file format: jedec
Output File Name: sc4pa3
File Title Line:
Checksum Format: full Fast Flag: no
Simulation Form(s):
PLD-Sim? _ Trace: 0 X-Level: 1 Z-Level: h Breakpoints:
Display Pins:

Special Fuse Number: Value:

Pin Name	Used As	Prod Terms	Pin Num	---Pin Attributes-- Act Lev	Reg Byp	Reg Typ	Feed Back
JC12	Input		2	high	yes	com	none
JC4	Input		4	high	yes	com	none
JC5	Input		3	high	yes	com	none
PA1	Output	3	5	high	yes	com	none
PA28	Output	3	6	high	yes	com	none
S13	Input		7	high	yes	com	none
S45	Input		8	high	yes	com	none
S46	Input		9	high	yes	com	none
S47	Input		10	high	yes	com	none
S50	Input		11	high	yes	com	none
S52	Output	1	15	high	yes	com	none
S53	Output	3	16	high	yes	com	none
S66	Input		14	high	yes	com	none


sc4pa1
EQ900
JC16
JC2
JC6
JC7
JC8
JC9
JD13
S13
S17
S46
S47
S48
JC23
JD14
JD5
S10
S11
S14
S15
S16
S18
S23
S25
S5
S54
S55
S56
S57
S50
S7
S77
S8
S9
S6
S51
S49
UNKN
sc4pa2
EQ900
JC16
JC2
JC10
JC3
JF7
PA18
S34
S13
S38
S46
S47
S45
JC13
JD18
JD7
JE7
PA2
S14
S22
S52
S59
S60
S26
S52
S17
S63
S56
S66
S67
S65
S61
S64
S6
S51
S49
UNKN
sc4pa3
EQ600
JC12
S50
S66
JC5
JC4
PA1
PA28
S19
S45
S46
S47
S52
S53
UNKN



Form type: reduction Form name: Reduce

Reduction Level
Polarity
Exclusive-OR
Display

```
o + n r  AUX50 = !S74 # !PA9
o + n r  BID1 = PA11 & S19 & S31
o + n r  BID2 =
           !S33 & !S48
         # !JC8 & !S48
         # !S39 & !S33
         # !S39 & !JC8
         # !JC2
         # !S2
o + n r  JC19 = S49 & !JC9 # S49 & !S36 # S49 & !S15
o + n r  JC20 = BID2 & S31 & !JC24 # !S49
o + n r  JC21 = PA21 & JC19 & JC20 # !S43 & PA21
o + n r  JD10 = !S42 # !S51
o + n r  JD9 = JC24 # !S55
o + n r  PA20 = S71 & PA13
o + n r  S1 = JC8 & S33 & S73 # !S33 & !S73 # !JC8 & !S73
o + n r  S19 =
           !S33 & !S13
         # !JC8 & !S13
         # !S39 & !S33
         # !S39 & !JC8
         # !S2
o + n r  S2 = S68 & S11
o + n r  S69 = PA9 & JE14 & JF7 # PA9 & S57 & JE14
o + n r  S70 = PA19 & !S29 # PA19 & !S42
o + n r  S71 = S69 & BID1 # !S42
o + n r  S72 = PA13 & S71
o + n r  S74 = BID1 & !S73 # !BID1 & S73
```

Group Definitions

Form type: pld-map Form name: sc5pa1

Partition: SC5PA1
Target Device Type: E0900 Output file format: jedec
Output File Name: sc5pa1
File Title Line:
Checksum Format: full Fast Flag: no
Simulation Form(s):
PLD-Sim? _ Trace: 0 X-level: 1 Z-level: h Breakpoints:
Display Pins:

Special Fuse Number: Value:

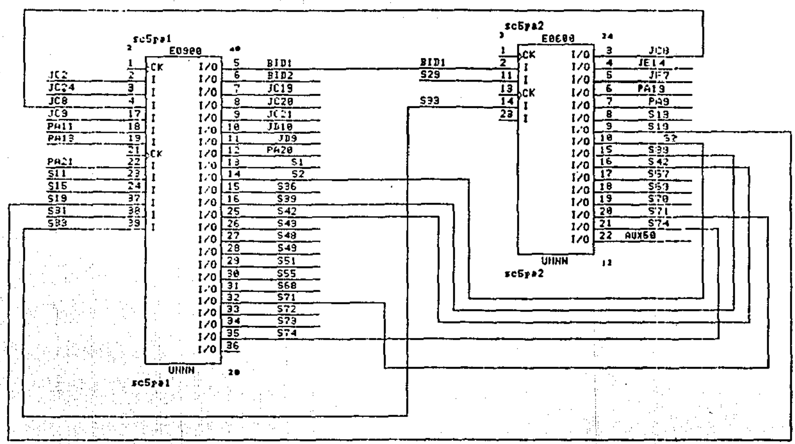
Pin Name	Used As	Prod Terms	Pin Num	Act Lev	Reg Typ	Reg Back	Feed
BID1	Bidir	1	5	high	yes	com	none
BID2	Bidir	6	6	high	yes	com	none
JC19	Bidir	3	7	high	yes	com	none
JC2	Input		2	high	yes	com	none
JC20	Bidir	2	8	high	yes	com	none
JC21	Output	2	9	high	yes	com	none
JC24	Input		3	high	yes	com	none
JC8	Input		4	high	yes	com	none
JC9	Input		17	high	yes	com	none
JD10	Output	2	10	high	yes	com	none
JD9	Output	2	11	high	yes	com	none
PA11	Input		18	high	yes	com	none
PA13	Input		19	high	yes	com	none
PA20	Output	1	12	high	yes	com	none
PA21	Input		22	high	yes	com	none
S1	Output	3	13	high	yes	com	none
S11	Input		23	high	yes	com	none
S15	Input		24	high	yes	com	none
S19	Input		37	high	yes	com	none
S2	Bidir	1	14	high	yes	com	none
S31	Input		38	high	yes	com	none
S33	Input		39	high	yes	com	none
S36	Input		15	high	yes	com	none
S39	Input		16	high	yes	com	none
S42	Input		25	high	yes	com	none
S43	Input		26	high	yes	com	none
S48	Input		27	high	yes	com	none
S49	Input		28	high	yes	com	none
S51	Input		29	high	yes	com	none
S55	Input		30	high	yes	com	none
S68	Input		31	high	yes	com	none
S71	Input		32	high	yes	com	none
S72	Output	1	33	high	yes	com	none
S73	Input		34	high	yes	com	none
S74	Output	2	35	high	yes	com	none

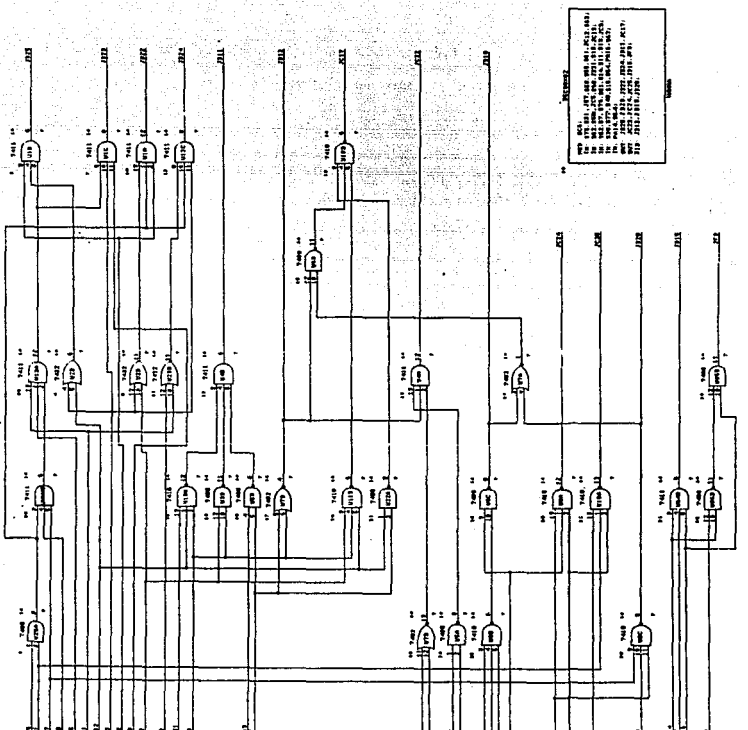
Form type: pld-map Form name: sc5pa2

Partition: SC5PA2
Target Device Type: E0600 Output file format: jedec
Output File Name: sc5pa2
File Title Line:
Checksum Format: full Fast Flag: no
Simulation Form(s):
PLD-Sim? _ Trace: 0 X-level: 1 Z-level: h Breakpoints:
Display Pins:

Special Fuse Number: Value:

Pin Name	Used As	Prod Terms	Pin Num	---Pin Attributes-- Act Lev	Reg Byp	Reg Typ	Feed Back
AUX50	Output	2	22	high	yes	com	none
BID1	Input		2	high	yes	com	none
JC8	Input		3	high	yes	com	none
JE14	Input		4	high	yes	com	none
JF7	Input		5	high	yes	com	none
PA19	Input		6	high	yes	com	none
PA9	Input		7	high	yes	com	none
S13	Input		8	high	yes	com	none
S19	Output	5	9	high	yes	com	none
S2	Input		10	high	yes	com	none
S29	Input		11	high	yes	com	none
S33	Input		14	high	yes	com	none
S39	Input		15	high	yes	com	none
S42	Input		16	high	yes	com	none
S57	Input		17	high	yes	com	none
S69	Bidir	2	18	high	yes	com	none
S70	Output	2	19	high	yes	com	none
S71	Output	2	20	high	yes	com	none
S74	Input		21	high	yes	com	none


scGpa1
E0900
UNNN
scGpa2
E0E00
UNNN
BID1
BID2
JC19
JC20
JC21
JD10
JD9
PA20
S1
S2
S36
S39
S42
S43
S48
S49
S51
S55
S68
S71
S72
S73
S74
JC2
JC24
JC8
JC9
PA11
PA13
PA21
S11
S15
S19
S31
S83
S29
S33
JC3
JE14
JF7
PA19
PA9
S13
S57
S53
S70
AUX60
CK
I
I/O



Form type: reduction Form name: Reduce

Reduction Level
Polarity
Exclusive-OR
Display

```
e + n r  JC17 = JD12 & !JD18 & !JD20 # JC12 & S18 & JC5 # JC12 & S52
e + n r  JC22 = JD12 & !S7 & !S76 & !S51 # !S14 & JD12 & !S7 & !S76
e + n r  JC24 = !S49 # !S38 # !S77
e + n r  JC25 = !S49 # !S31 # !S15
e + n r  JD11 =
           !JC5 & !JC13 & !JC12
         # !JC5 & !S52 & !JC12
         # !JD21 & !JC5 & !JC13
         # !JD21 & !JC5 & !S52
         # !JC13 & !S18
         # !S52 & !S18
e + n r  JD12 = !S52 & !S18
e + n r  JD15 = PA14 & S67 & PA16
e + n r  JD18 = S11 & S13 & JC9 # !S38
e + n r  JD20 = !S77 # !S54 # !JE7
e + n r  JD22 = S75 & S31 & S62 & JC5 # S63 & S75 & S31 & S62
e + n r  JD23 = S61 & S55 & S75 & S31 & S59 & JE7 & S63 & S58
e + n r  JD24 = S75 & S31 & S63 & S61 # S60 & S75 & S31 & S63
e + n r  JD25 =
           S62 & S61 & S55 & S75 & S31 & S59 & JE7 & S63
         # S62 & JC12 & S61 & S55 & S75 & S31 & S59 & JE7
e + n r  JF3 = PA16 & !PA14 # !S64 & PA16
```

Group Definitions

Form type: pld-map Form name: sc6pal

Partition: SC6PA1
Target Device Type: E0900 Output file format: jedec
Output File Name: sys6pal
File Title Line:
Checksum Format: full Fast Flag: no
Simulation Form(s):
PLD-Sim? _ Trace: 0 X-level: 1 Z-level: h Breakpoints:
Display Pins:

Special Fuse Number: Value:

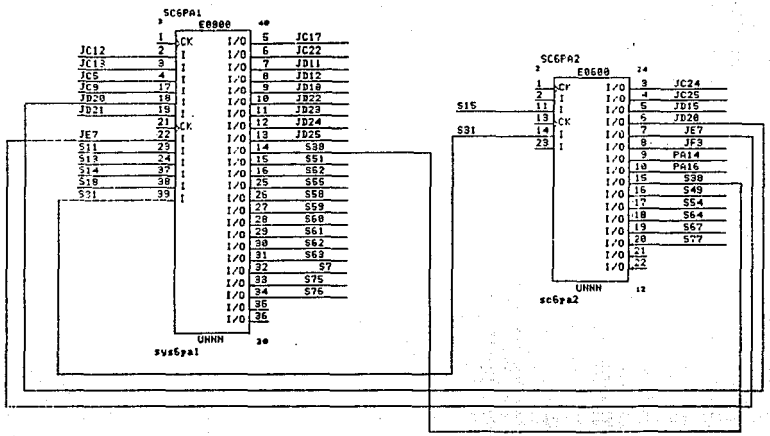
Pin Name	Used As	Prod Terms	Pin Num	Act Lev	Reg Typ	Reg Feed Back
JC12	Input		2	high	yes	com none
JC13	Input		3	high	yes	com none
JC17	Output	3	5	high	yes	com none
JC22	Output	2	6	high	yes	com none
JC5	Input		4	high	yes	com none
JC9	Input		17	high	yes	com none
JD11	Output	6	7	high	yes	com none
JD12	Bidir	1	8	high	yes	com none
JD18	Bidir	2	9	high	yes	com none
JD20	Input		18	high	yes	com none
JD21	Input		19	high	yes	com none
JD22	Output	2	10	high	yes	com none
JD23	Output	1	11	high	yes	com none
JD24	Output	2	12	high	yes	com none
JD25	Output	2	13	high	yes	com none
JE7	Input		22	high	yes	com none
S11	Input		23	high	yes	com none
S13	Input		24	high	yes	com none
S14	Input		37	high	yes	com none
S18	Input		38	high	yes	com none
S31	Input		39	high	yes	com none
S38	Input		14	high	yes	com none
S51	Input		15	high	yes	com none
S52	Input		16	high	yes	com none
S55	Input		25	high	yes	com none
S58	Input		26	high	yes	com none
S59	Input		27	high	yes	com none
S60	Input		28	high	yes	com none
S61	Input		29	high	yes	com none
S62	Input		30	high	yes	com none
S63	Input		31	high	yes	com none
S7	Input		32	high	yes	com none
S75	Input		33	high	yes	com none
S76	Input		34	high	yes	com none

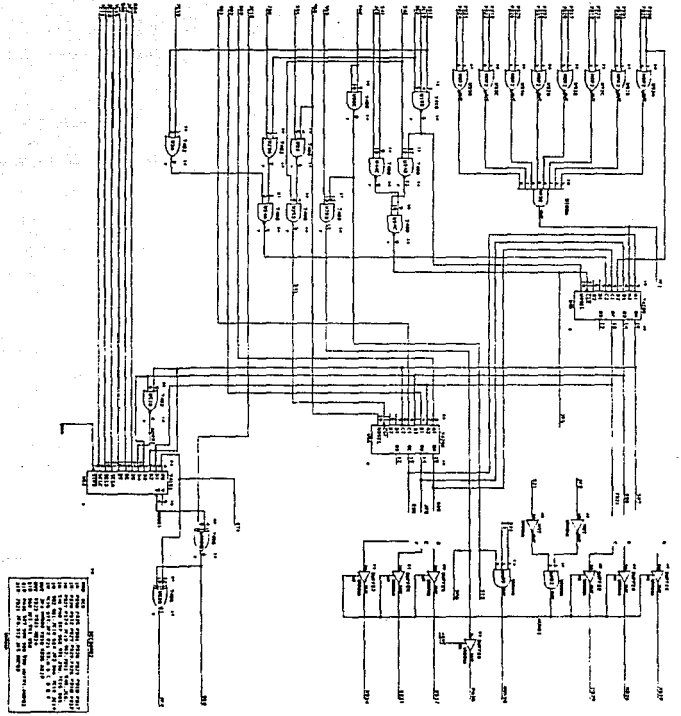
Form type: pld-map Form name: sc6pa2

Partition: SC6PA2
Target Device Type: E0600 Output file format: jedec
Output File Name: sc6pa2
File Title Line: _____
Checksum Format: full Fast Flag: no
Simulation Form(s):
PLD-Sim? _ Trace: 0 X-level: 1 Z-level: h Breakpoints: _____
Display Pins: _____

Special Fuse Number: Value:

Pin Name	Used As	Prod Terms	Pin Num	Act Lev	Reg Byp	Reg Typ	Feed Back
JC24	Output	3	3	high	yes	com	none
JC25	Output	3	4	high	yes	com	none
JD15	Output	1	5	high	yes	com	none
JD20	Output	3	6	high	yes	com	none
JE7	Input		7	high	yes	com	none
JF3	Output	2	8	high	yes	com	none
PA14	Input		9	high	yes	com	none
PA16	Input		10	high	yes	com	none
S15	Input		11	high	yes	com	none
S31	Input		14	high	yes	com	none
S38	Input		15	high	yes	com	none
S49	Input		16	high	yes	com	none
S54	Input		17	high	yes	com	none
S64	Input		18	high	yes	com	none
S67	Input		19	high	yes	com	none
S77	Input		20	high	yes	com	none


SC6PA1
E0800
UNNN
sys6pa1
SC6PA2
E0680
UNNN
sc6pa2



Form type: reduction Form name: Reduce

```
Reduction Level
 Polarity
  Exclusive-OR
   Display
-------------------------------------------------------------------------------
e + n r  AUX80 = S31 & S3 & S12
e + n r  AUX81 =
           !S73 & !JC11 & !JC14 & !JC15
         # !AUX82 & JC11 & !JC14 & !JC15
         # !S87 & !JC11 & JC14 & !JC15
         # !JF2 & JC11 & JC14 & !JC15
         # !JD21 & !JC11 & !JC14 & JC15
         # !S84 & JC11 & !JC14 & JC15
         # !S88 & !JC11 & JC14 & JC15
         # !S86 & JC11 & JC14 & JC15
e + n r  AUX82 = !S87 & !S88
e + n r  AUX83 = !JF3 & !S21
e + n r  JD21.clk = !JF6
e + n r  JD21.d =
           S62 & JD12 & JC19 & S89 # !JC18 & !JD12 # !S62 & !JD8
e + n r  JF1 =
           !PB30
         & !PB25
         & !PB31
         & !PB28
         & !PB23
         & !PB19
         & !PB17
         & !PB20
         & !PB32
         & !PB27
         & !PB29
         & !PB26
         & !PB18
         & !PB22
         & !PB21
         & !PB24
e + n r  JF4 = S73 & !S60 # !S73 & S68
e + n r  JF5.clk = !SX1
e + n r  JF5.d = !S65 & S82 # S88 & S65
e + n r  JF6 = S41 & JC5 # S43 & !JD12 # S43 & !JC19 # !S62 & S43
e + n r  PA36 = S50
e + n r  PA36.oe = 1
e + n r  S12 = PA8
e + n r  S50 = S12 & S53
e + n r  S68 = AUX81 & !JC16 # !AUX81 & JC16
e + n r  S87.clk = !JF6
e + n r  S87.d =
           S62 & JC19 & JD12 & S90
         # JF1 & !JD12
         # JF1 & !JC19
         # JF1 & !S62
e + n r  S88.clk = !JF6
```

```
e + n r  S88.d =
           S62 & JC19 & JD12 & JF5
         # PB30 & !JD12
         # PB30 & !JC19
         # PB30 & !S62
e + n r  S89.clk = !SX1
e + n r  S89.d = !S65 & S83 # JD21 & S65
e + n r  S90.clk = !SX1
e + n r  S90.d = !S65 & S81 # S87 & S65
e + n r  SX1 = S43 & !S65 # S43 & !S31
e + n r  XB18 = A
e + n r  XB18.oe = AUX83
e + n r  XB21 = E
e + n r  XB21.oe = AUX83
e + n r  XB22 = D
e + n r  XB22.oe = AUX83
e + n r  XB24 = F
e + n r  XB24.oe = AUX83
e + n r  XB26 = B
e + n r  XB26.oe = AUX83
e + n r  XB29 = C
e + n r  XB29.oe = AUX83
```

Group Definitions

Form type: pld-map Form name: sc8pal

Partition: SC8PA1
Target Device Type: EG900 Output file format: jedec
Output File Name: sc8pal
File Title Line:
Checksum Format: full Fast Flag: no
Simulation Form(s):
PLD-Sim? _ Trace: 0 X-level: 1 Z-level: h Breakpoints:
Display Pins:

Special Fuse Number: Value:

Pin Name	Used As	Prod Terms	Pin Num	Act Lev	Reg Byp	Reg Typ	Feed Back
				---Pin Attributes--			
A	Input		2	high	yes	com	none
AUX80	Output	1	36	high	yes	d	reg
AUX82	Output	1	35	high	yes	d	reg
AUX83	Input		3	high	yes	com	none
B	Input		4	high	yes	com	none
C	Input		5	high	yes	com	none
JC18	Input		6	high	yes	com	none
JC19	Input		7	high	yes	com	none
JC5	Input		8	high	yes	com	none
JD12	Input		9	high	yes	com	none
JD21	Output	3	34	high	no	d	reg
JD8	Input		10	high	yes	com	none
JF1	Input		11	high	yes	com	none
JF5	Input		12	high	no	com	none
JF6	Bidir	4	33	high	yes	com	none
PA36	Output	1	32	high	yes	com	none
PA8	Input		13	high	yes	com	none
PB30	Input		14	high	yes	com	none
S12	Bidir	1	31	high	yes	com	none
S3	Input		15	high	yes	com	none
S31	Input		17	high	yes	com	none
S41	Input		18	high	yes	com	none
S43	Input		19	high	yes	com	none
S50	Bidir	1	30	high	yes	d	reg
S53	Input		22	high	yes	com	none
S62	Input		23	high	yes	com	none
S65	Input		24	high	yes	com	none
S87	Bidir	4	29	high	no	d	reg
S88	Bidir	4	28	high	no	d	reg
S89	Input		37	high	yes	com	none
S90	Input		38	high	yes	com	none
SX1	Output	2	27	high	yes	d	reg
XB18	Output	1	26	high	yes	d	reg
XB26	Output	1	25	high	yes	d	reg
XB29	Output	1	16	high	yes	d	reg

Form type: pld-map Form name: sc8pa2

Partition: SC8PA2
Target Device Type: E0600 Output file format: jedec
Output File Name: sc8pa2
File Title Line:
Checksum Format: full Fast Flag: no
Simulation Form(s):
PLD-Sim? _ Trace: 0 X-Level: 1 Z-Level: h BreakPoints:
Display Pins:

Special Fuse Number: Value:

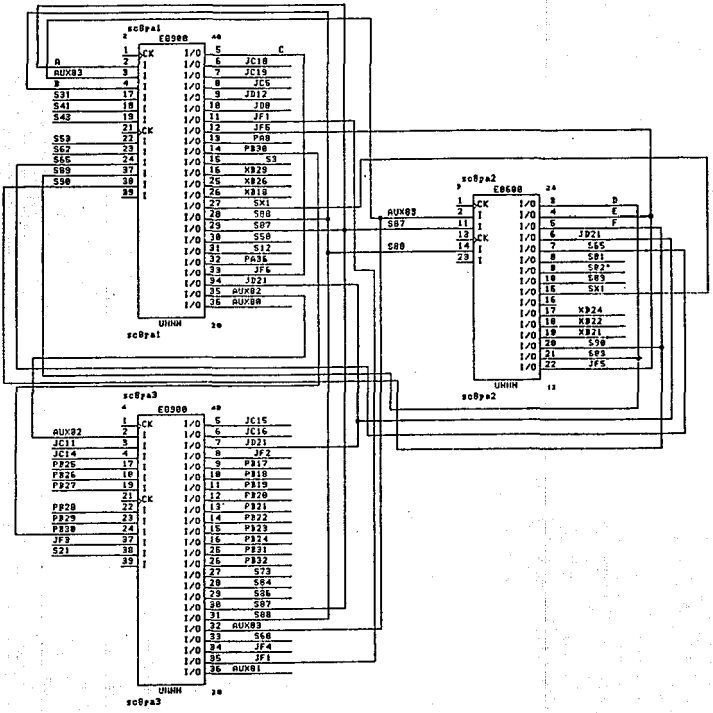
Pin Name	Used As	Prod Terms	Pin Num	---Pin Attributes--- Act Lev	Reg Byp	Reg Typ	Feed Back
AUX83	Input		2	high	yes	com	none
D	Input		3	high	yes	com	none
E	Input		4	high	yes	com	none
F	Input		5	high	yes	com	none
JD21	Input		6	high	no	com	none
JF5	Output	2	22	high	no	d	reg
S65	Input		7	high	yes	com	none
S81	Input		8	high	yes	com	none
S82	Input		9	high	yes	com	none
S83	Input		10	high	yes	com	none
S87	Input		11	high	yes	com	none
S88	Input		14	high	yes	com	none
S89	Output	2	21	high	no	d	reg
S90	Output	2	20	high	no	d	reg
SX1	Input		15	high	yes	com	none
XB21	Output	1	19	high	yes	com	none
XB22	Output	1	18	high	yes	com	none
XB24	Output	1	17	high	yes	com	none

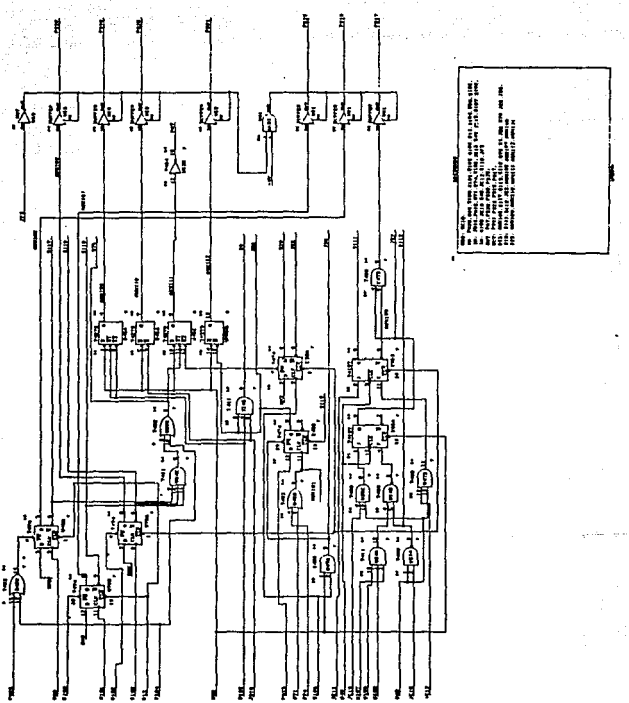
Form type: pld-map Form name: sc8pa3

Partition: SC8PA3
Target Device Type: E0900 Output file format: jedec
Output File Name: sc8pa31
File Title Line:
Checksum Format: full Fast Flag: no
Simulation Form(s):
PLD-Sim? _ Trace: 0 X-level: 1 Z-level: h Breakpoints:
Display Pins:

Special Fuse Number: Value:

Pin Name	Used As	Prod Terms	Pin Num	---Pin Attributes-- Act Lev	Reg Byp	Reg Typ	Feed Back
AUX81	Bidir	8	36	high	yes	com	none
AUX82	Input		2	high	yes	com	none
AUX83	Output	1	32	high	yes	com	none
JC11	Input		3	high	yes	com	none
JC14	Input		4	high	yes	com	none
JC15	Input		5	high	yes	com	none
JC16	Input		6	high	yes	com	none
JD21	Input		7	high	no	com	none
JF1	Output	1	35	high	yes	com	none
JF2	Input		0	high	yes	com	none
JF3	Input		37	high	yes	com	none
JF4	Output	2	34	high	yes	com	none
PB17	Input		9	high	yes	com	none
PB18	Input		10	high	yes	com	none
PB19	Input		11	high	yes	com	none
PB20	Input		12	high	yes	com	none
PB21	Input		13	high	yes	com	none
PB22	Input		14	high	yes	com	none
PB23	Input		15	high	yes	com	none
PB24	Input		16	high	yes	com	none
PB25	Input		17	high	yes	com	none
PB26	Input		18	high	yes	com	none
PB27	Input		19	high	yes	com	none
PB28	Input		22	high	yes	com	none
PB29	Input		23	high	yes	com	none
PB30	Input		24	high	yes	com	none
PB31	Input		25	high	yes	com	none
PB32	Input		26	high	yes	com	none
S21	Input		38	high	yes	com	none
S68	Bidir	2	33	high	yes	com	none
S73	Input		27	high	yes	com	none
S84	Input		28	high	yes	com	none
S86	Input		29	high	yes	com	none
S87	Input		30	high	no	com	none
S88	Input		31	high	no	com	none


sc0pa1
E0908
UHHH
sc0pa1
sc0pa2
E0608
UHHH
sc0pa2
sc0pa3
E0308
UHHH
sc0pa3



Form type: reduction Form name: Reduce

Reduction Level
Polarity
Exclusive-OR
Display

```
e + n r  AUX101 = S74 # S71
e + n r  AUX105.ap = !S12
e + n r  AUX105.ar = !PA29 & !S104
e + n r  AUX105.clk = S99
e - n r  !AUX105.d = 1
e + n r  AUX106.ar = !S12
e + n r  AUX106.clk = !S43
e - n r  !AUX106.d = !S111 & JE12 # S111 & !S40 # S111 & !JC12
e + n r  AUX107.ap = !S12
e + n r  AUX107.ar = !S100
e + n r  AUX107.clk = S101
e - n r  !AUX107.d = 1
e + n r  AUX108.ap = !S12
e + n r  AUX108.ar = !S102
e + n r  AUX108.clk = S103
e - n r  !AUX108.d = 1
e + n r  AUX109 = AUX109 & S50 # !S105
e + n r  AUX110 = AUX110 & S50 # !JD16
e + n r  AUX111 = AUX111 & S50 # !S3 # !S75
e + n r  AUX112 = AUX112 & S50 # !JE6
e + n r  JD6 = S50 & S106
e + n r  JE2.ap = !S50
e + n r  JE2.clk = !S43
e - n r  !JE2.d =
           !JE2 & !S40
         # !JE2 & !JC13
         # JE2 & !S107
         # JE2 & !S109
         # JE2 & !S108
         # JE2 & !S112
e + n r  JE5.ap = !S3
e + n r  JE5.ar = !S12
e + n r  JE5.clk = S106
e - n r  !JE5.d = 0
e + n r  JE6.ap = !S119
e + n r  JE6.ar = !JD6
e + n r  JE6.clk = AUX101
e - n r  !JE6.d = !PA15
e + n r  PA7 = !AUX111
e + n r  PB17 = AUX106 & JE2
e + n r  PB17.oe = !JF3
e + n r  PB19.ap = !S12
e + n r  PB19.ar = !PA29 & !S104
e + n r  PB19.clk = S99
e - n r  !PB19.d = !AUX105
e + n r  PB19.oe = !JF3
e + n r  PB23.ap = !S12
e + n r  PB23.ar = !S100
```

```
0 + n r  PB23.clk = S101
0 - n r  !PB23.d = !AUX107
0 + n r  PB23.oe = !JF3
0 + n r  PB25 = AUX110
0 + n r  PB25.oe = !JF3
0 + n r  PB28.ap = !S12
0 + n r  PB28.ar = !S102
0 + n r  PB28.clk = S103
0 - n r  !PB28.d = !AUX108
0 + n r  PB28.oe = !JF3
0 + n r  PB30 = AUX109
0 + n r  PB30.oe = !JF3
0 + n r  PB31 = AUX112
0 + n r  PB31.oe = !JF3
0 + n r  S110.ap = !S100
0 + n r  S110.ar = !S12
0 + n r  S110.clk = S101
0 - n r  !S110.d = 0
0 + n r  S111.ar = !S12
0 + n r  S111.clk = !S43
0 - n r  !S111.d = S111 & JC12 & S40 # !S111 & !JE12
0 + n r  S112 = !JC10 # !S40
0 + n r  S117.ap = !PA29 & !S104
0 + n r  S117.ar = !S12
0 + n r  S117.clk = S99
0 - n r  !S117.d = 0
0 + n r  S118.ap = !S102
0 + n r  S118.ar = !S12
0 + n r  S118.clk = S103
0 - n r  !S118.d = 0
0 + n r  S20.ap = !S12
0 + n r  S20.ar = !S3
0 + n r  S20.clk = S106
0 - n r  !S20.d = 1
0 + n r  S3 = JE6 & S105 & JD16
0 + n r  S75 = S117 & S118 & S110 # S104
```

Group Definitions

Form type: pld-map Form name: sc10pa1

Partition: SC10PA1
Target Device Type: E0900 Output file format: jedec
Output File Name: sc10pa1
File Title Line:
Checksum Format: full Fast Flag: no
Simulation Form(s):
PLD-Sim? _ Trace: 0 X-level: 1 Z-level: h Breakpoints:
Display Pins:

Special Fuse Number: Value:

Pin Name	Usad As	Prod Terms	Pin Num	---Pin Attributes-- Act Lev	Reg Byp	Reg Typ	Feed Back
AUX101	Output	2	36	high	yes	com	none
AUX106	Input		2	high	yes	com	none
AUX109	Bidir	2	35	high	yes	com	none
AUX110	Bidir	2	34	high	yes	com	none
AUX111	Bidir	3	33	high	yes	com	none
AUX112	Bidir	2	16	high	yes	com	none
JC10	Input		3	high	yes	com	none
JD16	Input		4	high	yes	com	none
JD6	Output	1	32	high	yes	com	none
JE2	Input		5	high	no	com	none
JE6	Input		6	high	no	com	none
JF3	Input		7	high	yes	com	none
PA7	Output	1	31	high	yes	com	none
PB17	Output	1	30	high	yes	com	none
PB25	Output	1	29	high	yes	com	none
PB30	Output	1	28	high	yes	com	none
PB31	Output	1	15	high	yes	com	none
S104	Input		8	high	yes	com	none
S105	Input		9	high	yes	com	none
S106	Input		10	high	yes	com	none
S110	Input		11	high	no	com	none
S112	Output	2	27	high	yes	com	none
S117	Input		12	high	no	com	none
S118	Input		13	high	no	com	none
S3	Bidir	1	26	high	yes	com	none
S40	Input		14	high	yes	com	none
S50	Input		17	high	yes	com	none
S71	Input		18	high	yes	com	none
S74	Input		19	high	yes	com	none
S75	Bidir	2	25	high	yes	d	reg

Form type: pld-map Form name: sc10pa2

Partition: SC10PA2
Target Device Type: P331 Output file format: jedec
Output File Name: sc10pa2
File Title Line:
Checksum Format: full Fast Flag: no
Simulation Form(s):
PLD-Sim? _ Trace: 0 X-level: 1 Z-level: h Breakpoints:
Display Pins:

Special Fuse Number: Value:

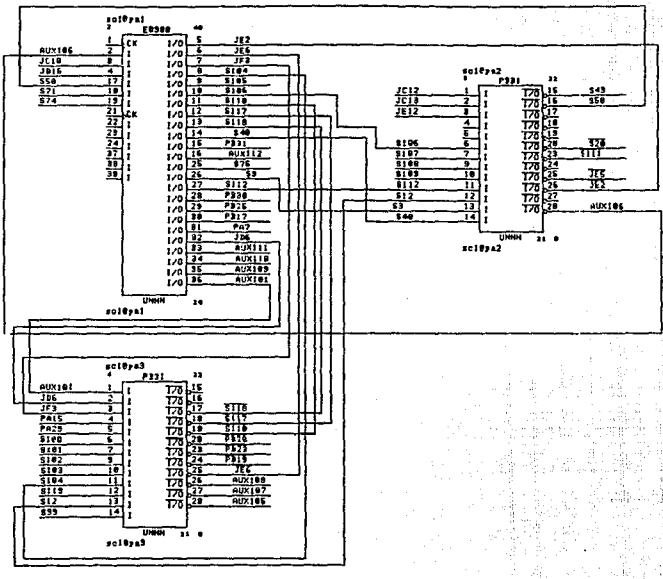
Pin Name	Used As	Prod Terms	Pin Num	---Pin Attributes-- Act Lev	Reg Byp	Reg Typ	Feed Back
AUX106	Output	3	28	high	no	d	reg
JC12	Input		1	high	yes	com	none
JC13	Input		2	high	yes	com	none
JE12	Input		3	high	yes	com	none
JE2	Bidir	6	26	high	no	d	reg
JE5	Output	1	25	high	no	d	reg
S106	Input		6	high	yes	com	none
S107	Input		7	high	yes	com	none
S108	Input		9	high	yes	com	none
S109	Input		10	high	yes	com	none
S111	Bidir	2	23	high	no	d	reg
S112	Input		11	high	yes	com	none
S12	Input		12	high	yes	com	none
S20	Output	1	20	high	no	d	reg
S3	Input		13	high	yes	com	none
S40	Input		14	high	yes	com	none
S43	Input		15	high	no	d	none
S50	Input		16	high	no	d	none

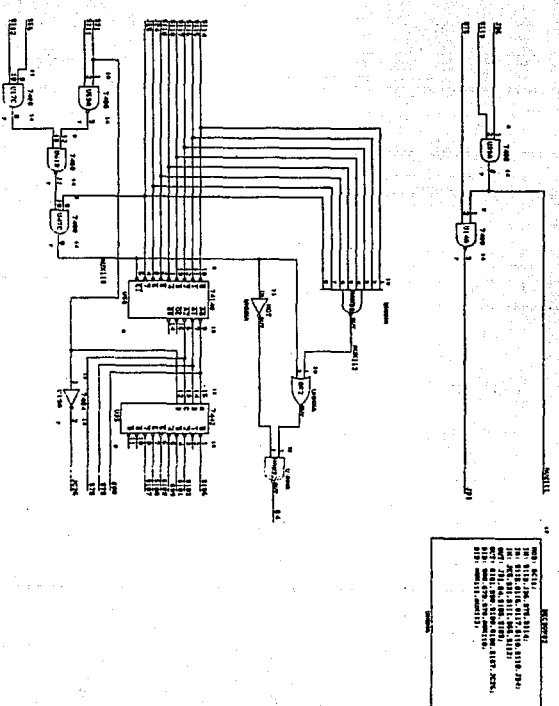
Form type: pld-map Form name: sc10pa3

Partition: SC10PA3
Target Device Type: P331 Output file format: jedec
Output File Name: sc10pa3
File Title Line:
Checksum Format: full Fast Flag: no
Simulation Form(s):
PLD-Sim? _ Trace: 0 X-Level: 1 Z-Level: h Breakpoints:
Display Pins:

Special Fuse Number: Value:

Pin Name	Used As	Prod Terms	Pin Num	---Pin Attributes-- Act Lev	Reg Byp	Reg Typ	Feed Back
AUX101	Input		1	high	yes	com	none
AUX105	Bidir	1	28	high	no	d	reg
AUX107	Bidir	1	27	high	no	d	reg
AUX108	Bidir	1	26	high	no	d	reg
JD6	Input		2	high	yes	com	none
JE6	Output	1	25	high	no	d	reg
JF3	Input		3	high	yes	com	none
PA15	Input		4	high	yes	com	none
PA29	Input		5	high	yes	com	none
PB19	Output	1	24	high	no	d	reg
PB23	Output	1	23	high	no	d	reg
PB28	Output	1	20	high	no	d	reg
S100	Input		6	high	yes	com	none
S101	Input		7	high	yes	com	none
S102	Input		9	high	yes	com	none
S103	Input		10	high	yes	com	none
S104	Input		11	high	yes	com	none
S110	Output	1	19	high	no	d	reg
S117	Output	1	18	high	no	d	reg
S118	Output	1	17	high	no	d	reg
S119	Input		12	high	yes	com	none
S12	Input		13	high	yes	com	none
S99	Input		14	high	yes	com	none


scl@ps1
E0900
UNNN
scl@ps1
scl@ps2
P331
UNNN
scl@ps2
scl@ps3
P331
UNNN
scl@ps3



Form type: reduction Form name: Reduce

```
Reduction Level
 Polarity
  Exclusive-OR
   Display
-------------------------------------------------------------------------------
e + n r  AUX110 = JE5 & S31 & S111 # JE5 & !S112 # JE5 & !S66
e + n r  AUX111 = JD6 & S113
e + n r  AUX112 =
           !JE5
         # !JD4
         # !S118
         # !S110
         # !S117
         # !S116
         # !S115
         # !S114
e + n r  JC26 = !S31
e + n r  JD1 = !S75 # !AUX111
e + n r  S101 = S31 # S78 # !S79 # !S80
e + n r  S103 = S31 # S78 # !S79 # S80
e + n r  S106 = S31 # S78 # S79 # S80
e + n r  S107 = S31 # !S78 # !S79 # !S80
e + n r  S108 = S31 # !S78 # !S79 # S80
e + n r  S109 = S31 # !S78 # S79 # !S80
e + n r  S4 = !AUX112 # AUX110
e + n r  S78 = S110 & S118 & JD4 & JE5 # AUX110
e + n r  S79 =
           S116 & S117 & JD4 & JE5
         # JD4 & JE5 & !S118
         # JD4 & JE5 & !S110
         # AUX110
e + n r  S80 =
           S117 & S118 & JE5 & !S116
         # S115 & S117 & S118 & JE5
         # S118 & JE5 & !S110
         # JE5 & !JD4
         # AUX110
e + n r  S99 = S31 # !S78 # S79 # S80

                          Group Definitions
-------------------------------------------------------------------------------
```

Form type: pld-map Form name: sc11pal

Partition: SC11PA1
Target Device Type: E0600 Output file format: jedec
Output File Name: sc11pal
File Title Line:
Checksum Format: full Fast Flag: no
Simulation Form(s):
PLD-Sim? _ Trace: 0 X-level: 1 Z-level: h Breakpoints:
Display Pins:

Special Fuse Number: Value:

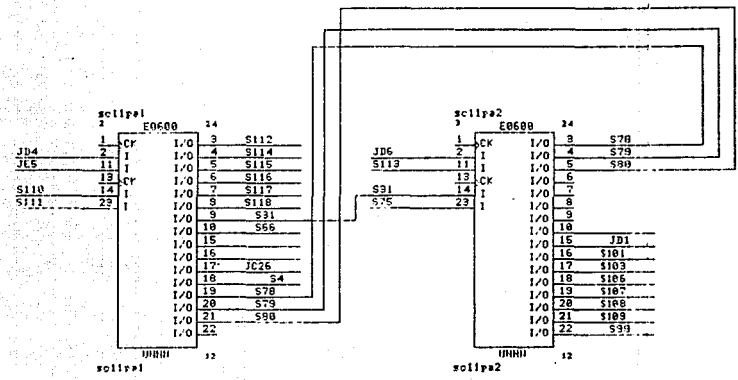
Pin Name	Used As	Prod Terms	Pin Num	---Pin Attributes-- Act Lev	Reg Byp	Reg Typ	Feed Back
AUX110	Bidir	3	15	high	yes	com	none
AUX112	Bidir	8	16	high	yes	com	none
JC26	Output	1	17	high	yes	com	none
JD4	Input		2	high	yes	com	none
JE5	Input		11	high	yes	com	none
S110	Input		14	high	yes	com	none
S111	Input		23	high	yes	com	none
S112	Input		3	high	yes	com	none
S114	Input		4	high	yes	com	none
S115	Input		5	high	yes	com	none
S116	Input		6	high	yes	com	none
S117	Input		7	high	yes	com	none
S118	Input		8	high	yes	com	none
S31	Input		9	high	yes	com	none
S4	Output	2	18	high	yes	com	none
S66	Input		10	high	yes	com	none
S78	Output	2	19	high	yes	com	none
S79	Output	4	20	high	yes	com	none
S80	Output	5	21	high	yes	com	none

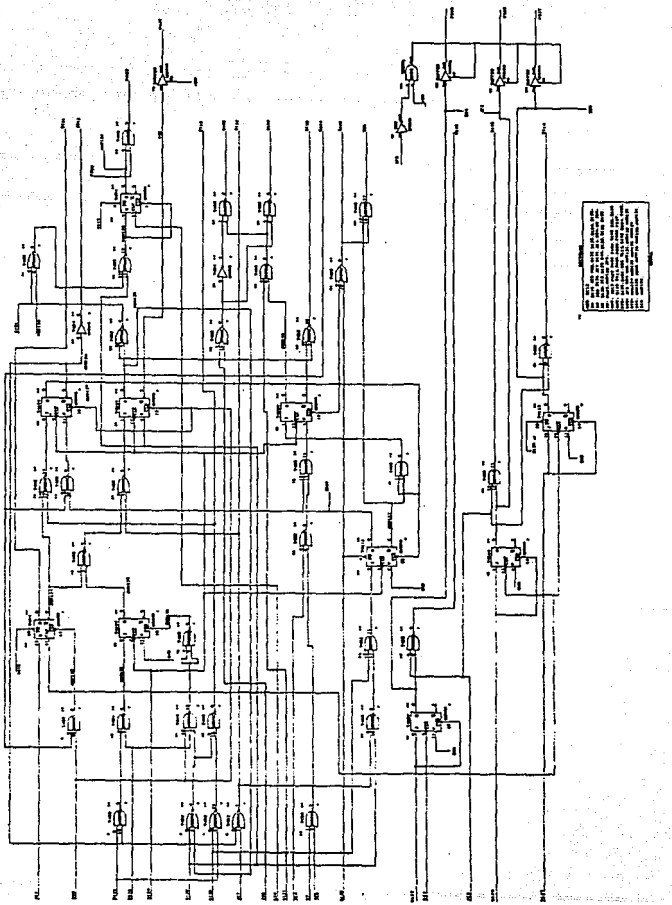
Form type: pld-map Form name: sc11pa2

Partition: SC11PA2
Target Device Type: E0600 Output file format: jedec
Output File Name: sc11pa2
File Title Line:
Checksum Format: full Fast Flag: no
Simulation Form(s):
PLD-Sim? _ Trace: 0 X-level: 1 Z-level: h Breakpoints:
Display Pins:

Special Fuse Number: Value:

Pin Name	Used As	Prod Terms	Pin Num	---Pin Attributes--- Act Lev	Reg Byp	Reg Typ	Feed Back
AUX111	Bidir	1	10	high	yes	com	none
JD1	Output	2	15	high	yes	com	none
JD6	Input		2	high	yes	com	none
S101	Output	4	16	high	yes	com	none
S103	Output	4	17	high	yes	com	none
S106	Output	4	18	high	yes	com	none
S107	Output	4	19	high	yes	com	none
S108	Output	4	20	high	yes	com	none
S109	Output	4	21	high	yes	com	none
S113	Input		11	high	yes	com	none
S31	Input		14	high	yes	com	none
S75	Input		23	high	yes	com	none
S78	Input		3	high	yes	com	none
S79	Input		4	high	yes	com	none
S80	Input		5	high	yes	com	none
S99	Output	4	22	high	yes	com	none


sc11ps1
E0600
JD4
JE5
S110
S111
S112
S114
S115
S116
S117
S118
S31
S66
JC26
S4
S78
S79
S80
sc11ps1
sc11ps2
E0600
JD6
S113
S31
S75
S78
S79
S80
JD1
$101
$103
$105
$107
$108
$109
S91
sc11ps2



201

Form type: reduction Form name: Reduce

```
Reduction Level
 Polarity
  Exclusive-OR
   Display
-------------------------------------------------------------------------------
e + n r  AUX121.ap = !S119
e + n r  AUX121.ar = !AUX129
e + n r  AUX121.clk = S24
e - n r  !AUX121.d = !AUX129
e + n r  AUX122.ap = !S120
e + n r  AUX122.ar = !AUX127
e + n r  AUX122.clk = !S124
e - n r  !AUX122.d = S132 & S125 # S126 # JE00
e + n r  AUX123.ar = !AUX127
e + n r  AUX123.clk = !S124
e - n r  !AUX123.d =
           AUX123 & JE00 & S7 & JE8
         # AUX123 & JE00 & JE4
         # !AUX123 & !AUX122
         # !AUX123 & !AUX127
e + n r  AUX124.ap = !S119
e + n r  AUX124.ar = !AUX130
e + n r  AUX124.clk = S44
e - n r  !AUX124.d = JE3
e + n r  AUX125.ar = !AUX131
e + n r  AUX125.clk = !S124
e - n r  !AUX125.d = !AUX125 & !AUX132
e + n r  AUX126.ar = !S50
e + n r  AUX126.clk = !S124
e - n r  !AUX126.d =
           AUX126 & S125
         # !AUX126 & S132
         # !AUX126 & !AUX125
         # !AUX126 & !AUX124
e + n r  AUX127.ar = !S50
e + n r  AUX127.clk = !S124
e - n r  !AUX127.d =
           AUX127 & S104 & JE00 # !AUX127 & S113 # !AUX127 & AUX124
e + n r  AUX128.ar = !S50
e + n r  AUX128.clk = !S124
e - n r  !AUX128.d =
           !AUX127 & !AUX124 & !S113
         # AUX127 & !JE00
         # AUX127 & !S104
e + n r  AUX129 = !AUX120 & !S101 & !S125
e + n r  AUX130 = S104 & S50
e + n r  AUX131 = S50 & S123 & !S125 & !AUX126
e + n r  AUX132 = S123 & !AUX128 # S123 & !S122
e + n r  JD13.ar = !S50
e + n r  JD13.clk = !S124
e - n r  !JD13.d = AUX128
e + n r  JE00.ap = !S120
e + n r  JE00.ar = !AUX127
```

```
e + n r  JE00.clk = !S124
e - n r  !JE00.d = !JE00 & !S126 & !S125 # !JE00 & !S126 & !S132
e + n r  JF2.ar = !S128
e + n r  JF2.clk = !S44
e - n r  !JF2.d = !JF2 & !S128
e + n r  PA22 = !AUX121 # !PA23
e + n r  PA35.ar = !S50
e + n r  PA35.clk = !S124
e - n r  !PA35.d = !S95
e + n r  PA35.oe = 1
e + n r  PB20.ar = !S127
e + n r  PB20.clk = !S44
e - n r  !PB20.d = !S84
e + n r  PB20.oe = !JF3
e + n r  PB27.ap = !S130
e + n r  PB27.ar = !S129
e + n r  PB27.clk = !S44
e - n r  !PB27.d = !S86
e + n r  PB27.oe = !JF3
e + n r  PB32.ar = !S128
e + n r  PB32.clk = !S44
e - n r  !PB32.d = !JF2
e + n r  PB32.oe = !JF3
e + n r  S100 = !JD3 & !S132 # !S121 # !AUX123
e + n r  S101 = !S132 & !AUX126
e + n r  S102 = !S121 # !AUX123 # S132 # JD3
e + n r  S104.ar = !AUX127
e + n r  S104.clk = !S124
e - n r  !S104.d =
           !AUX123 & AUX127 & AUX122
         # AUX123 & !JE8 & !JE4
         # AUX123 & !S7 & !JE4
         # AUX123 & !JE00
e + n r  S105 = S120 # AUX127
e + n r  S113 = S122 # S126 # AUX126 # S125
e + n r  S114 = !JE2 # !S86
e + n r  S115 = JE2 & JF2
e + n r  S116 = !JE2 # !S84
e + n r  S131.ap = !S119
e + n r  S131.ar = !AUX130
e + n r  S131.clk = S44
e - n r  !S131.d = !JE3
e + n r  S132 = !AUX128 & !JE7
e + n r  S133 = !S104 & !S132
e + n r  S26 = AUX122 & S105
e + n r  S84.ar = !S127
e + n r  S84.clk = !S44
e - n r  !S84.d = !S84 & !S127
e + n r  S86.ap = !S130
e + n r  S86.ar = !S129
e + n r  S86.clk = !S44
e - n r  !S86.d = !S86 & !S129
e + n r  S95.ar = !S50
e + n r  S95.clk = !S124
e - n r  !S95.d = !AUX126 & AUX124 & AUX125 & !S132 # AUX126 & !S125
```

Group Definitions

Form type: pld-map Form name: sc12pa1

Partition: SC12PA1
Target Device Type: P331 Output file format: jedec
Output File Name: sc12pa1
File Title Line:
Checksum Format: full Fast Flag: no
Simulation Form(s):
PLD-Sim? _ Trace: 0 X-level: 1 Z-level: h Breakpoints:
Display Pins:

Special Fuse Number: Value:

Pin Name	Used As	Prod Terms	Pin Num	Act Lev	Reg Byp	Reg Typ	Feed Back
AUX121	Output	1	28	high	no	d	reg
AUX124	Bidir	1	27	high	no	d	reg
AUX125	Bidir	1	26	high	no	d	reg
AUX126	Bidir	4	25	high	no	d	reg
AUX127	Bidir	3	24	high	no	d	reg
AUX128	Bidir	3	23	high	no	d	reg
AUX129	Input	___	1	high	yes	com	none
AUX130	Input	___	2	high	yes	com	none
AUX131	Input	___	3	high	yes	com	none
AUX132	Input	___	4	high	yes	com	none
JD13	Output	1	20	high	no	d	reg
JE00	Input	___	6	high	yes	com	none
JE3	Input	___	7	high	yes	com	none
PA35	Output	1	19	high	no	d	reg
S104	Input	___	9	high	yes	com	none
S113	Input	___	10	high	yes	com	none
S119	Input	___	11	high	yes	com	none
S124	Input	___	12	high	yes	com	none
S125	Input	___	13	high	yes	com	none
S131	Output	1	18	high	no	d	reg
S132	Input	___	14	high	yes	com	none
S24	Input	___	16	high	no	d	none
S44	Input	___	15	high	no	d	none
S50	Input	___	5	high	yes	com	none
S95	Bidir	2	17	high	no	d	reg

Form type: pld-map Form name: sc12pa2

Partition: SC12PA2
Target Device Type: E0600 Output file format: jedec
Output File Name: sc12pa2
File Title Line:
Checksum Format: full Fast Flag: no
Simulation Form(s):
PLD-Sim? _ Trace: 0 X-Level: 1 Z-Level: h Breakpoints:
Display Pins:

Special Fuse Number: Value:

Pin Name	Used As	Prod Terms	Pin Num	---Pin Attributes-- Act Lev	Reg Byp	Reg Typ	Feed Back
AUX120	Input		2	high	yes	com	none
AUX121	Input		3	high	yes	com	none
AUX126	Input		4	high	yes	com	none
AUX128	Input		5	high	yes	com	none
AUX129	Output	1	22	high	yes	com	none
AUX130	Output	1	21	high	yes	com	none
AUX131	Output	1	20	high	yes	com	none
AUX132	Output	2	19	high	yes	com	none
JE7	Input		6	high	yes	com	none
PA22	Output	2	18	high	yes	com	none
PA23	Input		7	high	yes	com	none
S101	Bidir	1	17	high	yes	com	none
S104	Input		8	high	yes	com	none
S113	Output	4	16	high	yes	com	none
S122	Input		9	high	yes	com	none
S123	Input		10	high	yes	com	none
S125	Input		11	high	yes	com	none
S126	Input		14	high	yes	com	none
S132	Bidir	1	15	high	yes	com	none
S50	Input		23	high	yes	com	none

Form type: pld-map Form name: sc12pa3

Partition: SC12PA3
Target Device Type: E0600 Output file format: jedec
Output File Name: sc12pa3
File Title Line:
Checksum Format: full Fast Flag: no
Simulation Form(s):
PLD-Sim? _ Trace: 0 X-level: 1 Z-level: h Breakpoints:
Display Pins:

Special Fuse Number: Value:

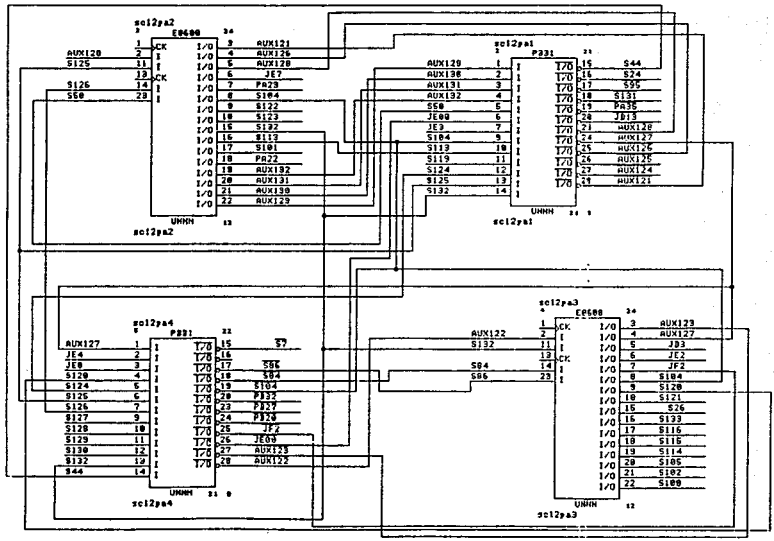
Pin Name	Used As	Prod Terms	Pin Num	Act Lev	Reg Byp	Reg Typ	Feed Back
				---Pin Attributes---			
AUX122	Input		2	high	yes	com	none
AUX123	Input		3	high	yes	com	none
AUX127	Input		4	high	yes	com	none
JD3	Input		5	high	yes	com	none
JE2	Input		6	high	yes	com	none
JF2	Input		7	high	no	com	none
S100	Output	3	22	high	yes	com	none
S102	Output	4	21	high	yes	com	none
S104	Input		8	high	no	com	none
S105	Bidir	2	20	high	yes	com	none
S114	Output	2	19	high	yes	com	none
S115	Output	1	18	high	yes	com	none
S116	Output	2	17	high	yes	com	none
S120	Input		9	high	yes	com	none
S121	Input		10	high	yes	com	none
S132	Input		11	high	yes	com	none
S133	Output	1	16	high	yes	com	none
S26	Output	1	15	high	yes	com	none
S84	Input		14	high	yes	com	none
S86	Input		23	high	yes	com	none

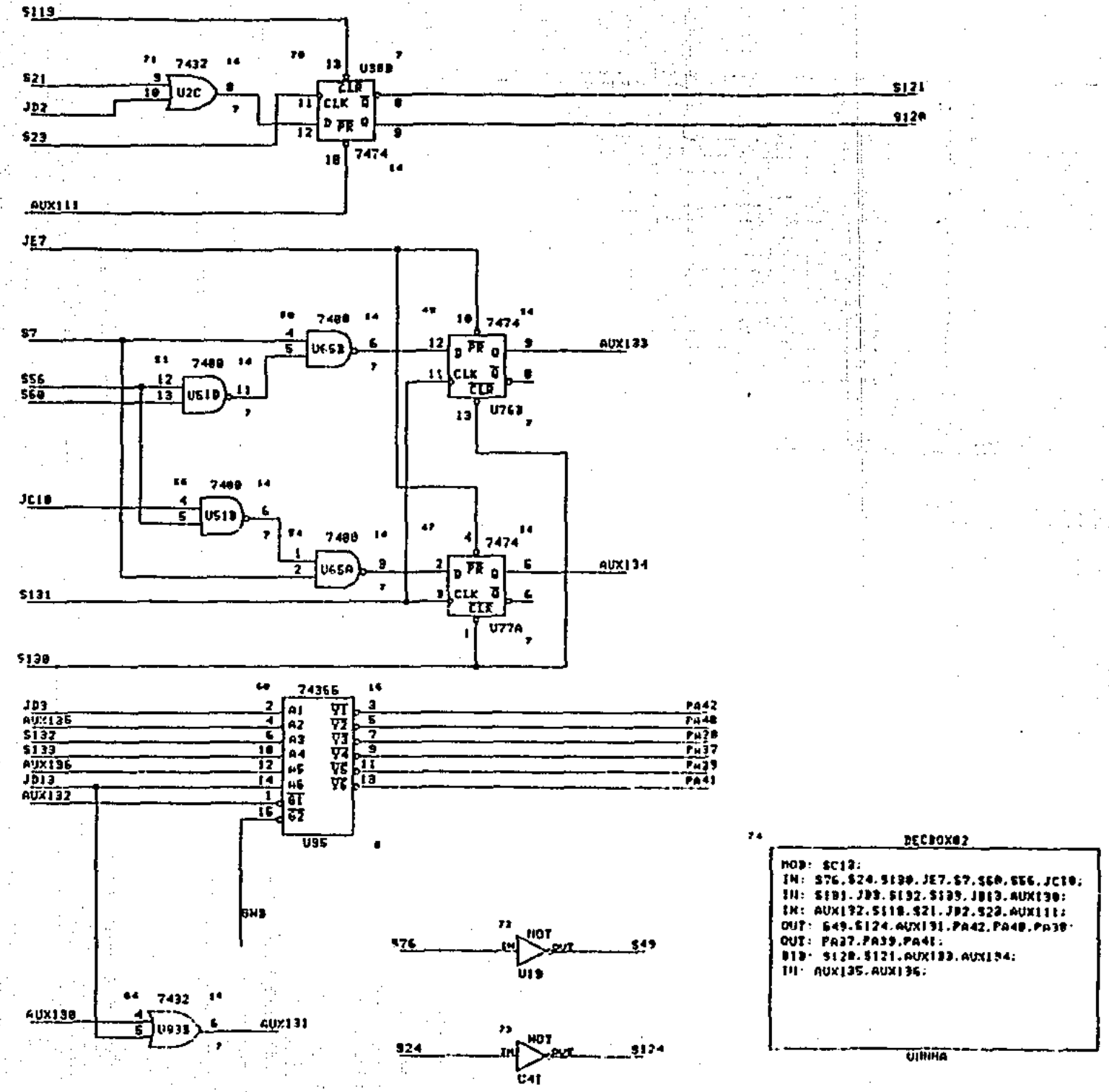
Form type: pld-asp Form name: sc12pa4

Partition: SC12PA4
Target Device Type: P331 Output file format: jedec
Output File Name: sc12pa4
File Title Line:
Checksum Format: full Fast Flag: no
Simulation Form(s):
PLD-Sim? _ Trace: 0 X-Level: 1 Z-Level: h Breakpoints:
Display Pins:

Special Fuse Number: Value:

Pin Name	Used As	Prod Terms	Pin Num	---Pin Attributes--- Act Lev	Reg Byp	Reg Typ	Feed Back
AUX122	Bidir	3	28	high	no	d	reg
AUX123	Bidir	4	27	high	no	d	reg
AUX127	Input		1	high	yes	com	none
JE00	Bidir	2	26	high	no	d	reg
JE4	Input		2	high	yes	com	none
JE8	Input		3	high	yes	com	none
JF2	Bidir	1	25	high	no	d	reg
JF3	Input		16	high	no	d	none
PB20	Output	1	24	high	no	d	reg
PB27	Output	1	23	high	no	d	reg
PB32	Output	1	20	high	no	d	reg
S104	Output	4	19	high	no	d	reg
S120	Input		4	high	yes	com	none
S124	Input		5	high	yes	com	none
S125	Input		6	high	yes	com	none
S126	Input		7	high	yes	com	none
S127	Input		9	high	yes	com	none
S128	Input		10	high	yes	com	none
S129	Input		11	high	yes	com	none
S130	Input		12	high	yes	com	none
S132	Input		13	high	yes	com	none
S44	Input		14	high	yes	com	none
S7	Input		15	high	no	d	reg
S84	Bidir	1	18	high	no	d	reg
S86	Bidir	1	17	high	no	d	reg


sc12pa2
E0680
UNNN
sc12pa1
P331
UNNN
sc12pa4
P331
UNNN
sc12pa3
E0608
UNNN


S119
S21
JD2
S23
7432
U2C
U38B
7474
S121
S120
AUX111
JE7
S7
S56
S60
7408
U65B
U61B
U76B
AUX133
JC10
U51B
U65A
U77A
AUX134
S131
S130
74366
U95
JD3
AUX135
S132
S133
AUX136
JD13
AUX132
PA42
PA40
PA38
PA37
PA39
PA41
GND
S76
HOT
U19
S49
AUX130
U93B
AUX131
S24
U41
S124
DECBOX02
MOD: SC13;
IN: S76,S24,S130,JE7,S7,S60,S56,JC10;
IN: S131,JD3,S132,S133,JD13,AUX130;
IN: AUX132,S119,S21,JD2,S23,AUX111;
OUT: S49,S124,AUX131,PA42,PA40,PA38;
OUT: PA37,PA39,PA41;
BID: S120,S121,AUX133,AUX134;
IN: AUX135,AUX136;
OINHA

Form type: reduction Form name: Reduce

Reduction Level
Polarity
Exclusivo-OR
Display

```
------------------------------------------------------------------------------
e + n r  AUX131 = AUX130 # JD13
e + n r  AUX133.ap = !S130
e + n r  AUX133.ar = !JE7
e + n r  AUX133.clk = S131
e - n r  !AUX133.d = !S60 & S7 # !S56 & S7
e + n r  AUX134.ap = !S130
e + n r  AUX134.ar = !JE7
e + n r  AUX134.clk = S131
e - n r  !AUX134.d = !JC10 & S7 # !S56 & S7
e + n r  PA37 = !S133
e + n r  PA37.oe = !AUX132
e + n r  PA38 = !S132
e + n r  PA38.oe = !AUX132
e + n r  PA39 = !AUX136
e + n r  PA39.oe = !AUX132
e + n r  PA40 = !AUX135
e + n r  PA40.oe = !AUX132
e + n r  PA41 = !JD13
e + n r  PA41.oe = !AUX132
e + n r  PA42 = !JD3
e + n r  PA42.oe = !AUX132
e + n r  S120.ap = !S119
e + n r  S120.ar = !AUX111
e + n r  S120.clk = S23
e - n r  !S120.d = !S21 & !JD2
e + n r  S121.ap = !AUX111
e + n r  S121.ar = !S119
e + n r  S121.clk = S23
e - n r  !S121.d = JD2 # S21
e + n r  S124 = !S24
e + n r  S49 = !S76
```

Group Definitions

Form type: pld-map Form name: sc13pal

Partition: SC13PA1
Target Device Type: PJ31 Output file format: jedec
Output File Name: sc13pal
File Title Line:
Checksum Format: full Fast Flag: no
Simulation Form(s):
PLD-Sim? _ Trace: 0 X-Level: 1 Z-Level: h Breakpoints:
Display Pins:

Special Fuse Number: Value:

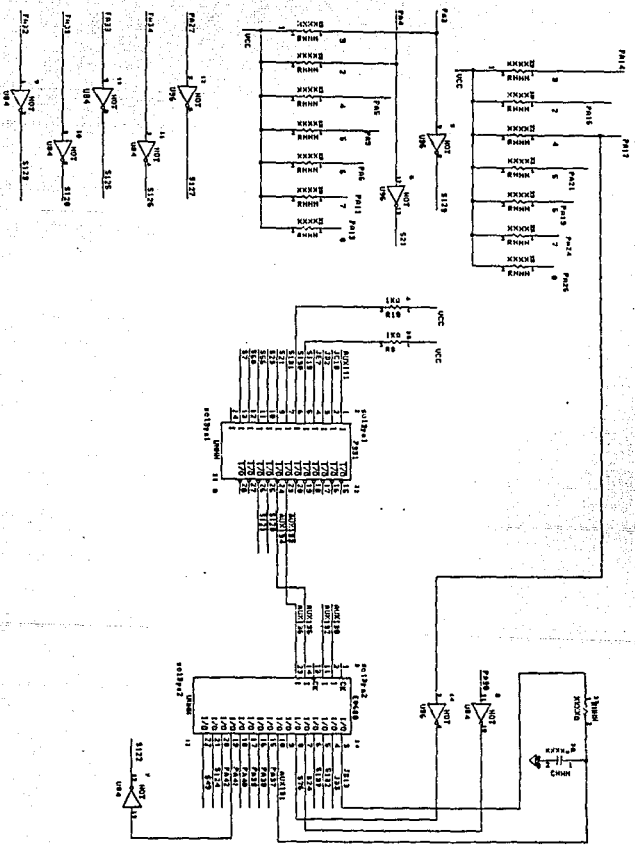
Pin Name	Used As	Prod Terms	Pin Num	Act Lev	Reg Byp	Reg Typ	Feed Back
AUX111	Input		1	high	yes	com	none
AUX133	Output	2	23	high	no	d	reg
AUX134	Output	2	24	high	no	d	reg
JC10	Input		2	high	yes	com	none
JD2	Input		3	high	yes	com	none
JE7	Input		4	high	yes	com	none
S119	Input		5	high	yes	com	none
S120	Output	1	25	high	no	d	reg
S121	Output	2	26	high	no	d	reg
S130	Input		6	high	yes	com	none
S131	Input		7	high	yes	com	none
S21	Input		9	high	yes	com	none
S23	Input		10	high	yes	com	none
S56	Input		11	high	yes	com	none
S60	Input		12	high	yes	com	none
S7	Input		13	high	yes	com	none

Form type: pld-map Form name: sc13pa2

Partition: SC13PA2
Target Device Type: E0600 Output file format: jedec
Output File Name: sc13pa2
File Title Line:
Checksum Format: Full Fast Flag: no
Simulation Form(s):
PLD-Sim? _ Trace: 0 X-level: 1 Z-level: h Breakpoints:
Display Pins:

Special Fuse Number: Value:

Pin Name	Used As	Prod Terms	Pin Num	Act Lev	Reg Byp	Reg Typ	Feed Back
AUX130	Input		2	high	yes	com	none
AUX131	Output	2	10	high	yes	com	none
AUX132	Input		11	high	yes	com	none
AUX135	Input		14	high	yes	com	none
AUX136	Input		23	high	yes	com	none
JD13	Input		3	high	yes	com	none
JD3	Input		4	high	yes	com	none
PA37	Output	1	15	high	yes	com	none
PA38	Output	1	16	high	yes	com	none
PA39	Output	1	17	high	yes	com	none
PA40	Output	1	18	high	yes	com	none
PA41	Output	1	19	high	yes	com	none
PA42	Output	1	20	high	yes	com	none
S124	Output	1	21	high	yes	com	none
S132	Input		5	high	yes	com	none
S133	Input		6	high	yes	com	none
S24	Input		7	high	yes	com	none
S49	Output	1	22	high	yes	com	none
S76	Input		8	high	yes	com	none



APENDICE B

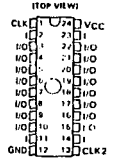
INFORMACION TECNICA DE
PLD's UTILIZADOS

EP610
HIGH-PERFORMANCE 16-MACROCELL ERASABLE PROGRAMMABLE LOGIC DEVICE (EPLD)

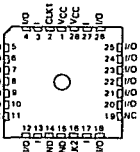
D3177, OCTOBER 1986 – AUGUST 1990

- High-Density (Over 600 Gates) Replacement for TTL and 74HC
- Virtually Zero Standby Power . . . Typ 20 µA
- High Speed:
  Propagation Delay Time . . . 25 ns
- Asynchronous Clocking of All Registers or Banked Register Operation from 2 Synchronous Clocks
- Sixteen Macrocells with Configurable I/O Architecture Allowing for Up to 20 Inputs and 16 Outputs
- Each Output Macrocell User-Programmable for D, T, SR, or JK Flip-Flops with Individual Clear Control or Combinational Operation
- UV-Light-Erasable Cell Technology Allows for:
  – Reconfigurable Logic
  – Reprogrammable Cells
  – Full Factory Testing for 100% Programming Yields
- Programmable Design Security Bit Prevents Copying of Logic Stored in Device
- Advanced Software Support Featuring Schematic Capture, Interactive Netlist, Boolean Equations, and State Machine Design Entry
- Package Options Include Plastic (for One-Time-Programmable (OTP) Devices) and Ceramic Dual-In-Line Packages and Chip Carriers

DUAL-IN-LINE PACKAGE
(TOP VIEW)



CHIP CARRIER PACKAGE
(TOP VIEW)



NC – No internal connection

AVAILABLE OPTIONS

T_A RANGE	SPEED CLASS	PACKAGE TYPE			
		CERAMIC DUAL-IN-LINE PACKAGE (CDIP)	CERAMIC CHIP CARRIER (CLCC)	PLASTIC† DUAL-IN-LINE PACKAGE (PDIP)	PLASTIC† CHIP CARRIER (PLCC)
0°C – 70°C	25 ns	EP610DC-25	EP610JC-25	EP610PC-25	EP610LC-25
	30 ns	EP610DC-30	EP610JC-30	EP610PC-30	EP610LC-30
	35 ns	EP610DC-35	EP610JC-35	EP610PC-35	EP610LC-35
–40°C – 85°C	40 ns	EP610DI-40	EP610JI-40	EP610PI-40	EP610LI-40

†This package is for one-time programmable (OTP) devices.

TEXAS INSTRUMENTS
POST OFFICE BOX 655012 • DALLAS, TEXAS 75265

description

general

The Texas Instruments EP610 Erasable Programmable Logic Device is capable of implementing over 600 equivalent gates of SSI and MSI logic functions all in plastic and ceramic space-saving 24-pin, 300-mil dual-in-line (DIP) packages and 28-pin chip-carrier packages. It uses the familiar sum-of-products logic, providing a programmable AND with a fixed OR structure. The device accommodates both combinational and sequential (registered) logic functions with up to 20 inputs and 16 outputs. The EP610 has a user programmable output logic macrocell that allows each output to be configured as a combinational or registered output and feedback signals active high or active low.

A unique feature of the EP610 is the ability to program D, T, SR, or JK flip-flop operation individually for each output without sacrificing product terms. In addition, each register can be individually clocked from any of the input or feedback paths available in the AND array. These features allow a variety of logic functions to be simultaneously implemented.

The CMOS EPROM technology reduces the power consumption to less than 20% of equivalent bipolar devices without sacrificing speed performance. Erasable EPROM bits allow for enhanced factory testing. Design changes can be easily implemented by erasing the device with ultraviolet (UV) light.

Programming the EP610 is accomplished by using the TI EPLD Development System, which supports four different design entry methods. When the design has been entered, the software performs automatic translation into logical equations, Boolean minimization, and design fitting directly into an EPLD.

functional

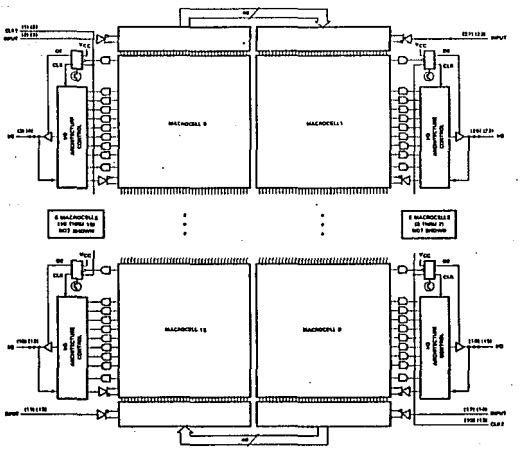
The EP610 is an Erasable Programmable Logic Device (EPLD) that uses a CMOS EPROM technology to implement logic designs in a programmable AND logic array. The device contains a revolutionary programmable I/O architecture that provides advanced functional capability for user programmable logic.

Externally, the EP610 provides 4 dedicated data inputs and 16 I/O pins, which may be configured for input, output, or bidirectional operation. Figure 1 shows the EP610 basic logic array macrocell. The internal architecture is organized with familiar sum-of-products (AND-OR) structure. Inputs to the programmable AND array come from true and complement signals from the 4 dedicated data inputs and the 16 I/O architecture-control blocks. The 40-input AND array encompasses 160 product terms, which are distributed among 16 available macrocells. Each EP610 product term represents a 40-input AND gate.

Each macrocell contains 10 product terms, 8 of which are dedicated for logic implementation. One product term is used for clear control of the macrocell internal register. The remaining product terms are used for output enable/asynchronous clock implementation.

There is an EPROM connection at the intersection point of each input signal and each product term. In the erased state, all connections are made. This means both the true and complement forms of all inputs are connected to each product term. Connections are opened during the programming process. Therefore, any product term may be connected to the true or complement form of any array input signal.

functional block diagram



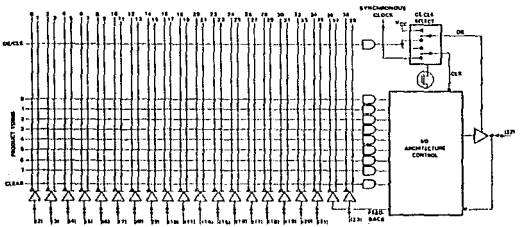
Pin numbers in ( ) are for DIP packages; pin numbers in [ ] are for chip carrier packages.

When both the true and complement forms of any signal are left intact, a logical false state results on the output of the AND gate. If both the true and complement connections are open, then a logical "don't care" applies for that input. If all inputs for the product term are programmed open, then a logical true state results on the output of the AND gate.

Two dedicated clock inputs provide synchronous clock signals to the EP610 internal registers. Each of the clock signals controls a bank of 8 registers. CLK1 controls registers associated with macrocells 9-16, and CLK2 controls registers associated with macrocells 1-8. The EP610 advanced I/O architecture allows the number of synchronous registers to be user defined, from one to sixteen. Both dedicated clock inputs are positive-edge triggered.

I/O architecture

The EP610 input/output architecture provides each macrocell with over 50 possible I/O configurations. Each I/O can be configured for combinational or registered output, with programmable output polarity. Four different types of registers (D, T, JK, and SR) can be implemented into every I/O without any additional logic requirements. I/O feedback selection can also be programmed for registered or input (pin) feedback. Another benefit of the EP610 I/O architecture is its ability to individually clock each internal register from asynchronous clock signals.



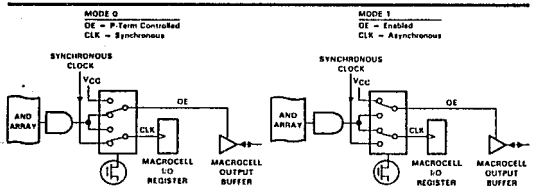
Pin numbers are for dual in line packages

FIGURE 1. LOGIC ARRAY MACROCELL (MACROCELL 1 ILLUSTRATED)

OE/CLK selection

Figure 2 shows the two modes of operation that are provided by the OE/CLK select multiplexer. The operation of this multiplexer is controlled by a single EPROM bit and may be individually configured for each EP610 I/O pin. In Mode 0, the 3-state output buffer is controlled by a single product term. If the output of the AND gate is true, the output buffer is enabled. If the output of the AND gate is false, the output buffer is in the high-impedance state. In this mode, the macrocell flip-flop may be clocked by its respective synchronous clock input. After erasure, the OE/CLK select multiplexer is configured in Mode 0.

In Mode 1, the output-enable buffer is always enabled. The macrocell flip-flop may now be triggered from an asynchronous clock signal generated by the OE/CLK multiplexable product term. This mode allows individual clocking of flip-flops from any available signal in the AND array. Because both true and complement signals reside in the AND array, the flip-flop may be configured for positive- or negative edge-triggered operation. With the clock now controlled by a product term, gated clock structures are also possible.



The register is clocked by the synchronous clock signal, which is common to 11 other Macrocells. The output is enabled by the logic from the product term.

The output is permanently enabled and the register is clocked via the product term. This allows for gated clocks that may be generated from elsewhere in the EP610.

FIGURE 2. OE/CLK SELECT MULTIPLEXER

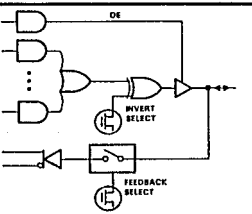
output/feedback selection

Figure 3 shows the EP610 basic output configurations. Along with combinational output, 4 register types are available. Each macrocell I/O may be independently configured. All registers have individual asynchronous clear control from a dedicated product term. When the product term is asserted, the macrocell register will immediately be loaded with a zero independently of the clock. On power-up, the EP610 performs the clear function automatically.

When the D or T register is selected, 8 product terms are ORed together and made available to the register input. The invert select EPROM bit determines output polarity. The feedback-select multiplexer enables register, I/O (pin), or no feedback to the AND array.

If the JK or SR registers are selected, the 8 product terms are shared between 2 OR gates. The allocation of product terms for each register input is optimized by the TI EPLD Development System. The invert select EPROM bit configures output polarity. The feedback-select multiplexer enables registered or no feedback to the AND array.

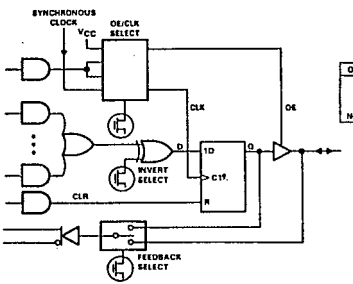
Any I/O pin may be configured as a dedicated input by selecting no output and pin feedback. No output is obtained by disabling the macrocell output buffer. In the erased state, each I/O is configured for combinational active-low output with input (pin) feedback.



I/O SELECTION

OUTPUT/POLARITY	FEEDBACK
Combinational/High	Pin, None
Combinational/Low	Pin, None
None	Pin

(a) COMBINATIONAL



I/O SELECTION

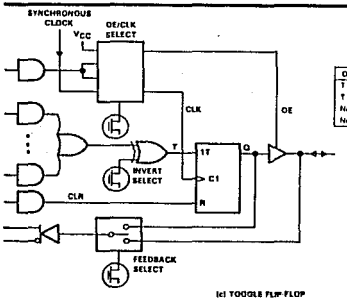
OUTPUT/POLARITY	FEEDBACK
JK Register/High	JK Register, None
JK Register/Low	JK Register, None
None	JK Register
None	Pin

FUNCTION TABLE

INPUTS			OUTPUT
CLR	CLK	D	Q
L	↑	L	L
L	↑	H	H
L	L or H	X	Q_0
H	X	X	L

(b) D-TYPE FLIP FLOP

FIGURE 3. I/O CONFIGURATIONS



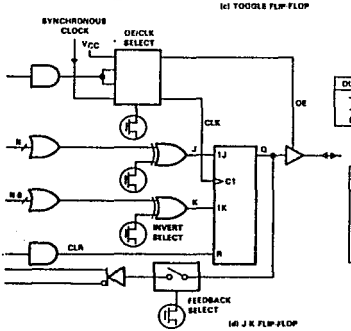
I/O SELECTION

OUTPUT/POLARITY	FEEDBACK
T Register/high	T Register, Pin, None
T Register/low	T Register, Pin, None
None	T Register
None	Pin

FUNCTION TABLE

INPUTS			OUTPUT
CLR	CLK	T	Q
L	↑	L	Q_0
L	↑	H	$\bar{Q}_0$
L	L or H	X	Q_0
H	X	X	L

(c) TOGGLE FLIP-FLOP



I/O SELECTION

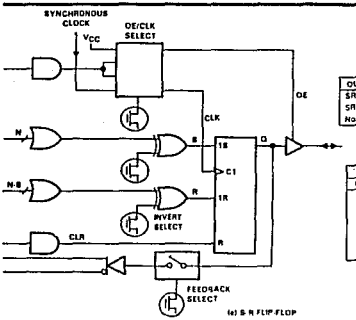
OUTPUT/POLARITY	FEEDBACK
JK Register/high	JK Register, None
JK Register/low	JK Register, None
None	JK Register

FUNCTION TABLE

INPUTS				OUTPUT
CLR	CLK	J	K	Q
L	↑	L	L	Q_0
L	↑	L	H	L
L	↑	H	L	H
L	↑	H	H	$\bar{Q}_0$
L	L or H	X	X	Q_0
H	X	X	X	L

(d) J K FLIP-FLOP

FIGURE 3. I/O CONFIGURATIONS (CONTINUED)



I/O SELECTION

OUTPUT/POLARITY	FEEDBACK
SR Register/high	SR Register, None
SR Register/low	SR Register, None
None	SR Register

FUNCTION TABLE

INPUTS				OUTPUT
CLR	CLK	S	R	Q
L	↑	L	L	Q_0
L	↑	L	H	L
L	↑	H	L	H
L	↑	H	H	Undefined
L	L or H	X	X	Q_0
H	X	X	X	L

FIGURE 3. I/O CONFIGURATIONS (CONTINUED)

absolute maximum ratings over operating free-air temperature range (unless otherwise noted)

Supply voltage range, V_{CC} (see Note 1) . . . . . −0.3 V to 7 V
Instantaneous supply voltage range, V_{CC} ($t \leq 20$ ns) . . . . . −2 V to 7 V
Programming supply voltage, V_{PP} . . . . . −0.3 V to 13.5 V
Instantaneous programming supply voltage range, V_{PP} ($t \leq 20$ ns) . . . . . −2 V to 13.5 V
Input voltage range, V_I . . . . . −0.3 V to 7 V
Instantaneous input voltage range, V_I ($t \leq 20$ ns) . . . . . −2 V to 7 V
V_{CC} or GND current . . . . . ±175 mA to 175 mA
Power dissipation at 25°C free-air temperature (see Note 2) . . . . . 1000 mW
Operating free-air temperature, T_A . . . . . −65°C to 135°C
Storage temperature range . . . . . −65°C to 150°C

NOTES: 1. All voltage values are with respect to GND terminal.
2. For operation above 25°C free-air temperature, derate to 170 mW at 135°C at the rate of 8.0 mW/°C.

recommended operating conditions

PARAMETER			EP610I		EP610C		UNIT
			MIN	MAX	MIN	MAX	
V_{CC}	Supply voltage		4.5	5.5	4.75	5.25	V
V_I	Input voltage		0	V_{CC}	0	V_{CC}	V
V_{IH}	High-level input voltage		2	$V_{CC}+0.3$	2	$V_{CC}+0.3$	V
V_{IL}	Low-level input voltage (see Note 3)		−0.3	0.8	−0.3	0.8	V
V_O	Output voltage		0	V_{CC}	0	V_{CC}	V
t_r	Rise time	CLK input		100		250	ns
		Other inputs		250		500	
t_f	Fall time	CLK input		100		250	ns
		Other inputs		250		500	
T_A	Operating free-air temperature		−40	85	0	70	°C

Note 3: The algebraic convention, in which the more negative value is designated minimum, is used in this data sheet for logic voltage levels and temperature only

electrical charateristics over recommended operating free-air temperature range (unless otherwise noted)

PARAMETER			TEST CONDITIONS		EP610I			EP610C			UNIT
					MIN	TYP†	MAX	MIN	TYP†	MAX	
V_{OH}	High-level output voltage	TTL	I_{OH} = −4 mA		2.4			2.4			V
		CMOS	I_{OH} = −2 mA		3.84			3.84			
V_{OL}	Low-level output voltage		I_{OL} = 4 mA				0.45			0.45	V
I_I	Input current		V_I = V_{CC} or GND				±10			±10	µA
I_{OZ}	Off-state output current		V_O = V_{CC} or GND				±10			±10	µA
I_{CC}	Supply current	Standby	V_I = V_{CC} or GND, No load			0.02	0.15		0.02	0.1	mA
		Non-turbo		See Note 4		3	15		3	10	
		Turbo		See Note 5		32	75		32	60	
C_I	Input capacitance		V_I = 0, f = 1 MHz, T_A = 25°C				20			20	pF
C_O	Output capacitance		V_O = 0, f = 1 MHz, T_A = 25°C				20			20	pF
C_{CK}	Clock capacitance		V_I = 0, f = 1 MHz, T_A = 25°C				20			20	pF

† All typical values are at V_{CC} = 5 V, T_A = 25°C

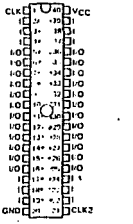
NOTES: 4. When in the non-turbo mode, the device automatically goes into the standby mode approximately 100 ns after the last transition.
5. These parameters are measured with device programmed as a 16-bit counter and f = 1 MHz.

EP910
HIGH-PERFORMANCE 24-MACROCELL ERASABLE PROGRAMMABLE LOGIC DEVICE (EPLD)

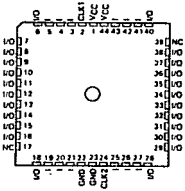
DU187, OCTOBER 1988 – REVISED AUGUST 1990

- High-Density (Over 900 Gates) Replacement for TTL and 74HC
- Virtually Zero Standby Power . . . Typ 20 µA
- High Speed:
  Propagation Delay Time . . . 30 ns
- Asynchronous Clocking of All Registers or Banked Register Operation from 2 Synchronous Clocks
- 24 Macrocells with Configurable I/O Architecture Allowing for Up to 38 Inputs and 24 Outputs
- Each Output Macrocell User-Programmable for D, T, SR, or JK Flip-Flops with Individual Clear Control or Combinational Operation
- UV-Light-Erasable Cell Technology Allows for:
  Reconfigurable Logic
  Reprogrammable Cells
  Full Factory Testing for 100% Programming Yields
- Programmable Design Security Bit Prevents Copying of Logic Stored in Device
- Advanced Software Support Featuring Schematic Capture, Interactive Netlist, Boolean Equations, and State Machine Design Entry
- Package Options Include Plastic [For One-Time-Programmable (OTP) Devices] and Ceramic Dual-In-Line Packages and J-Leaded Chip Carriers

DUAL-IN-LINE PACKAGE
(TOP VIEW)



CHIP-CARRIER PACKAGE
(TOP VIEW)



NC – No internal connection

AVAILABLE OPTIONS

T_A RANGE	SPEED CLASS	PACKAGE TYPE			
		CERAMIC DUAL-IN-LINE PACKAGE (CDIP)	CERAMIC CHIP CARRIER (CLCC)	PLASTIC† DUAL-IN-LINE PACKAGE (PDIP)	PLASTIC† CHIP CARRIER (PLCC)
0°C – 70°C	30 ns	EP910DC-30	EP910JC-30	EP910PC-30	EP910LC-30
	35 ns	EP910DC-35	EP910JC-35	EP910PC-35	EP910LC-35
	40 ns	EP910DC-40	EP910JC-40	EP910PC-40	EP910LC-40
–40°C – 85°C	45 ns	EP910DI-45	EP910JI-45	EP910PI-45	EP910LI-45

†This package is for One-Time Programmable (OTP) devices.

TEXAS INSTRUMENTS
POST OFFICE BOX 655012 • DALLAS, TEXAS 75265

description

general

The Texas Instruments EP910 Erasable Programmable Logic Device is capable of implementing over 900 equivalent gates of SSI and MSI logic functions accommodating up to 36 inputs and 24 outputs all in plastic and ceramic space-saving 40-pin, 600-mil dual-in-line (DIP) packages and 44-pin chip-carrier packages.

Each of the 24 macrocells contains a programmable-AND, fixed-OR PLA structure that yields 8 product terms for logic implementation and a single product term for output-enable and asynchronous-clear control functions. The architecture of the output logic macrocell allows the EP910 user to program output and feedback paths for both combinational or registered operation, active high or active low.

For increased flexibility, the EP910 also includes programmable registers. Each of the 24 internal registers may be programmed to be a D, T, SR, or JK flip-flop. In addition, each register may be clocked asynchronously on an individual basis or synchronously on a banked register basis.

In addition to density and flexibility, the performance characteristics allow the EP910 to be used in the widest possible range of applications. The CMOS EPROM technology reduces the power consumption to less than 20% of equivalent bipolar devices without sacrificing speed performance. Another advantage is 100% generic testing. The device can be erased with ultraviolet (UV) light. Design changes are no longer costly, nor is there a need for post-programming testing.

Programming the EP910 is accomplished by using the TI EPLD Development System, which supports four different design entry methods. When the design has been entered, the software performs automatic translation into logical equations, Boolean minimization, and design fitting directly into an EP910. The device may then be programmed to achieve customized working silicon within minutes at the designer's own desktop.

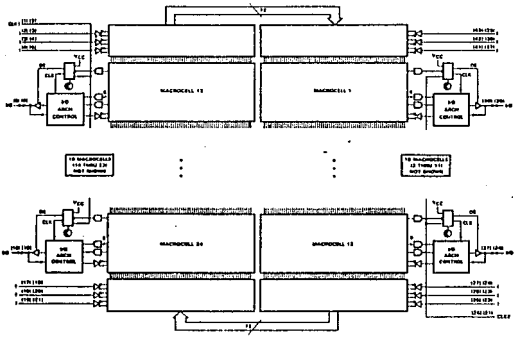
functional

The EP910 is an Erasable Programmable Logic Device (EPLD) that uses a CMOS EPROM technology to implement logic designs in a programmable AND-logic array. The device also contains a revolutionary programmable I/O architecture that provides advanced functional capability for user programmable logic.

Externally, the EP910 provides 12 dedicated data inputs and 24 I/O pins, which may be configured for input, output, or bidirectional operation. Figure 1 shows the EP910 basic macrocell. The internal architecture is organized with the familiar sum-of-products (AND-OR) structure. Inputs to the programmable AND array come from the true and complement signal from 12 dedicated data inputs and 24 feedback signals originating from each of the 24 I/O architectural control blocks. The 72-input AND array encompasses 240 product terms, which are distributed among 24 available macrocells. Each EP910 product term represents a 72-input AND gate.

At the intersection point between each AND array input and each product term, there is an EPROM control cell. In the erased state, all connections are made. This means both the true and complement of all inputs are connected to each product term. Connections are opened during the programming process. Therefore, any product term may be connected to the true or complement form of any array input signal.

functional block diagram

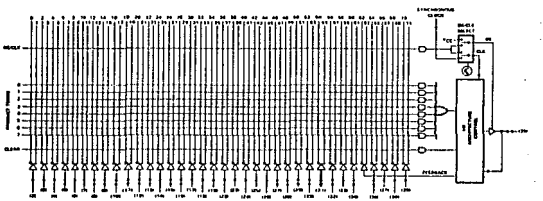


Pin numbers in parenthesis are for DIP packages. Pin numbers in brackets are for chip-carrier packages.

When both the true and complement of an array input signal are connected, a logical false results on the output of the AND gate. When both the true and complement forms of any array input signal are programmed open, then a logical "don't care" results for that input. If all 72 inputs for a given product term are programmed open, then a logical true state results on the output of the corresponding AND gate. Two dedicated clock inputs (not available in the AND array) provide the clock signals used for asynchronous clocking of the EP910 internal registers. Each of these clock signals is positive-edge triggered and has control over a bank of 12 registers. CLK1 controls registers associated with macrocells 13 through 24. CLK2 controls registers associated with macrocells 1 through 12. The EP910 advanced I/O architecture allows the number of synchronous registers to be user defined, from 1 to 24. Both dedicated clock inputs are positive-edge triggered.

I/O architecture

The EP910 input/output architecture provides each macrocell with over 50 possible I/O configurations. Each I/O can be configured for combinational or registered output, with programmable output polarity. Four different types of registers (D, T, JK, and SR) can be implemented into every I/O without any additional logic requirements. I/O feedback selection can also be programmed for registered or input (pin) feedback. Another benefit of the EP910 I/O architecture is its ability to individually clock each internal register from asynchronous clock signals.



Pin numbers shown are for dual-in-line packages

FIGURE 1. LOGIC ARRAY MACROCELL (MACROCELL 1 ILLUSTRATED)

OE/CLK selection

Figure 2 shows the two modes of operation that are provided by the OE/CLK select multiplexer. The operation of this multiplexer is controlled by a single EPROM bit and may be individually configured for each of the EP910 I/O pins. In Mode 0, the 3-state output buffer is controlled by the OE/CLK product term. If the output of the AND gate is true, the output buffer is enabled. If the output of the AND gate is false, the output buffer is in the high-impedance state. In this mode, the macrocell flip-flop is clocked by its respective synchronous clock input signal (CLK1 or CLK2). After erasure, the OE/CLK select multiplexer is configured in Mode 0.

In Mode 1, the output-enable buffer is always enabled. The macrocell flip-flop may now be triggered from an asynchronous clock signal generated by the OE/CLK multiplexable product term. This mode allows individual clocking of flip-flops from any of the 72 available AND array input signals. Because both true and complement signals reside in the AND array, the flip-flop may be configured for positive- or negative-edge-triggered operation. With the clock now controlled by a product term, gated clock structures are also possible.

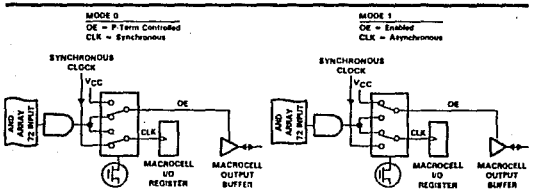


FIGURE 2. OE/CLK SELECT MULTIPLEXER

output/feedback selection

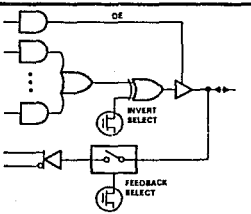
Figure 3 shows the EP910 basic output configurations. Along with combinational output, 4 register types are available. Each macrocell I/O may be independently configured. All registers have individual asynchronous-clear control from a dedicated product term. When the product term is asserted, the macrocell register will immediately be loaded with a zero independently of the clock. On power-up, the EP910 performs the clear function automatically.

In the combinational configuration, 8 product terms are ORed together to acquire the output signal. The invert-select EPROM bit controls output polarity and the output-enable buffer is product term controlled. The feedback-select multiplexer enables registered I/O (pin), feedback, or no feedback to the AND array.

When the D or T register is selected, 8 product terms are ORed together and made available to the register input. The invert select EPROM bit determines output polarity. The OE/CLK select multiplexer is used to configure the mode of operation to Mode 0 or Mode 1 (see Figure 2). The feedback-select multiplexer enables registered I/O (pin) or no feedback to the AND array.

If the JK or SR registers are selected, the 8 product terms are shared among two OR gates whose outputs feed the two primary register inputs. The allocation of product terms for each register input is optimized by the TI EPLD Development System. The invert select EPROM bit controls output polarity while the OE/CLK select multiplexer allows the mode of operation be Mode 0 or Mode 1. The feedback-select multiplexer enables registered I/O (pin) or no feedback to the AND array.

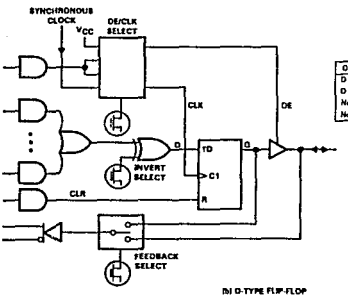
Any I/O pin may be configured as a dedicated input by selecting no output and pin feedback. No output is obtained by disabling the macrocell output buffer. In the erased state, I/O is configured for combinational active-low output with input (pin) feedback.



I/O SELECTION

OUTPUT/POLARITY	FEEDBACK
Combinational/high	Pin, None
Combinational/low	Pin, None
None	Pin

(a) COMBINATIONAL



I/O SELECTION

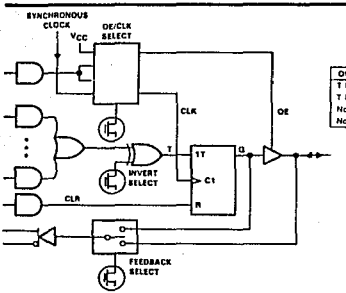
OUTPUT/POLARITY	FEEDBACK
D Register/High	D Register, Pin, None
D Register/low	D Register, Pin, None
None	D Register
None	Pin

FUNCTION TABLE

INPUTS			OUTPUT
CLR	CLK	D	Q
L	↑	L	L
L	↑	H	H
L	L or H	X	Q_0
H	X	X	L

(b) D-TYPE FLIP-FLOP

FIGURE 3. I/O CONFIGURATIONS



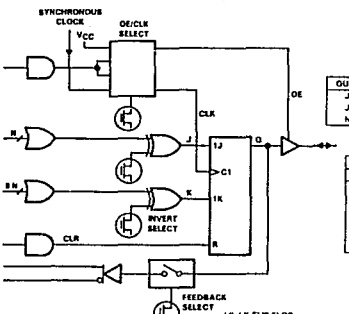
I/O SELECTION

OUTPUT/POLARITY	FEEDBACK
T Register/high	T Register, Pin, None
T Register/low	T Register, Pin, None
None	T Register
None	Pin

FUNCTION TABLE

INPUTS			OUTPUT
CLR	CLK	T	Q
L	↑	L	Q_0
L	↑	H	$\overline{Q}_0$
L	L or H	X	Q_0
H	X	X	L

(c) TOGGLE FLIP-FLOP



I/O SELECTION

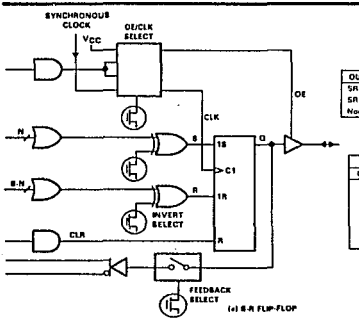
OUTPUT/POLARITY	FEEDBACK
JK Register/high	JK Register, None
JK Register/low	JK Register, None
None	JK Register

FUNCTION TABLE

INPUTS				OUTPUT
CLR	CLK	J	K	Q
L	↑	L	L	Q_0
L	↑	L	H	L
L	↑	H	L	H
L	↑	H	H	$\overline{Q}_0$
L	L or H	X	X	Q_0
H	X	X	X	L

(d) J-K FLIP-FLOP

FIGURE 3. I/O CONFIGURATIONS (CONTINUED)



I/O SELECTION

OUTPUT/POLARITY	FEEDBACK
SR Register/high	SR Register, None
SR Register/low	SR Register, None
None	SR Register

FUNCTION TABLE

INPUTS				OUTPUT
CLR	CLK	S	R	Q
L	↑	L	L	Q_0
L	↑	L	H	L
L	↑	H	L	H
L	↑	H	H	Undefined
L	L or H	X	X	Q_0
H	X	X	X	L

FIGURE 3. I/O CONFIGURATIONS (CONTINUED)

absolute maximum ratings over operating free-air temperature range (unless otherwise noted)

Supply voltage range, V_{CC} (see Note 1) . . . . . −0.3 V to 7 V
Instantaneous supply voltage range, V_{CC} ($t \leq 20$ ns) . . . . . −2 V to 7 V
Programming supply voltage range, V_{PP} . . . . . −0.3 V to 13.5 V
Instantaneous programming supply voltage range, V_{PP} ($t \leq 20$ ns) . . . . . −2 V to 13.5 V
Input voltage range, V_I . . . . . −0.3 V to 7 V
Instantaneous input voltage range, V_I ($t \leq 20$ ns) . . . . . −2 V to 7 V
V_{CC} or GND current . . . . . −250 mA to 250 mA
Power dissipation at 25°C free-air temperature (see Note 2) . . . . . 1200 mW
Operating free-air temperature, T_A . . . . . −65°C to 135°C
Storage temperature range . . . . . −65°C to 150°C

NOTES: 1. All voltage values are with respect to GND terminal.
2. For operation above 25°C free-air temperature, derate to 144 mW at 135°C at the rate of 9.6 mW/°C.

recommended operating conditions

PARAMETER			EP910I		EP910C		UNIT
			MIN	MAX	MIN	MAX	
V_{CC}	Supply voltage		4.5	5.5	4.75	5.25	V
V_I	Input voltage		0	V_{CC}	0	V_{CC}	V
V_{IH}	High-level input voltage		2	$V_{CC}+0.3$	2	$V_{CC}+0.3$	V
V_{IL}	Low-level input voltage (see Note 3)		−0.3	0.8	−0.3	0.8	V
V_O	Output voltage		0	V_{CC}	0	V_{CC}	V
t_r	Rise time	CLK input		50		100	ns
		Other inputs		50		100	
t_f	Fall time	CLK input		50		100	ns
		Other inputs		50		100	
T_A	Operating free-air temperature		−40	85	0	70	°C

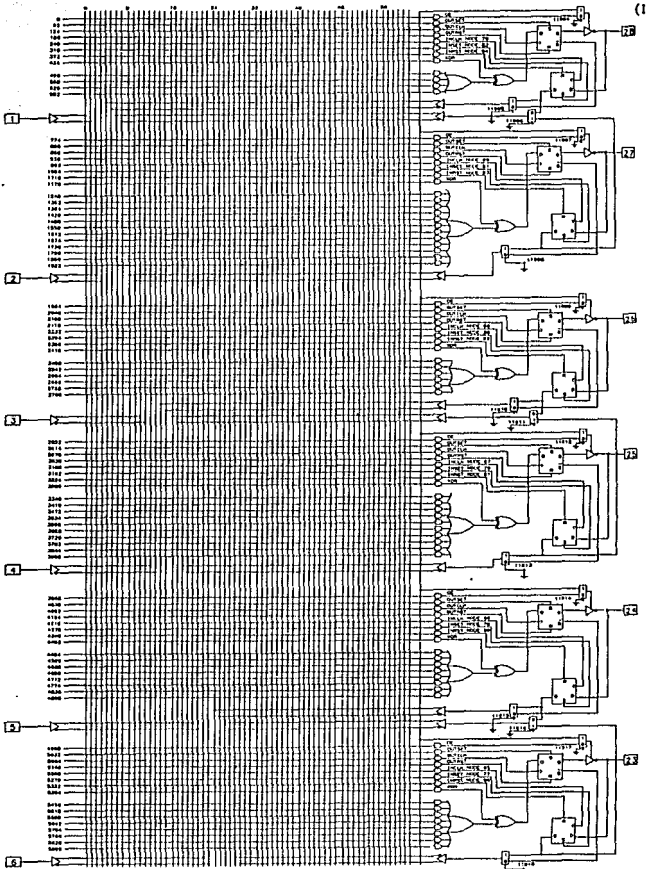
Note 3: The algebraic convention, in which the more negative value is designated minimum, is used in this data sheet for logic voltage levels and temperature only.

electrical characteristics over recommended operating free-air temperature range (unless otherwise noted)

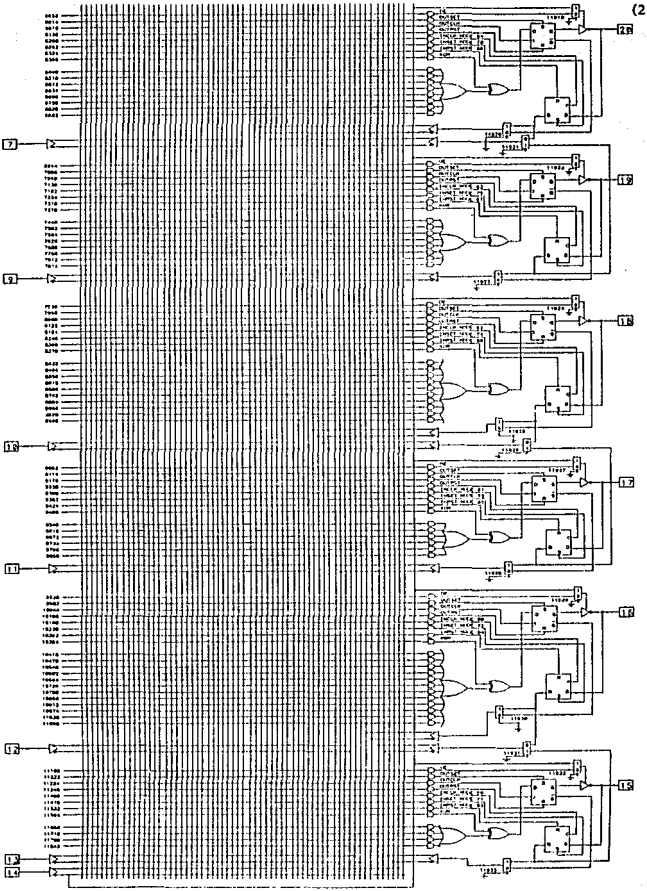
PARAMETER			TEST CONDITIONS		EP910I			EP910C			UNIT
					MIN	TYP†	MAX	MIN	TYP†	MAX	
V_{OH}	High-level output voltage	TTL	I_{OH} = −4 mA		2.4			2.4			V
		CMOS	I_{OH} = −2 mA		3.84			3.84			
V_{OL}	Low-level output voltage		I_{OL} = 4 mA				0.45			0.45	V
I_I	Input current		V_I = V_{CC} or GND				±10			±10	µA
I_{OZ}	Off-state output current		V_O = V_{CC} or GND				±10			±10	µA
I_{CC}	Supply current	Standby	V_I = V_{CC} or GND, No load	See Note 4		0.02	0.15		0.02	0.1	mA
		Non-turbo		See Note 5		6	30		8	20	
		Turbo		See Note 5		45	100		45	80	
C_I	Input capacitance		V_I = 0, f = 1 MHz,	T_A = 25°C			20			20	pF
C_O	Output capacitance		V_O = 0, f = 1 MHz,	T_A = 25°C			20			20	pF
C_{clk}‡	Clock capacitance		V_I = 0, f = 1 MHz,	T_A = 25°C			20			20	pF

† All typical values are at V_{CC} = 5 V, T_A = 25°C.
‡ During programming, the clock capacitance of CLK2 is 60 pF maximum.
NOTES: 4. When in the non-turbo mode, the device automatically goes into the standby mode approximately 100 ns after the last transition.
5. These parameters are measured with device programmed as a 16-bit counter, and f = 1 MHz.

P331
(1 of 2)



P331
(2 of 2)



APENDICE C

COTIZACIONES

PLD'S

DESCRIPCION	P.U. (DOLARES)
AMPAL23S8-20DC (17 PS MIN)	$ 15.97
AMPAL23C8-20PL	$ 9.37
PALCE 26V12H20PC	$ 10.56
PALCE 22V10Q25PC	$ 8.12
EPM5016DC-1	$ 17.81
EPM5032DC-2	$ 36.72
GAL16V8Q25LNC	$ 1.62
GAL20V8QC25LNC	$ 2.56
GAL22V10-20LD	$ 39.22
GAL22V10-25LP	$ 6.65
GAL6001-30P	$ 11.25
CY7C330-50WC	$ 25.86
CY7C331-20WC	$ 25.18
CY7332-20WX	$ 2568.00
PLS105AF	$ 17.18
PLS159AN	$ 4.46
PLS168AN	$ 9.37
PLS179AN	$ 6.65
EP310DC-3	$ 10.16
EP610DC-30	$ 10.93
EP910DC-30	$ 23.43

DISPOSITIVOS TTL

DESCRIPCION	P.U.(PESOS)
CIRCUITO INTEGRADO 74LS367	$ 1769.00
CIRCUITO INTEGRADO 7433	$ 1975.00
CIRCUITO INTEGRADO 74S86	$ 1428.00
CIRCUITO INTEGRADO 74LS157	$ 1467.00
CIRCUITO INTEGRADO 74148	$ 3905.00
CIRCUITO INTEGRADO 74LS112	$ 1312.00
CIRCUITO INTEGRADO 74LS02	$ 923.00
CIRCUITO INTEGRADO 74LS04	$ 923.00
CIRCUITO INTEGRADO 74LS00	$ 923.00
CIRCUITO INTEGRADO 74LS08	$ 923.00
CIRCUITO INTEGRADO 74LS21	$ 1157.00
CIRCUITO INTEGRADO 74LS27	$ 923.00
CIRCUITO INTEGRADO 74LS42	$ 1770.00
CIRCUITO INTEGRADO 7474	$ 1600.00
CIRCUITO INTEGRADO 74LS82	$ 1925.00
CIRCUITO INTEGRADO 74LS93	$ 1882.00
CIRCUITO INTEGRADO 74LS158	$ 1467.00
CIRCUITO INTEGRADO 74172N	$ 50000.00
CIRCUITO INTEGRADO 74LS181	$ 8460.00
CIRCUITO INTEGRADO 74LS153	$ 1467.00
CIRCUITO INTEGRADO 74LS257	$ 1538.00
CIRCUITO INTEGRADO 74LS283	$ 1929.00
CIRCUITO INTEGRADO 74LS298N	$ 7900.00
CIRCUITO INTEGRADO 74LS240	$ 2002.00
CIRCUITO INTEGRADO 74LS253	$ 1537.00
CIRCUITO INTEGRADO 74LS365	$ 1770.00

CIRCUITO INTEGRADO 74366	$ 1740.00
CIRCUITO INTEGRADO 74367	$ 1085.00
CIRCUITO INTEGRADO 74LS390	$ 2235.00
CIRCUITO INTEGRADO 74LS393	$ 1703.00
CIRCUITO INTEGRADO 74LS151	$ 1467.00
CIRCUITO INTEGRADO 74S241N	$ 23150.00
CIRCUITO INTEGRADO 74S11	$ 5100.00
CIRCUITO INTEGRADO 74LS32	$ 1156.00
CIRCUITO INTEGRADO 74S10	$ 1225.00
CIRCUITO INTEGRADO 74S04	$ 1400.00
CIRCUITO INTEGRADO 74S08	$ 5100.00
CIRCUITO INTEGRADO 74LS16	$ 1600.00
CIRCUITO INTEGRADO 7414	$ 1647.00
CIRCUITO INTEGRADO 74S138	$ 4447.00
CIRCUITO INTEGRADO 74S175	$ 14100.00
CIRCUITO INTEGRADO 74LS279	$ 1537.00

APENDICE D

ASIGNACION DE VARIABLES
PARA CAPTURA ESQUEMATICA

ORIGINAL	DASH	ORIGINAL	DASH
E1EXT	S1	I1ENTER	JE15
SL3ENA	S2	EXREQ	JE3
FLTRESTRT/	S3	LSTCYC	S36
INTREQ/	S4	I1CYC	JE12
EXTIDX	S5	BRREG	S23
LSFCYC	S6	MCLK/	S24
STORE	S7	CPHLD/	PA5
CLASS2	S8	CLASS2/	S25
EXTIDX/	S9	EXHLD	S26
ALUIMM/	S10	E2CYC	S34
CLASS3	S11	CYCHLD/	PA10
RMST1/	S12	CPRUN	PA24
IR02/	S13	FENTER/	JE14
BRENA/	JF4	FENTER	S29
LOAD	S14	I1CYC/	S31
CLASS3/	S15	I2CYC	AUX2
ALURR/	S16	E1CYC	S33
CLASS0	S17	FCYC/	JF7
IOACK/	JE7	XCYC	AUX1
CLASS1	S18	XCYC/	S38
BRPEND/	S19	FCYC	S39
RESTART	S20	BRRLCYC	S40
POFF/	JD17	BRRCLK	S41
CPCONN/	PA6	PIECLK	S42
CPSGLCYC	S21	CYCCLK1	S43
HALT	S22	CYCCLK2	S44
CPSTOP/	PA26	CYCENA	JE4
E2ENTER	JE11	CYCENA/	S45
XENTER	JE10	RUN	JE8
E1ENTER	JE9	BRUN	PA25

ORIGINAL	DASH	ORIGINAL	DASH
IR00/	S46	DEXENA/	JD5
IR01/	S47	RBSEL/	S77
IR01	JC7	REGBB/	JC23
IR00	JC6	STRBDATC/	S58
IR02	JC8	LDBI/	S59
IR03/	S48	IR09/	S60
IR03	JC9	LDBDATC/	S61
IR10	JC13	RTSH/	S62
IR09	JC10	MOVBYT/	S63
IR08	JC2	STAREG	S64
IR07	JC16	MREGSTA/	S65
CPIRDIS/	PA10	RDSWS/	PA2
REGDIS/	S49	INTINH/	S66
IR14	JC3	STALCLK/	JD7
SRSTD/	S50	STAREG/	S67
IR11	JC12	CPPRINC/	PA9
IR12	JC5	CPIRLD/	PA19
IR13	JC4	CPPRCLK/	PA13
BBD	JD13	REGDIS/	S49
BMASEL	JD19	BRENA	S68
ERLDENA	S51	RSELA	JC24
RTCIACK/	PA28	CPPRLD/	PA11
BIT	S52	CPREGLD/	PA21
CPIACK/	PA1	INCPR/	S69
IORST/	S52	LDER/	JD10
LSDATC/	S54	LDIR/	S70
BMBSEL	JD14	PMFCLK/	S71
BPGADRC/	S55	PRCLK/	S72
WORD/	S56	CYCSTRT/	PA20
EXTENDED/	S57	WRTCB/	JC19

ORIGINAL	DASH	ORIGINAL	DASH
KHLT1	S73	CB09	PB27
CBER	JD9	CB10	PB29
LDPR	S74	CB11	PB26
WRTSB/	JC20	CB12	PB18
LDREG/	JC21	INT2	S78
EXDATAFLT/	S75	CB13	PB22
CPREGDIS	S76	INT1	S79
CPSTACB/	PA16	CB14	PB21
CPCBINH/	PA14	INT0	S80
SMHB	JD25	CB15	PB24
SMHA	JD23	PM15	S81
SMLB	JD22	PM14	S82
SNLA	JD24	PM13	S83
CIN/	JD11	PB00	VCC
FSEL	JD12	PB01	PB47A
MODE	JC17	PB02	PB34A
REGAB/	JC22	PB03	PB37A
SUMSEL	JD18	PB04	PB33A
RSELB	JC25	PB05	PB35A
ASEL	JD20	PB06	PB39A
SMCB	JD15	PB07	PB36A
STACB/	JP3	PB08	PB42A
CB01	PB25	PB09	PB40A
CB02	PB31	PB10	PB38A
CB03	PB28	PB11	PB41A
CB04	PB23	PB12	PB48A
CB05	PB19	PB13	PB44A
CB06	PB17	PB14	PB46A
CB07	PB20	PB15	PB43A
CB08	PB32	IB00	PB4

ORIGINAL	DASH	ORIGINAL	DASH
IB01	PB9	LDCOND	JF6
IB02	PB13	CCSV	S89
IB03	PB11	SCSV	JF5
IB04	PB8	ZCSV	S90
IB05	PB3	BSEL/	PA35
IB06	PB2	SRST/	PA36
IB07	PB15	LEXFAULT	AUX109
IB08	PB12	LSTALL	AUX110
IB09	PB5	LPMF	AUX112
IB10	PB14	NEXMINT	AUX108
IB11	PB16	BSELD/	S95
IB12	PB1	NEXPINT	AUX107
IB13	PB7	MPEINT	AUX105
IB14	PB6	INTENA	-
IB15	PB10	DMPERR/	PA29
IR04	JC11	MPECLR/	S99
IR05	JC14	NEXPER/	S100
IR06	JC15	NEXPCLR/	S101
IR15	JC1	NEXMEM/	S102
CB00	PB30	NEXMCLR/	S103
MRST/	PA8	BATTN/	S104
ALU15/	JD8	EXFAULT/	S105
COUT/	JC18	STALL/	JD16
RTCREQF	S84	PMERR	PA15
IOREQF	JF2	RESTRTCLR/	S106
CPREQF	S86	CPCLR/	S107
CBZERO	JF1	IOCLR/	S108
Z	S87	RTCCLR/	S109
S	S88	KHLT2	S119
C	JD21	MPEINT/	S117

ORIGINAL	DASH	ORIGINAL	DASH
NEXMINT/	S118	DRDY	JE00
NEXPINT/	S110	KHLT3	S130
FAULT/	PA7	PBREQ	S131
PMFLT/	JE6	CPHPRI/	PA23
RESTART/	JE5	PROT/	PA22
FAULTCLR/	JD6	ISELD	S132
INTSYSENA/	S111	BAD	S133
LEV567ENA	JE2	WRTHD/	AUX133
DISINT/	S112	WRTLD/	AUX134
EXTIMOUT/	JD2	MEM	PA42
EXIDLE/	S113	WRTLB	PA40
CPINT/	S114	ISEL/	PA38
IOINT/	S115	BA/	PA37
RTCINT/	S116	WRTHB	PA39
PFINT/	JD4	BB/	PA41
TIMOUT/	S120		
TIMOUT	S121		
EXTOCLR	JD1		
BI1CYC	JC26		
DRDY/	AUX122		
BATTN	AUX123		
BBR	S122		
BREQR	S123		
MCLK1	S124		
SACKR	S125		
ACKR	S126		
PER	JD3		
RTCIREQ	S127		
IREQR	S128		
CPIREQ	S129		

BIBLIOGRAFIA

1. The TTL Data Book for Design Engineers
Texas Instruments

2. Logic Data Book Volumen 2
Texas Instruments

3. Logica Digital y Diseño de Computadoras
M. Morris Mano
Prentice Hispanoamericana, S.A.

4. Circuitos Integrados
Robert G. Hibberd
Marcombo

5. Programmable Logic Data Book
Texas Instruments
1990

6. Programmable Logic Devices Data Book and Design Guide
National Semiconductor

7. Programmable Logic Designer's Guide
Roger C. Alford
Macmillan Publishing

8. Manual, Schematic Designer Volumen I, II, III
Future Net
1989

9. Manual, Gates 5.0
Data I/O
1989

10. Manual, Unisite
Data I/O
1989

11. Manual, PLDLinx 1.0
Data I/O
1989

12. Manual, Logic Diagram Package
Data I/O
1989

13. Manual, orCAD
Systems Corporation
1986