MANUAL DE CONSULTA

FACULTAD DE INGENIERIA

ENTORNO DE DISEÑO MAX+PLUS II

FACULTAD DE INGENIERÍA
DIVISIÓN DE INGENIERÍA ELÉCTRICA
DEPARTAMENTO DE INGENIERÍA EN COMPUTACIÓN
E INGENIERÍA EN ELECTRÓNICA

FACULTAD

ING. NORMA ELVA CHÁVEZ RODRÍGUEZ
ING. JORGE VALERIANO ASSEM
M.I. ARTURO HARO RUIZ

CIUDAD UNIVERSITARIA, FEBRERO 2001

ÍNDICE

TEMA	PÁGINA
1 Introducción	*3*
2 Fundamentos de Max+Plus II	*4*
2 1 Las once aplicaciones de Max+Plus II	*5*
2 2 Ejemplo de aplicaciones simultaneas en Max+Plus II	*6*
3 Arquitectura de los dispositivos de Altera	*7*
3 1 Chip Max EPM7128SLC84-15	*7*
3 2 Chip Flex 10K20RC240-4	*8*
4 Tipos de archivos de diseño	*8*
4 1 Proyectos	*9*
4 2 Archivos y ventanas de ayuda	*9*
5 Diseñando en Max+Plus II con el editor gráfico	*12*
5 1 Graphic Editor (editor gráfico)	*13*
5 2 Compilando un diseño	*16*
5 3 Archivo de reporte	*19*
5 4 Simulando un proyecto	*20*
5 5 Convirtiendo nuestro diseño en un bloque funcional	*24*
5,6 Programando el dispositivo	*25*
6 Diseñando en Max+Plus II con el editor de texto	*26*
7 Tarjeta de desarrollo UP1	*28*
7 1 Descripción de la tarjeta UP1	*28*
7 2 Uso del dispositivo EMP7128S	*32*
7 3 Uso del dispositivo EPF10K20	*35*
8 Tarjeta de desarrollo Upx	*39*
9 BIBLIOGRAFÍA	*44*

FACULTAD DE INGENIERIA

APUNTE
60-B

FACULTAD DE INGENIERIA UNAM.

612500

2001
G.- 612500

1. INTRODUCCIÓN.

Los entornos EDA (Electronic Design Automation) son las herramientas de trabajo más comúnmente empleadas en el diseño e implantación de circuitos lógicos Permiten una gran versatilidad al momento de simular cualquier posible situación en el funcionamiento de los diseños efectuados por los ingenieros El entorno EDA es sencillo de manejar gracias a las interfaces gráficas que proporcionan las plataformas de diseño como ALTERA, OrCAD, XILINX, que permiten la instrumentación de dichos diseños y su implantación en los CPLD's y FPGA's

Una de estas herramientas (de la compañía Altera) es puesta al alcance del alumno en esta obra con el objetivo de dar a conocer las últimas herramientas de diseño Para lo cual a lo largo de diferentes temas, el alumno adquirirá los conocimientos que le permitan utilizar los dispositivos lógicos programables complejos fabricados por ALTERA, junto con el software Max+Plus II como principal herramienta de diseño, verificación, simulación y programación de los proyectos y practicas del Laboratorio de Dispositivos Lógicos Programables

Para la implantación de los diseños se dispone de una tarjeta de desarrollo llamada UP1, que contiene dos CPLD's Esta tarjeta va a permitir una fácil implantación y verificación de diversos diseños digitales, mediante una serie de visualizadores, pulsadores, microinterruptores y otros elementos útiles para realizar simulaciones físicas en la tarjeta de desarrollo

G.- 612500

2. FUNDAMENTOS DE MAX+PLUS II

El software MAX+PLUS II (Multiple Array Matrix Programmable Logic User System II) suministra una plataforma múltiple e independiente de una arquitectura en particular, que se adapta fácilmente a las necesidades de cualquier diseño Y por ello ofrece una forma fácil de diseñar proyectos y hacer su programación directa en los dispositivos

Existen herramientas para crear los diseños de una manera jerárquica, poderosa síntesis lógica, compilación, partición, simulación funcional y en tiempo, simulación enlazada con varios dispositivos, análisis de tiempo etc Max+Plus II incluye once programas de aplicación y el ambiente integrado Esto se ilustra en la Fig 1

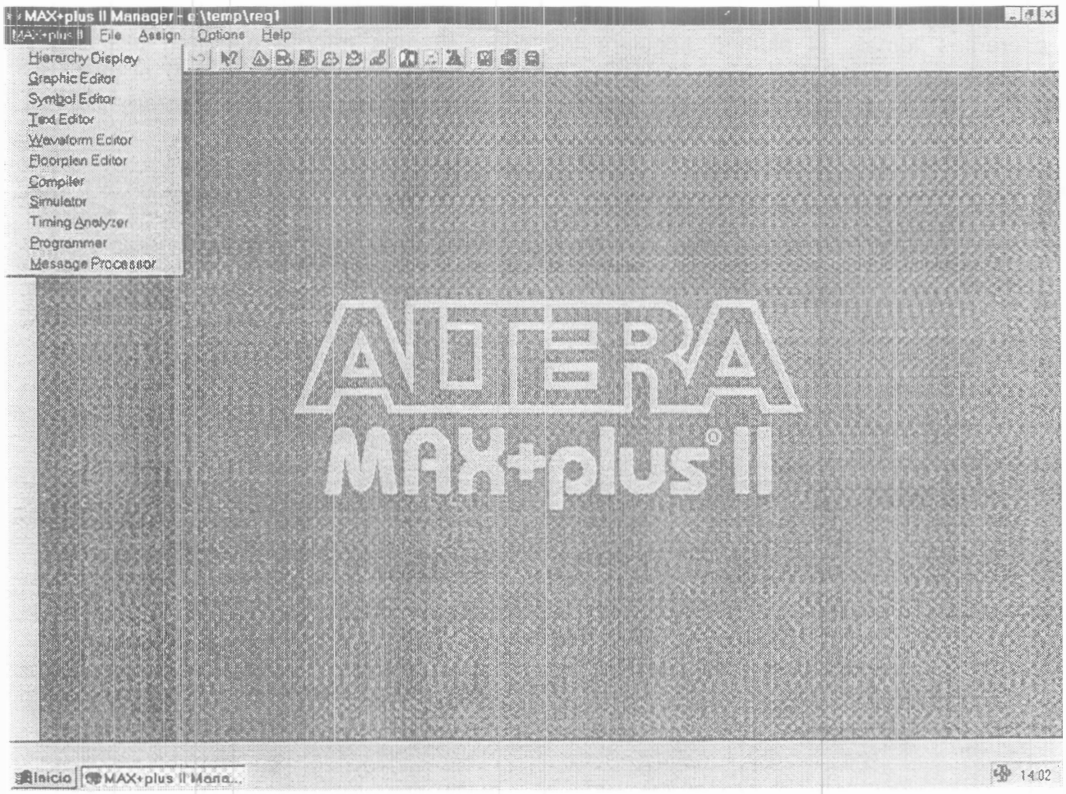


Fig 1 Ambiente Integrado con las once aplicaciones

El software Max+Plus II se accede desde el menú "Inicio" en el ambiente "Windows", el menú de las once aplicaciones se accede haciendo "click" en la barra de menús titulada con el nombre de "Max+Plus II"

Se pueden realizar varios diseños en aplicaciones diferentes al mismo tiempo, también es posible ejecutar simultáneamente el compilador, simulador, analizador en el tiempo y el programador

2.1 Las once aplicaciones de Max+Plus II

Las once aplicaciones incluidas en el software de Max+Plus II son las siguientes y se describen brevemente a continuación

- **Hierarchy Display.** Muestra la distribución actual de la jerarquía del proyecto Constituye un árbol con ramas que representan los subdiseños Se puede identificar el tipo de archivo por medio de su extensión

- **Graphic Editor.** Te permite crear un diseño visual en un ambiente (WYSIWYG) "lo que ves es lo que tienes" Puedes usar los bloques predeterminados por ALTERA, o crear los que se necesiten

- **Symbol Editor.** Puede convertir diseños previos en símbolos nuevos para ser utilizados en la jerarquía

- **Text Editor.** Puede crear y editar diseños de texto basados en los lenguajes AHDL, VHDL y Verilog HDL También permite crear, ver o editar archivos ASCII

- **Waveform Editor.** Tiene dos funciones básicas (i) Como herramienta de diseño, (ii) Como herramienta para crear vectores de prueba y observar los vectores resultantes de la simulación

- **Floorplan Editor.** Es para hacer una asignación manual de pines en el dispositivo en relación al diseño lógico

- **Compiler.** Procesa los proyectos creados, encontrando posibles errores y generando los archivos de programación necesarios, así como la información necesaria para la simulación

- **Simulator.** Permite revisar la operación lógica y funcional dando los diagramas de tiempo de las entradas del circuito lógico para encontrar el diagrama de tiempo de las salidas

- **Timing Analyzer.** Analiza el rendimiento (performance) del circuito lógico, después de haber pasado por el compilador

- **Programmer.** Sirve para programar, verificar, examinar y probar los dispositivos de ALTERA

- **Message Processor:** Muestra mensajes de error, advertencias, e información sobre el estado actual del proyecto

Nota: Un diseño lógico incluyendo sus subdiseños, es llamado "Project" en Max+Plus II

2.2 Ejemplo de aplicaciones simultaneas en Max+Plus II

Muchas de las once aplicaciones de Max+Plus II son independientes y pueden funcionar simultáneamente En la Fig 2 se ilustra un ejemplo, se tienen abiertas las ventanas de jerarquías, el simulador y la ayuda

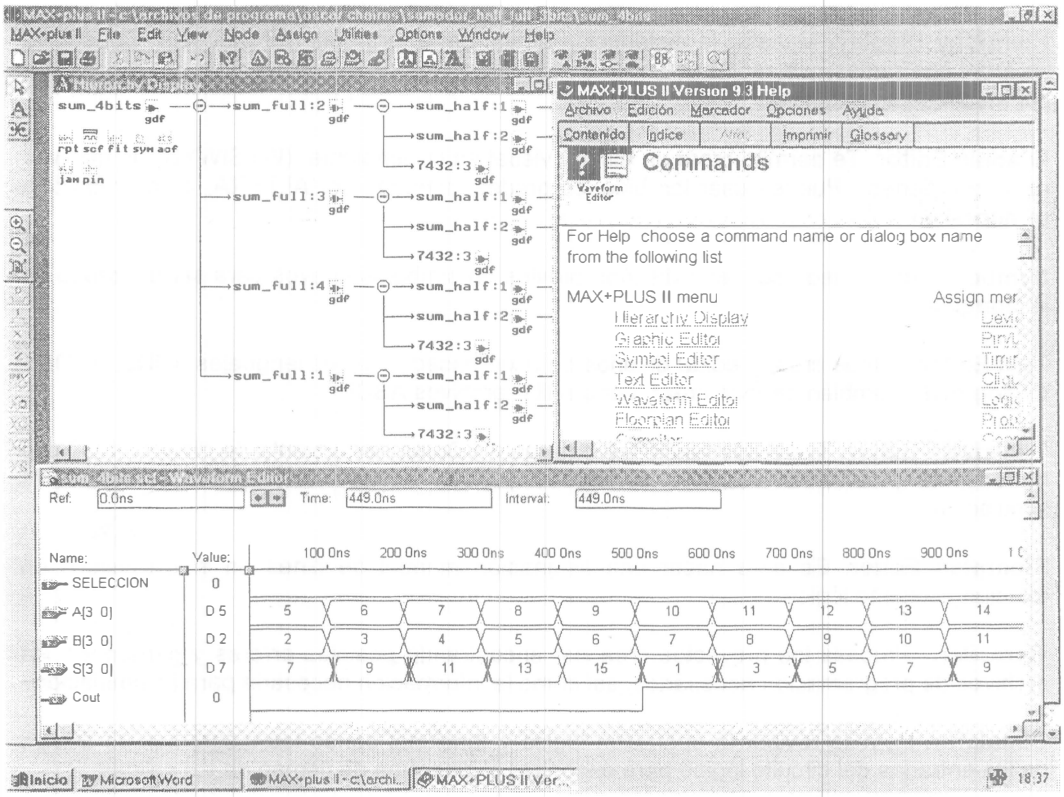


Fig 2 Tres aplicaciones de Max+Plus II trabajando simultáneamente

Muchas comandos tales como apertura de archivos, selección del chip, asignación de pines, celdas lógicas y compilación del proyecto actual están compartidos por varias de las aplicaciones de Max+Plus II, y por lo tanto, aprendiendo el uso de una aplicación nos da una idea de cómo usar las otras

Los editores de diseño que son los que usamos para introducir nuestro diseño (Graphic, Text, Waveform) y sus editores auxiliares (Floorplan y Symbol) comparten muchas características Cada editor de diseño nos permite realizar tareas similares tales como acomodar una señal o símbolo, de manera que fácilmente se pueden combinar diferentes tipos de archivos, y así en un proyecto jerárquico cada subdiseño puede ser hecho en el editor que se ajuste mejor a cada bloque funcional

ALTERA suministra una gran variedad de megafunciones y macrofunciones en las librerías denominadas LPM, gracias a esto, se suministra una gran cantidad de módulos listos para usarse

El compilador es el corazón del sistema Max+Plus II, suministrando un poderoso procesador por medio del cual podemos tomar la mejor posible implementación de cualquier proyecto

También gracias al compilador podemos detectar la localización de errores y mensajes acerca de ciertas precauciones que debemos tomar en cuenta haciendo fáciles y rápidas las posibles modificaciones a nuestro proyecto Se pueden crear archivos de salida, en una variedad de formatos funcionales, de tiempo o ligados a distintos tipos de dispositivos

3. ARQUITECTURA DE LOS DISPOSITIVOS DE ALTERA.

- *Celdas lógicas (logic cell)* Es un bloque básico en cualquier dispositivo
- *Macroceldas (macrocell)* Bloque básico basado en suma de productos (SOP) en cualquier dispositivo Max
- *Elementos lógicos (logic elements)* Bloque básico basado en "look-up table" en cualquier dispositivo Flex
- *Arreglos lógicos(logic array)* Grupo de celdas lógicas

ALTERA maneja siete familias lógicas programables, cuatro que utilizan (SOP)

- Clásica
- Max 5000
- Max 7000 y Max 7000S
- Max 9000

Tres familias están basadas en tablas de "LOOK – UP"

- Flash Logic
- Flex 8000
- Flex 10k

3.1. Chip Max EPM7128SLC84-15. Este chip contiene 84 patas con 128 macroceldas Tiene un arreglo de "AND´s" programables, "OR" fijas y un flip-flop que se puede configurar independientemente Este chip tiene una capacidad aproximada de 2,500 compuertas y es ideal para diseños combinacionales y funciones secuenciales En la Fig 3 se ilustra el diagrama a bloques de este dispositivo

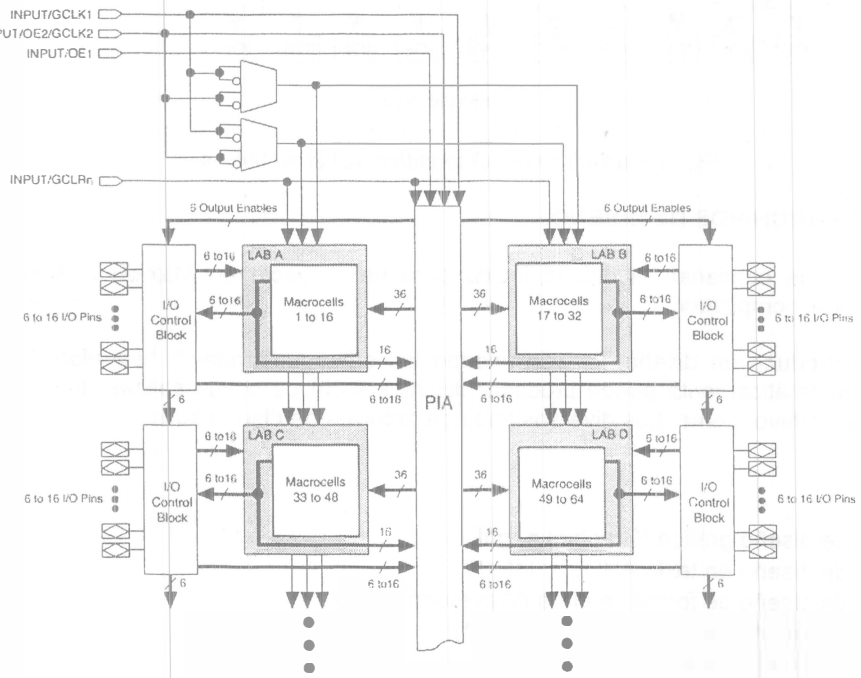


Fig 3 Estructura del dispositivo EPM7128SLC84-15

3.2. Chip Flex 10k20RC240-4. Contiene 240 pines con 1152 elementos lógicos y 6 bloques de RAM, cada elemento lógico consiste de 4 entradas, un flip-flop programable y señales dedicadas a posibles acarreos para poder usar funciones en cascada, cada arreglo de RAM suministra 2,048 bits de memoria que pueden ser usados para crear una RAM, ROM o FIFO Sin embargo los arreglos también pueden ser usados para implementar funciones lógicas como multiplexores, contadores, máquinas de estados y en general para cualquier procesamiento digital En la Fig 4 se ilustra el diagrama a bloques de este dispositivo

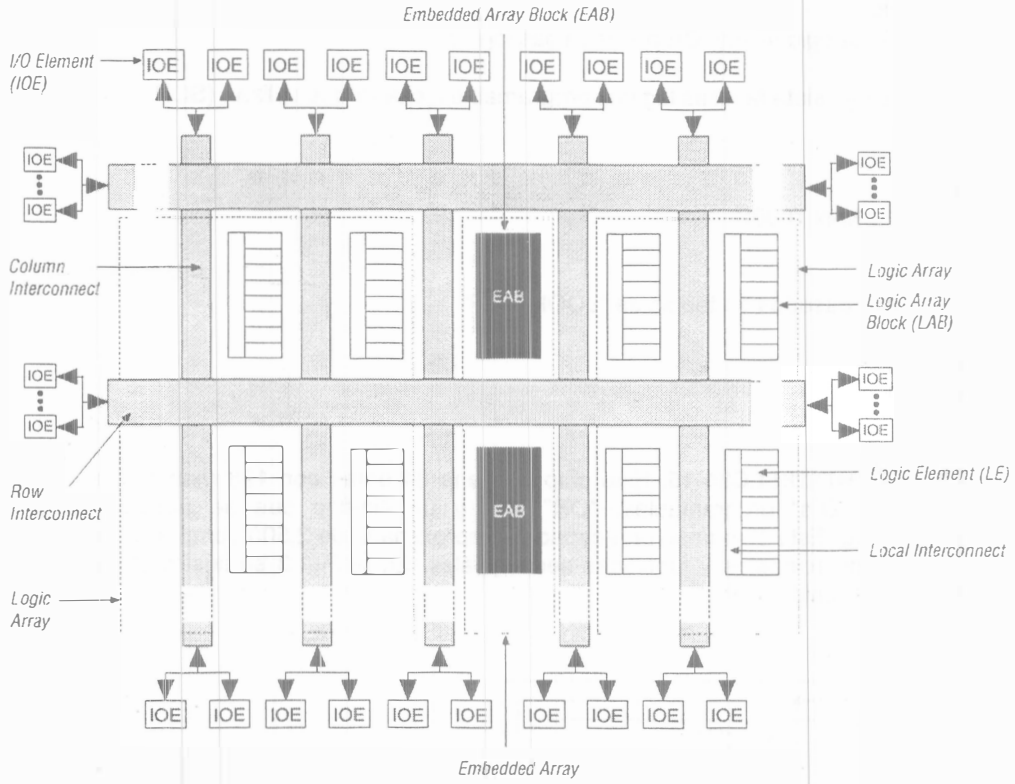


Fig 3 Estructura del dispositivo 10k20RC240-4

4. TIPOS DE ARCHIVOS DE DISEÑO.

Son archivos que contienen el diseño lógico para un proyecto en Max+Plus II, el cual es procesado por el compilador

Cuando se introduce un diseño, ya sea usando los editores "graphic", "waveform", "text", el compilador automáticamente puede procesar distintos archivos cuyas salidas dependen de la extensión del archivo fuente Los diversos tipos de archivos de diseño soportados por Max+Plus II son

- Archivo de diseño gráfico (Graphic) = gdf
- Archivo de diseño en texto AHDL = tdf (*)
- Archivo de diseño de forma de onda (Waveform) = scf
- Archivo de diseño de texto VHDL = vhd (*)
- Archivo de diseño de texto verilog = v (*)

- Archivo esquemático OrCAD = sch (*)
- Archivo de entradas Edif = edf (*)

- Archivo en formato "Netlist Xilinx" = xnf
- Archivo de diseños de Altera = adf
- Archivo de máquinas de estados = smf

Otros archivos secundarios que puede generar el compilador son

- Archivo de configuración y asignación = acf
- Archivo de símbolo = sym
- Archivo de vectores = vec
    - Hierarchy Display

Nota: Un asterisco (*) indica que esos archivos de diseño pueden existir como archivos de mayor importancia en un proyecto jerárquico El resto de los tipos de archivo deben ser únicamente subdiseños en un proyecto o existir como el nivel mas bajo en un proyecto jerárquico

4 1 Proyectos.

Un proyecto contiene todos los archivos de la jerarquía de un diseño El nombre del proyecto es el nombre del archivo del diseño que tenga la jerarquía más alta Max+Plus II realiza la compilación, simulación, análisis en el tiempo y programación del chip de un solo proyecto a la vez, sin embargo, se pueden editar archivos pertenecientes a otro proyecto Por ejemplo se pueden compilar proyect1, y editarlo como parte de proyect2 y salvar este segundo, pero si deseas compilar el proyect2, debes primero especificar que ahora el nombre de tu proyecto es proyect2

Nota: Como regla general se debe guardar cada proyecto en subdirectorios distintos

4 2 Archivos y ventanas de ayuda.

"ON –LINE –HELP" es un conjunto comprensivo de información que siempre esta disponible

Para obtener esta ventana se presiona "shift + F1" y se abrirá automáticamente la ventana de ayuda Esencialmente la ayuda es utilizada para

- Verificar procedimientos en "How to"
- Para resolver cualquier duda relacionada con el software Max+Plus II

El Help suministra una completa información acerca del software de Max+Plus II y enseña todo lo que se requiere saber acerca de las herramientas básicas de cada aplicación, de sus comandos, procedimientos, tips, mensajes y reglas de oro Todo acerca de funciones primitivas, megafunciones y macrofunciones, de los lenguajes AHDL, VHDL y Verilog HDL

También nos ofrece información acerca de los dispositivos y adaptadores de Altera, permitiéndonos seleccionar el dispositivo apropiado antes de empezar con nuestro diseño lógico y suministrar tips de cómo diseñar más efectivamente con las herramientas de Max+Plus II En la Fig 4 se ilustra la ventana de ayuda

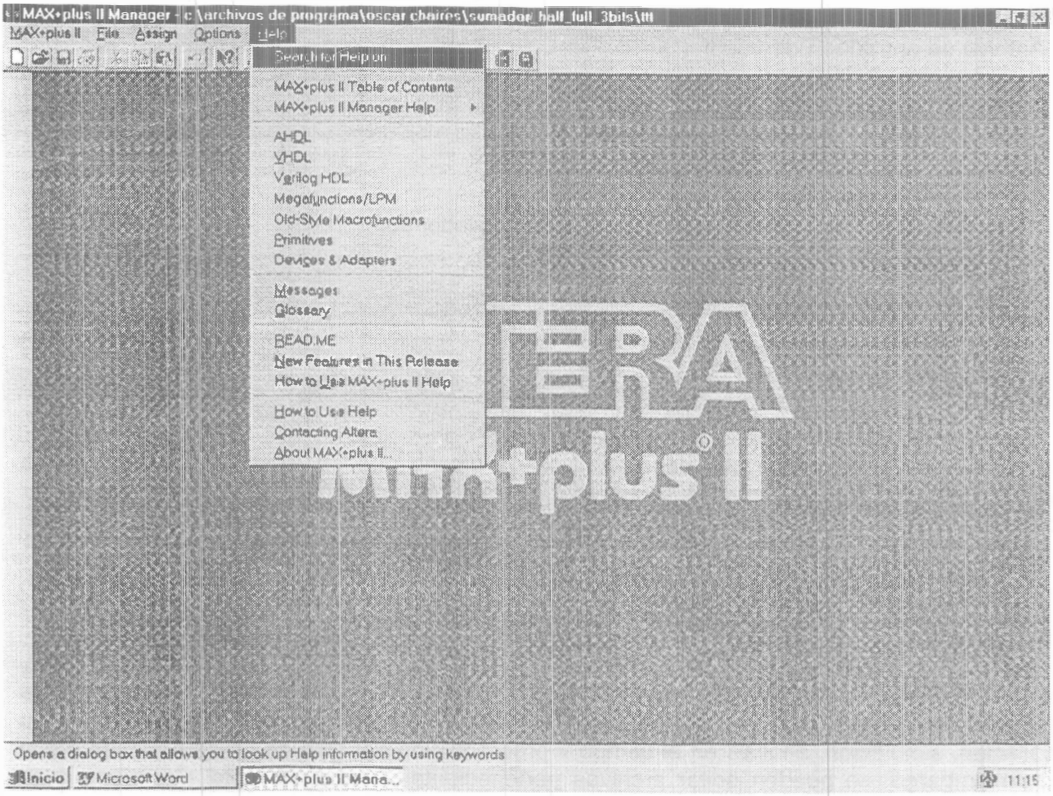


Fig 4 Ventana de ayuda

La ventana de ayuda puede ser utilizada de acuerdo a las siguientes opciones del mismo menú ilustrado en la Fig 4

- ***Search for help on.*** Tiene la función de un buscador de palabras, seleccionando las definiciones que concuerden
- ***Max+Plus II Table of contents.*** Enumera los tópicos más importantes y su contenido
- ***Table of contents.*** Da un informe del contenido de la aplicación abierta
- ***Introduction.*** Indica la forma general de utilizar la aplicación abierta
- ***Basic tools.*** Describe en detalle las herramientas a usar, los archivos de entrada y salida, acompañados de ilustraciones y ejemplos de la aplicación abierta
- ***Commands.*** Detalla los comandos con ilustraciones y ejemplos de la aplicación abierta
- ***Procedures.*** Indica el procedimiento paso a paso con ilustraciones y ejemplos de cómo especificar alguna tarea en la aplicación abierta
- ***Golden Rules.*** Contiene un sumario esencial de tipo y reglas para el uso de la aplicación abierta
- ***Short cuts.*** Obtiene en forma rápida los comandos del teclado, mouse, barra de herramientas de la aplicación abierta
- ***AHDL (Help on AHDL).*** Incluye instrucciones a detalle de cómo escribir un diseño en este lenguaje y que sintaxis usar
- ***VHDL (Help on VHDL).*** Incluye instrucciones de cómo desarrollar un programa de diseño con Max+Plus II utilizando VHDL, el soporte que este lenguaje tiene y una guía para la sintaxis a usar
- ***Verilog HDL (Help Verilog HDL).*** Incluye instrucciones sobre como desarrollar un diseño con Max+Plus II utilizando Verilog HDL describiendo el soporte que este lenguaje tiene, un estilo y sintáxis para usarlo

- ***Mega Functions/LPM.*** Contiene una lista de megafunciones incluyendo módulos de la librería (LPM) Si se selecciona una megafunción específica, se podrán ver sus reglas y forma de uso
- ***OLD – Style Macrofunctions.*** Listado en orden alfabético de todas las categorías de macrofunciones Se puede seleccionar una de esas categorías y se podrá ver el listado de los nombres de macrofunciones de esa categoría
- ***Primitives.*** Contiene una lista en orden alfabético de todos los elementos primitivos, si se selecciona específicamente un elemento primitivo, aparecerá su descripción y sus reglas para ser usado
- ***Devices & Adapters.*** Detalla una lista de dispositivos que la compañía Altera vende y que están aptos para programarse usando Max+Plus II, incluso tiene una guía de selección entre las familias de dispositivos que la compañía Altera tiene y se puede seleccionar un dispositivo en especial de esa lista y se podrá ver su descripción, localización de pines y celdas lógicas
- ***Messages.*** Enumera en orden alfabético información acerca de errores y advertencias que existieron al compilar un diseño y vienen acompañados de una explicación acerca del posible motivo del error
- ***Glossary.*** Lista los términos y sus definiciones usados en Max+Plus II
- ***Read – Me.*** Suministra una copia del archivo read me, en el cual se da información sobre los requerimientos mínimos del sistema, bugs conocidos y resueltos del software Max+Plus II
- ***How to use help.*** Indica el mecanismo para utilizar la ayuda
- ***How to use Max+Plus II help.*** Detalla la información de cómo usar la ayuda, incluyendo descripciones de distintas categorías de ayuda
- ***About Max+Plus II.*** Nos da información acerca del numero de la versión usada, aplicación abierta, derechos de autor y patente

5. DISEÑANDO EN MAX+PLUS II CON EL EDITOR GRÁFICO

Introducir un diseño significa el proceso de describir la arquitectura del diseño utilizando el método que sea soportado por Max+Plus II Existen muchos métodos para introducir un diseño, tres editores a seleccionar (Text Editor, Graphic Editor, Waveform Editor), y dos editores secundarios para ayudar a introducir un diseño (Floorplan Editor, Symbol Editor) Algunas simples reglas a seguir son las siguientes

- La captura de cualquier esquemático, se efectúa en el editor gráfico
- La captura de diseños hechos en lenguaje de descripción AHDL, se efectúa en el editor de texto
- La captura de diseños hechos en lenguaje de descripción VHDL, se efectúa en el editor de texto
- La captura de diseños hechos en lenguaje de descripción Verilog HDL, se efectúa en el editor de texto
- La captura de diseños hechos en forma de onda se efectúa en el editor de forma de onda
- La captura de archivos generados en Edit, netlist y Xilinx netlist que son diseños hechos con herramientas EDA de otras industrias se efectúa en el editor de texto

Esta sección suministra información para aprender a diseñar con MAX+PLUS II, y de esta forma, saber que los diseños trabajan apropiadamente al ser compilados en MAX+PLUS II, pero esta sección no describe construcciones a detalle, en su lugar describe combinaciones de construcciones que pueden ser usados para crear una lógica de muy alto nivel

Al introducir un diseño, tanto el dispositivo usar, como la asignación de parámetros puede ser hecha por nosotros o permitir que el proyecto al ser compilado haga esta asignación automáticamente, en la Fig 5 se ilustra el menú que permite hacer esta asignación de forma manual

Entenderemos por "**resource**" una porción de un dispositivo de "Altera" como un pin o celdas lógicas que realiza un trabajo específico También se puede asignar manualmente los recursos, para asegurarnos que el compilador de Max+Plus II acomode exactamente el diseño en la forma que lo deseamos

Un diseño jerárquico se logra utilizando las metodologías "top-down" (arriba-abajo) o "bottom-up" (abajo-arriba) Se pueden crear diseños grandes y complejos, fáciles de manejar o utilizar la librería de módulos (LPM) para crear subdiseños

También se pueden usar macrofunciones ya diseñadas que clasifiquen los requerimientos de algunos diseños, del mercado (PCI, PCMCIA, ATM, DSP)



Fig 5 Ventana para la asignación de recursos en forma manual

5.1 Editor Gráfico (Graphic Editor)

Este editor nos permite introducir diseños en un ambiente visual Este editor permite crear y editar archivos (gdf) que pueden incluir una combinación de megafunciones, macrofunciones y símbolos primitivos Para acceder al editor gráfico se requiere hacer un "click" en el menú de las once distintas aplicaciones y seleccionar el "Graphic Editor" La ventana del editor gráfico será abierta Haciendo un click en el segundo renglón de la barra de herramientas sobre el ícono "Symbol" o haciendo un "click" a la mitad de la pantalla, aparecerá una ventana, en la cual se mostrarán las distintas bibliotecas, y se pueden seleccionar más o menos 300 símbolos distintos que pueden ser combinados libremente en este editor en un diseño de un proyecto jerárquico Los símbolos se interconectan dibujando líneas, nodos o buses Los buses representan un grupo lógico de nodos En la Fig 6 se ilustra la ventana del editor gráfico junto con la ventana de bibliotecas La primer biblioteca de esa ventana es para que el usuario guarde símbolos de su creación Seleccionando esta librería, aparecerán en la misma ventana los nombres de los símbolos creados por el usuario El segundo y subsiguientes renglones de esa ventana constituyen las librerías proporcionadas por el fabricante, a saber

- **Prim:** Significa componentes primitivos (compuertas lógicas, Flip-Flops, entradas, salidas, etc) Si se hace un "click" en este renglón aparecerán encerrados en un rectángulo todos los símbolos primarios y seleccionando alguno, aparecerá en la pantalla
- **mf:** Significa macrofunciones (tiene a toda la familia 74, y funciones diseñadas por Altera) Si se hace un "click" en este renglón, en el rectángulo aparecerán todos los símbolos que existen en está librería, y se puede seleccionar cualquiera y aparecerá en la pantalla
- **mega_lpm:** Son megafunciones (tiene módulos y megafunciones hechas por Altera)

En esta misma ventana, hasta arriba, tenemos un rectángulo pequeño que es un buscador, en el cual, si sabemos el nombre del símbolo deseado, con sólo anotarlo ahí y presionar "enter" aparecerá en la pantalla

Nota: Para escribir el nombre se debe saber como esta escrito en MAX+PLUS II, por ejemplo, si queremos un Flip-Flop "D", debemos escribir en el buscador "DFF"

Para mover un símbolo se presiona el puntero del mouse en medio del símbolo que se desea mover, y sin soltarlo se lleva hasta donde deseamos posicionarlo y finalmente lo soltamos

Este editor cuenta con una columna a la izquierda, en la cual se pueden seleccionar distintos tipos de líneas, simplemente se hace un "click" en alguna y después al llegar al "área de trabajo", en lugar de flecha en la pantalla se verá una cruz y al unir un símbolo con otro, sólo se presiona el mouse donde se quiere que empiece la línea y se desliza el mouse sin soltarlo hasta donde se quiere que termine

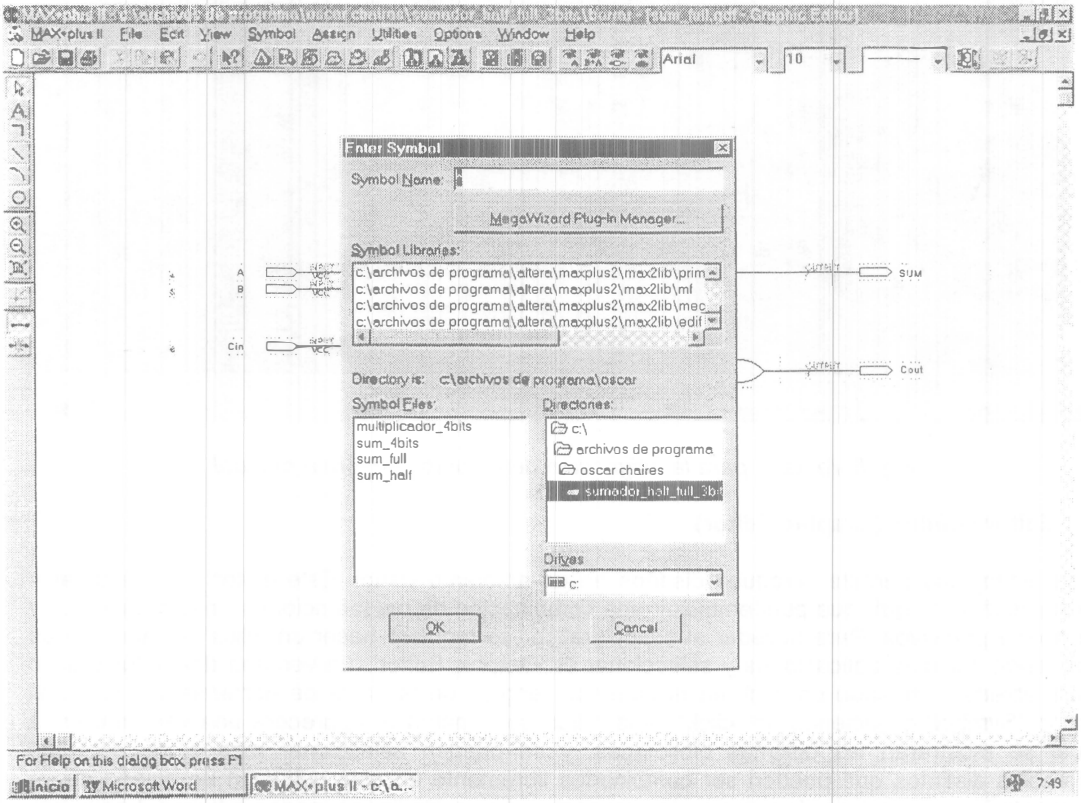


Fig 6 Ventana del editor gráfico y de las librerías

Ejemplo 1: Si se requiere introducir el diseño de las siguientes funciones

$$OUT1 = a_0\, a_1$$
$$OUT2 = a_0\, a_1 + b$$

Solución:

Paso 1 Estando en el editor gráfico se debe decidir si queremos hacer nuestro diseño en algún dispositivo en especial o dejaremos que sea asignado cualquiera de forma automáticamente Si tenemos planeado, por ejemplo, hacer nuestra implementación en el dispositivo "EPM7128SLC84-15", lo que se requiere hacer es un "click" en el menú de "Options", arriba de los iconos aparecerá una ventana en la que la primer palabra es "Device", se hace un "click" ahí, y aparecerá otra ventana en la cual se nos pregunta la familia y debemos seleccionar

"MAX7000S", al hacerlo en un rectángulo que aparece en la misma ventana, debemos seleccionar el dispositivo, así que buscamos el "EPM7128-15" y debemos hacer un "click" en "OK", y de nuevo nos encontraremos en la ventana del editor gráfico

Paso 2 En la zona de trabajo en el lugar donde queremos, que aparezca el símbolo, hacemos un doble "click" y se abrirá la ventana que contiene todas las distintas librerías, pero como ya sabemos lo que queremos, en el buscador escribimos "OR 2, oprimimos "OK" y en nuestra ventana veremos la OR de dos entradas, así poco a poco podremos tener en pantalla todo lo que necesitamos en nuestro diseño, recordando que no se vale dejar ninguna línea sin conectar, para las entradas necesitamos "input" (tres) y para las salidas "output" (dos), vea la Fig 7 para detalles

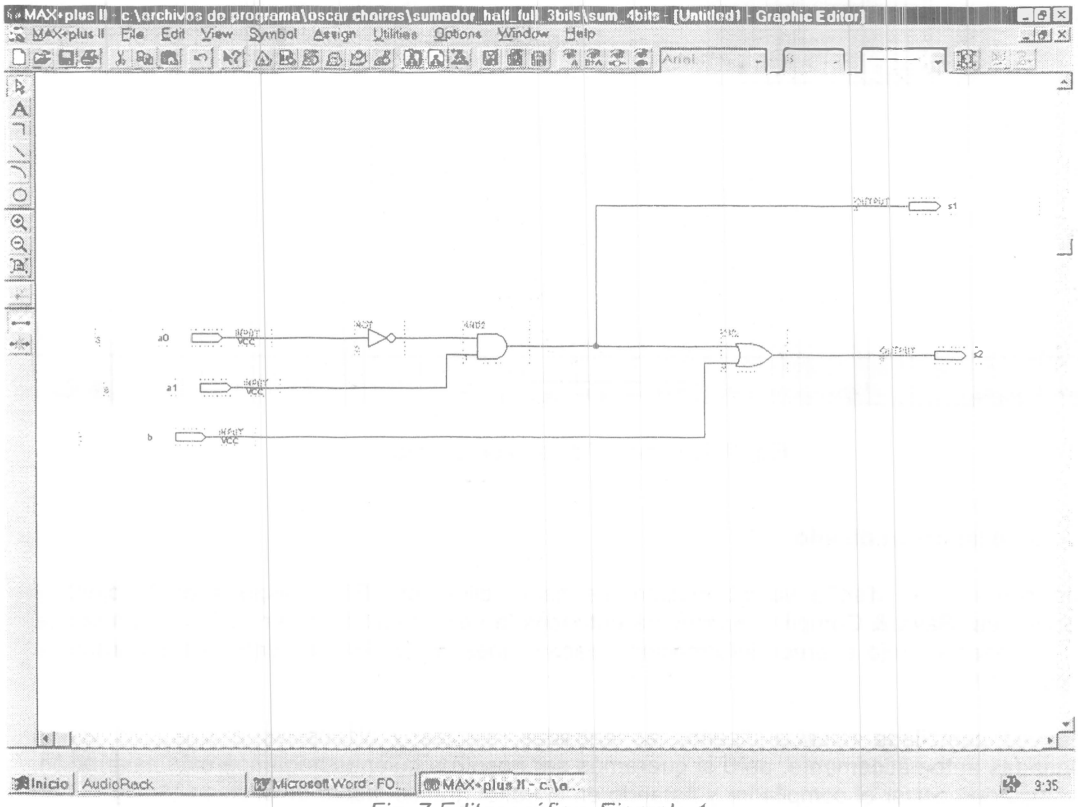


Fig 7 Editor gráfico Ejemplo 1

Paso 3 Se hace un "click" en donde dice "File" y aparecerá un menú y hacemos de nuevo un "click" en "Save as", aparecerá una ventana en donde debemos escribir el nombre del diseño y hacemos un "click" en "OK"

Paso 4 De nuevo hacemos "click" en "File" y seleccionamos "Project" y ahora hacemos otro "click" en "Set project to current file"

Estos últimos dos pasos se ilustran en la Fig 8 En este momento acabamos de introducir nuestro diseño, así que el siguiente paso será compilar

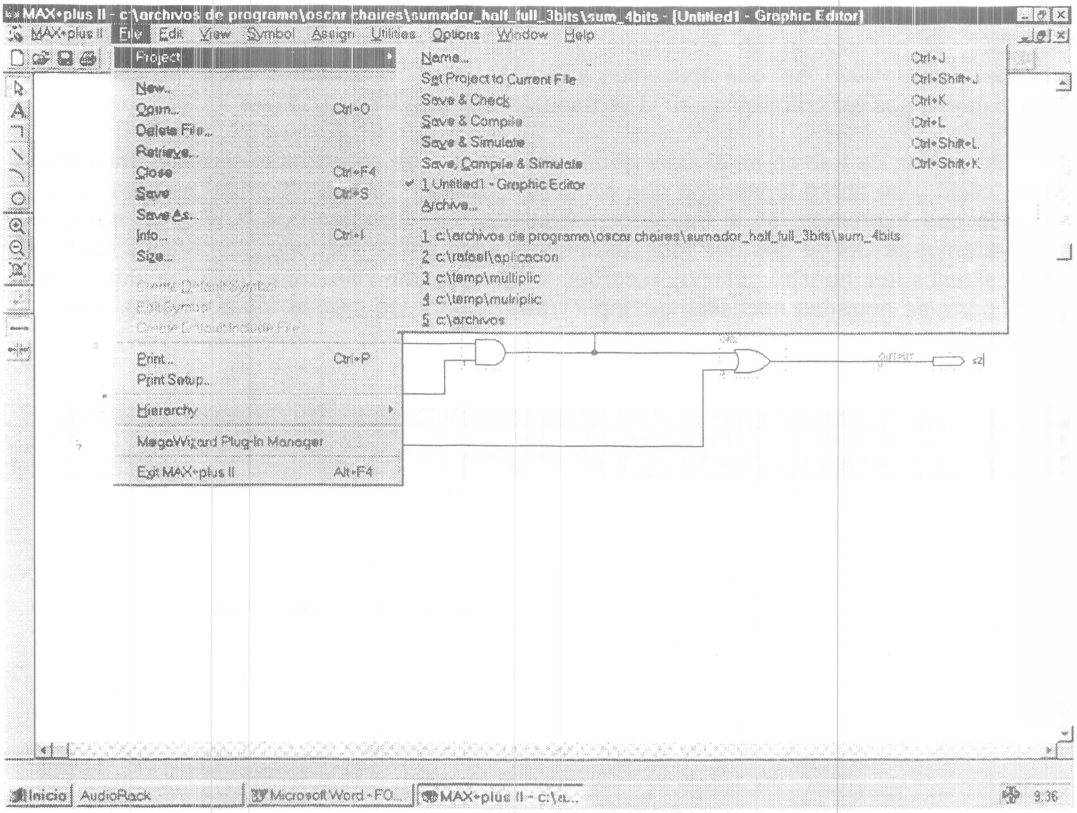


Fig 8 Salvando el diseño del Ejemplo 1

5.2 Compilando un diseño

Para compilar un diseño ya introducido, se hace "click" en "File" después en "Project" y después en "Save & Compile" Aparecerá entonces la ventana ilustrada en la Fig 9, en la que en renglones en rojo aparece información, precauciones y los errores detectados durante la compilación

Una vez compilado nuestro diseño, las entradas, salidas y dispositivo a usar, ya fueron asignadas automáticamente, pero si queremos ser nosotros quienes hagamos esta asignación, tenemos que cerrar el compilador y estando en el editor gráfico, hacer un click en "Assign", otro click en "device", y entonces aparecerá una ventana en la que tenemos que seleccionar la familia en el buscador y al hacerlo tendremos que seleccionar el numero del dispositivo de la familia a usar, como se ilustra en la Fig 10

De esta forma acabamos de seleccionar el dispositivo a usar ahora se necesitan seleccionar los pines en donde nosotros queremos que se encuentren nuestras entradas y nuestras salidas, para ello debemos ir al menú y seleccionar el "Floorplan Editor" La forma de seleccionar el "floorplan" editor se ilustra en la Fig 11

En la ventana del "floorplan editor" como se muestra en la Fig 12 se tiene en una ventana todas las entradas y salidas que posee nuestro diseño y abajo se encuentra el dispositivo seleccionado

Nuestras entradas y salidas únicamente deben estar en los pines que digan (I/O) Ya que los otros pines son utilizados ya sea para ser programado o para polarizar el chip

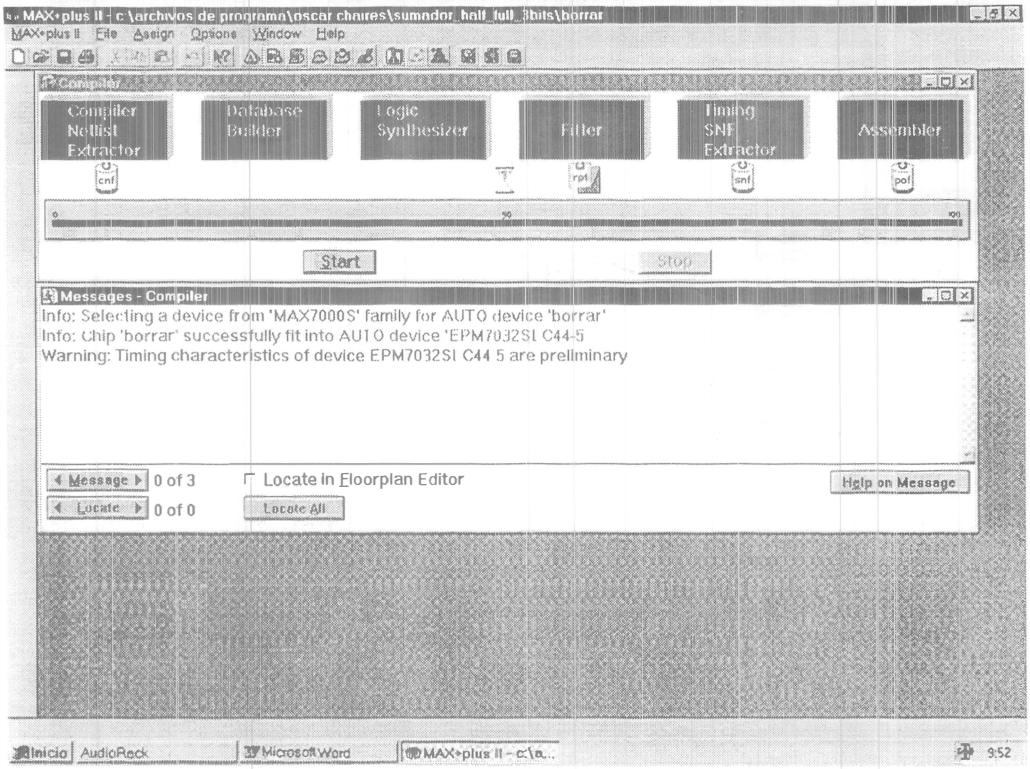


Fig 9 Compilando el diseño del Ejemplo 1

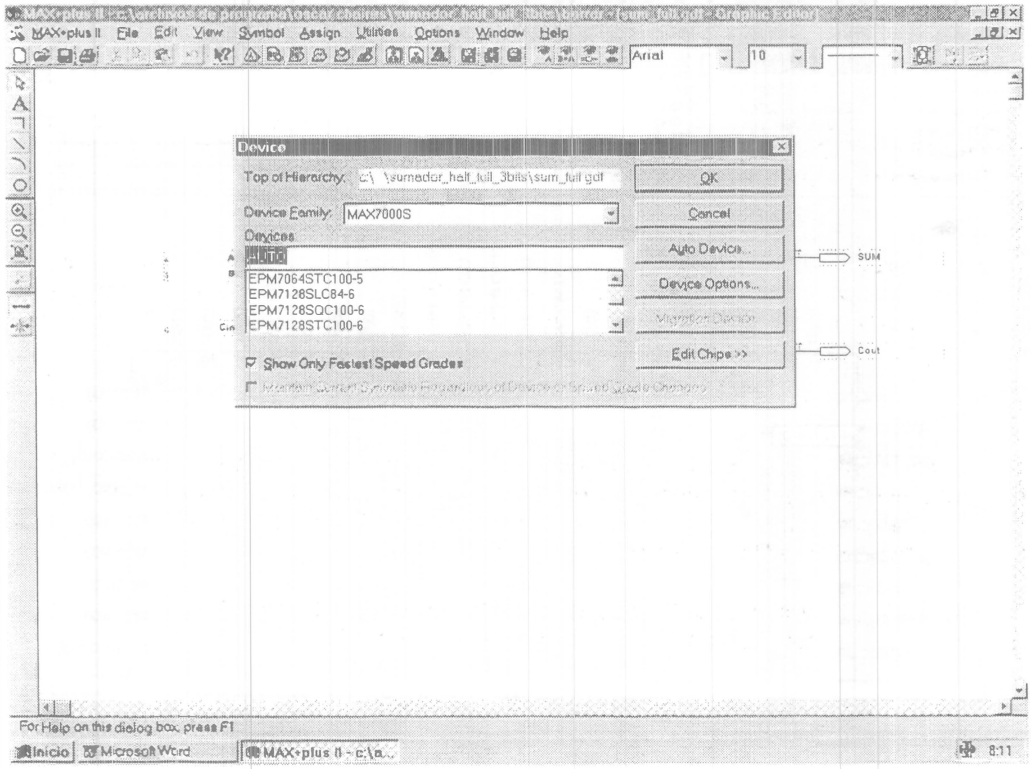


Fig 10 Asignando el dispositivo manualmente para el diseño del Ejemplo 1

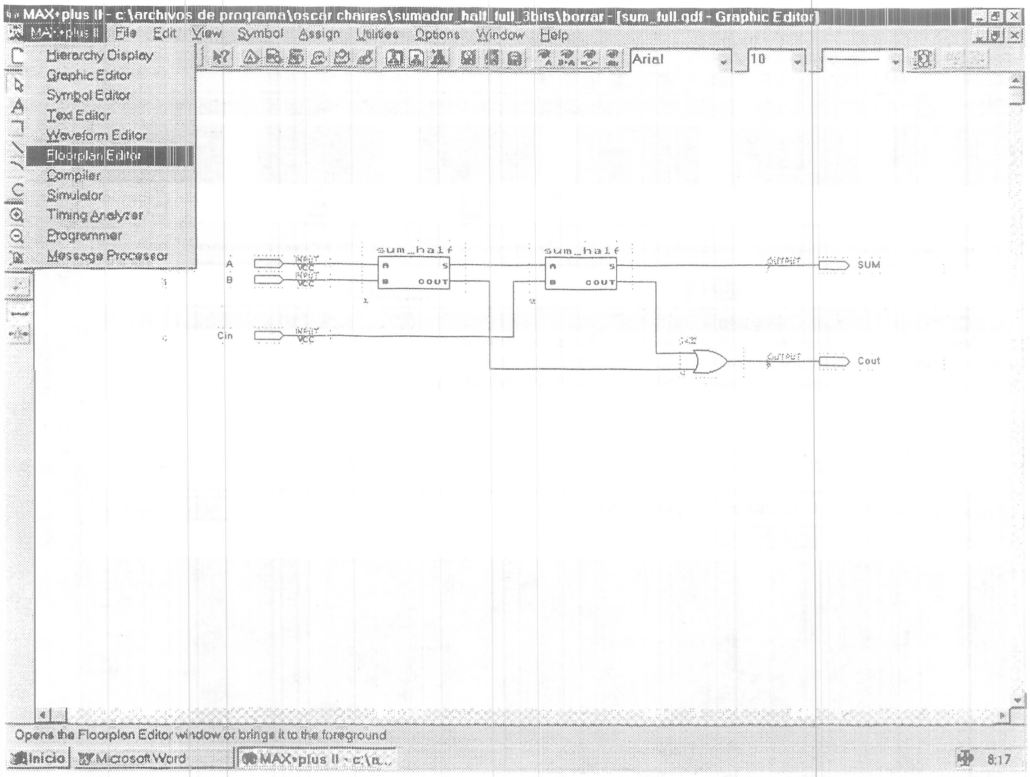


Fig 11 Forma de acceder al "floorplan editor" para el diseño del Ejemplo 1

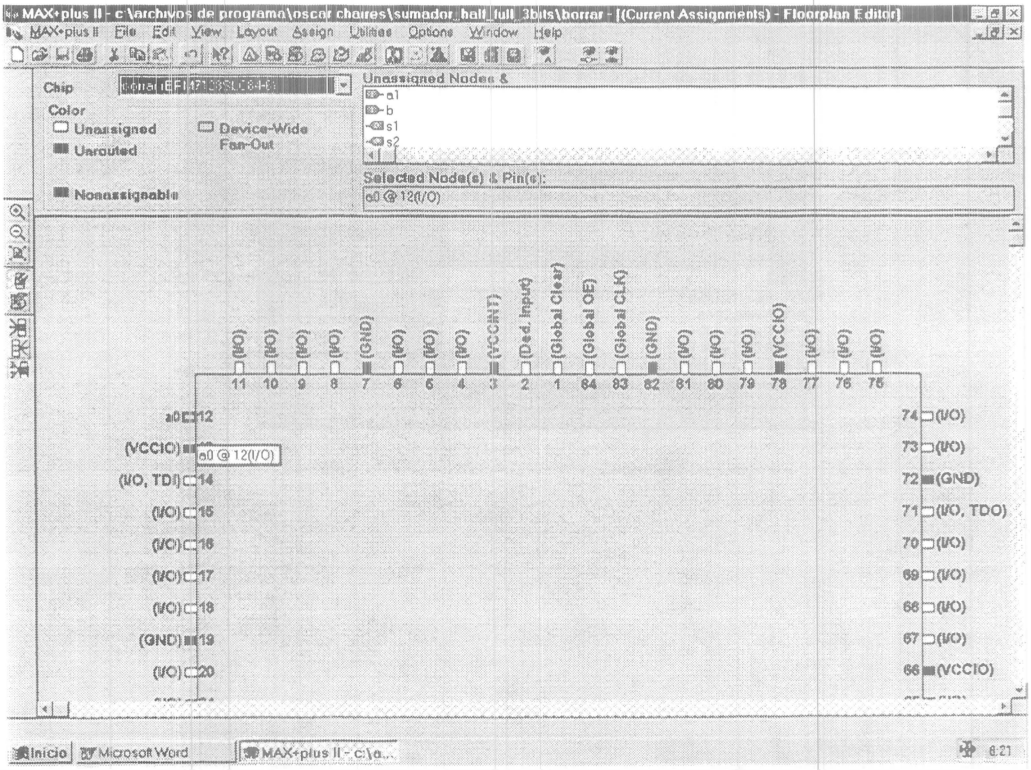


Fig 12 Vista del "floorplan editor" para el diseño del Ejemplo 1

La forma de hacer asignaciones en el "floorplan editor" es desde la ventana donde se encuentran las entradas y salidas de nuestro diseño Se selecciona un elemento y se oprime el puntero del mouse y sin soltarlo se lleva hasta el lugar en donde se encuentra el "pin" donde deseamos quede ese elemento, y soltamos el puntero del mouse Esto lo hacemos con todas nuestras entradas y salidas y posteriormente tenemos que volver a compilar nuestro diseño para que los cambios tengan efecto

Para saber si en realidad todo quedó como lo deseamos cerramos el compilador y hacemos un click en el menú Max+Plus II en la opción "Hierarchy Display" y ahí aparecerán todos los archivos generados en el diseño y también muestra (si es que tuvo sub-diseños) la jerarquía de los archivos empezando por los de la más alta jerarquía, como se ilustra en la Fig 13

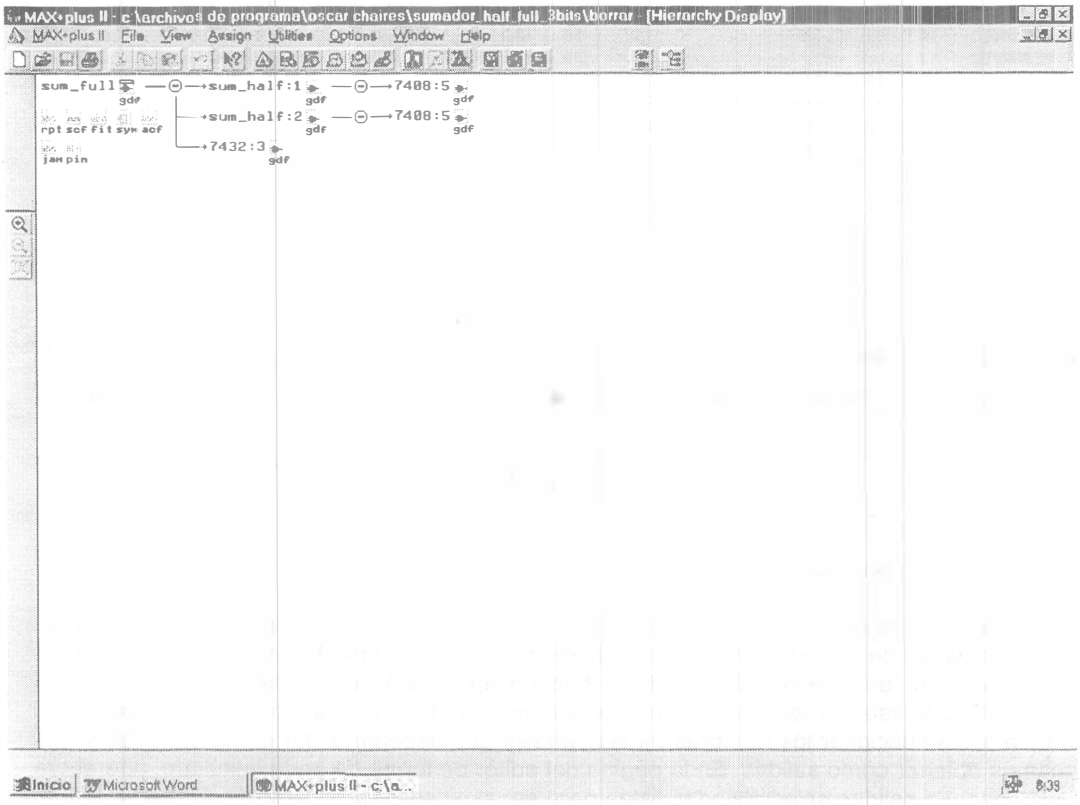


Fig 13 Vista de la jerarquía del proyecto para el diseño del Ejemplo 1

5.3 Archivo de reporte.

En el archivo de reporte "RPT" (que es el reporte de nuestro diseño) hacemos un "click" y aparecerá entre otras cosas el dispositivo a usar, la asignación de pines y la cantidad de recursos utilizados entre otras cosas, como se ilustra en la Fig 14

Otra aspecto que también es importante observar antes de programar el chip es ver la simulación del funcionamiento de nuestro proyecto Para ello necesitamos abrir el editor de forma de onda (waveform editor)

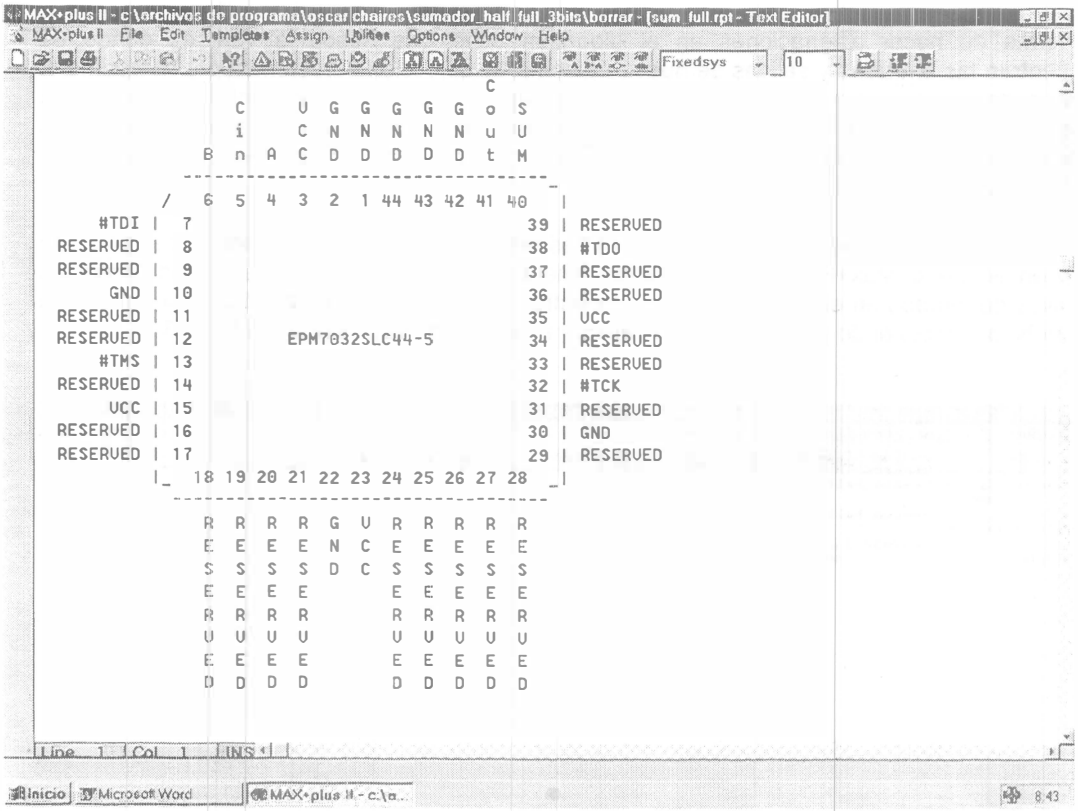


Fig 14 Vista del archivo de reporte RPT mostrando la asignación final de pines para el diseño del Ejemplo 1

5.4 Simulando un proyecto

Para simular un proyecto debemos, abrir el editor de forma de onda, desde el menú de las once aplicaciones de Max+Plus II Ya estando en el editor de forma de onda hacer un click en "node" y otro en "enter nodes from snf", al hacerlo aparecerá una ventana donde debemos oprimir "list", y la lista de todo el conjunto de entras y salidas de nuestro proyecto aparecerá Ahí debemos seleccionar los nombres de las señales que deseamos dar como entradas y que deseamos obtener como salidas En la página del editor de forma de onda intitulado, aparecerá las entradas y/o salidas seleccionadas listas para ser simuladas tal y como se ilustra en la Fig 15

Para simular, se requiere que nosotros demos distintos valores a los nodos de entrada, la forma de hacerlo es presionando con el puntero del mouse el espacio de la entrada a la cual se le quiere asignar un valor y soltarlo hasta abarcar el espacio en el que queremos suponer, esa entrada estará presente Al hacerlo esa sección se pondrá en negro y la columna a la izquierda en color azul mostrará un icono con un pulso en uno lógico, y en el icono de abajo un pulso con un cero lógico Dependiendo del valor que en ese espacio se desee tener en el nodo, se oprime uno de esos dos iconos De esta manera podemos comenzar a dibujar las excitaciones a nuestro circuito Este proceso se ilustra en la Fig 16

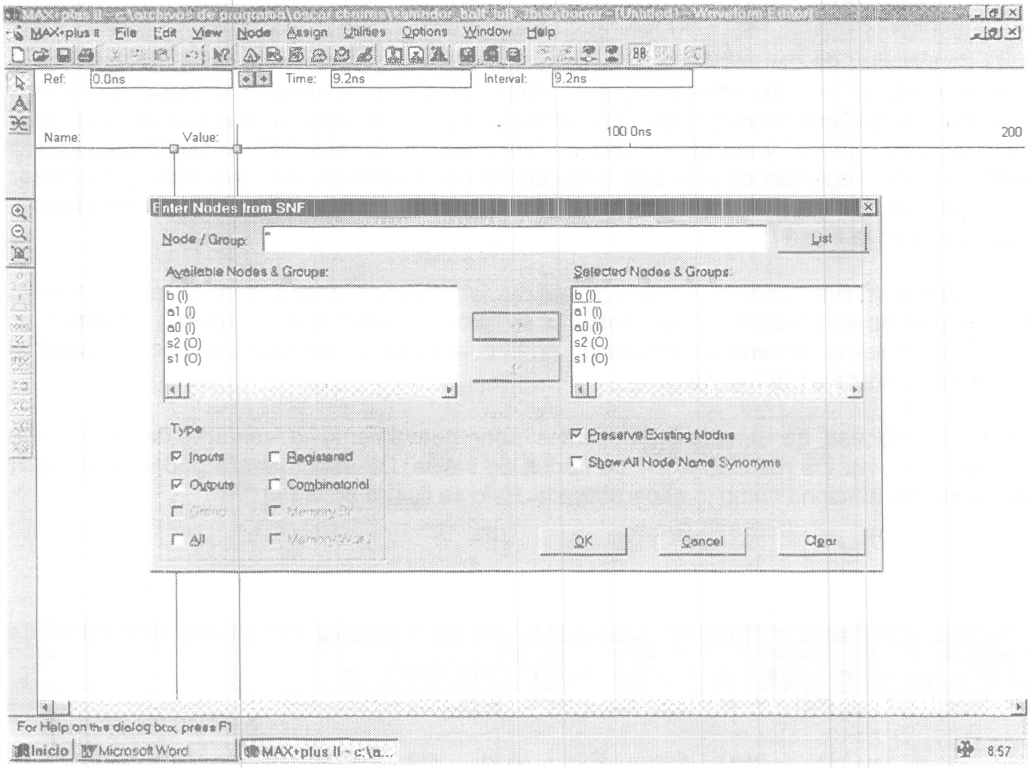


Fig 15 Vista del "waveform editor" mostrando los nodos a ser mostrados por el simulador para el diseño del Ejemplo 1

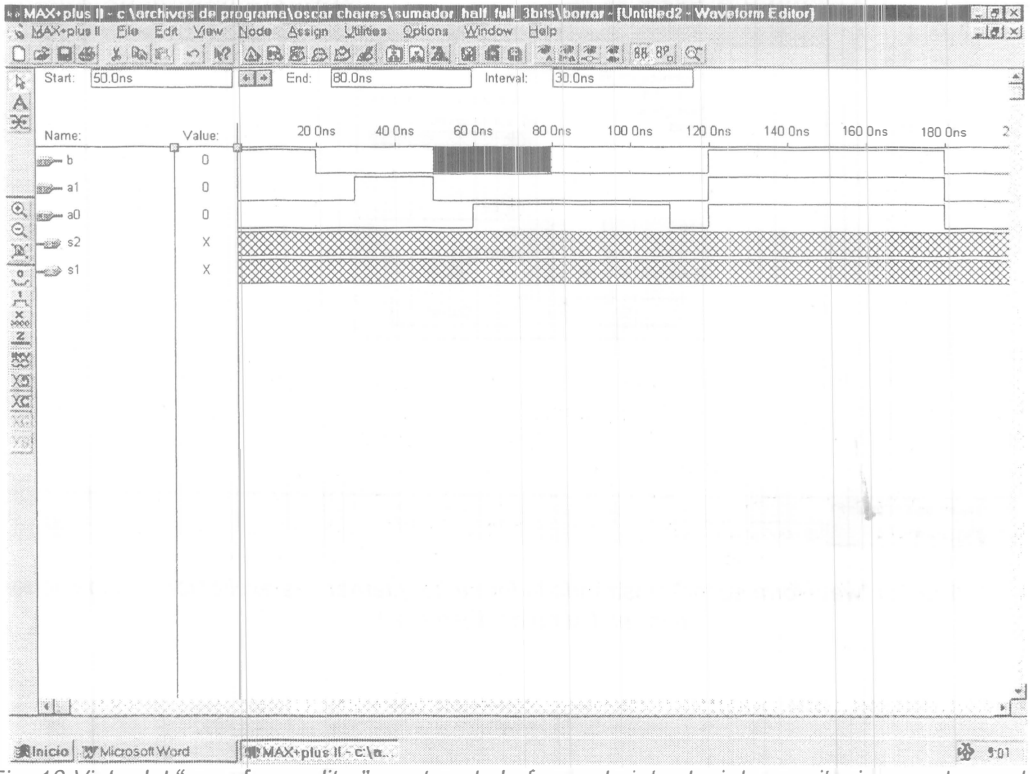


Fig 16 Vista del "waveform editor" mostrando la forma de introducir las excitaciones a los nodos para el diseño del Ejemplo 1

Después de terminar de dar valores a todas nuestras entradas, se requiere hacer la simulación Los pasos a seguir son los mismos que al introducir cualquier diseño, excepto que como nuestro proyecto ya tiene nombre nos saltaremos ese paso A continuación hacemos un click en "file", luego en "project" y por último otro en "set project to current file" Al hacerlo aparecerá una ventana para preguntarnos si lo queremos salvar con extensión scf, tenemos que aceptar con "o k " ya que esa es la terminación para indicar que se trata de un archivo de simulación, como se ilustra en la Fig 17

Para ejecutar una simulación se requiere utilizar los mismos pasos que ya anteriormente hicimos, esto es hacer un click en "file", después en "project" y otro más en "save & simulate" Aparecerá entonces la ventana de simulación en el cual se debe de hacer un click, en "start" el panorama se ilustrada en la Fig 18

Terminado el proceso de simulación debemos abrir nuevamente la ventana del "waveform editor" para observar los resultados en los nodos de salida De esta manera podemos concluir si nuestro diseño funciona como lo esperábamos Esto se ilustra en la Fig 19

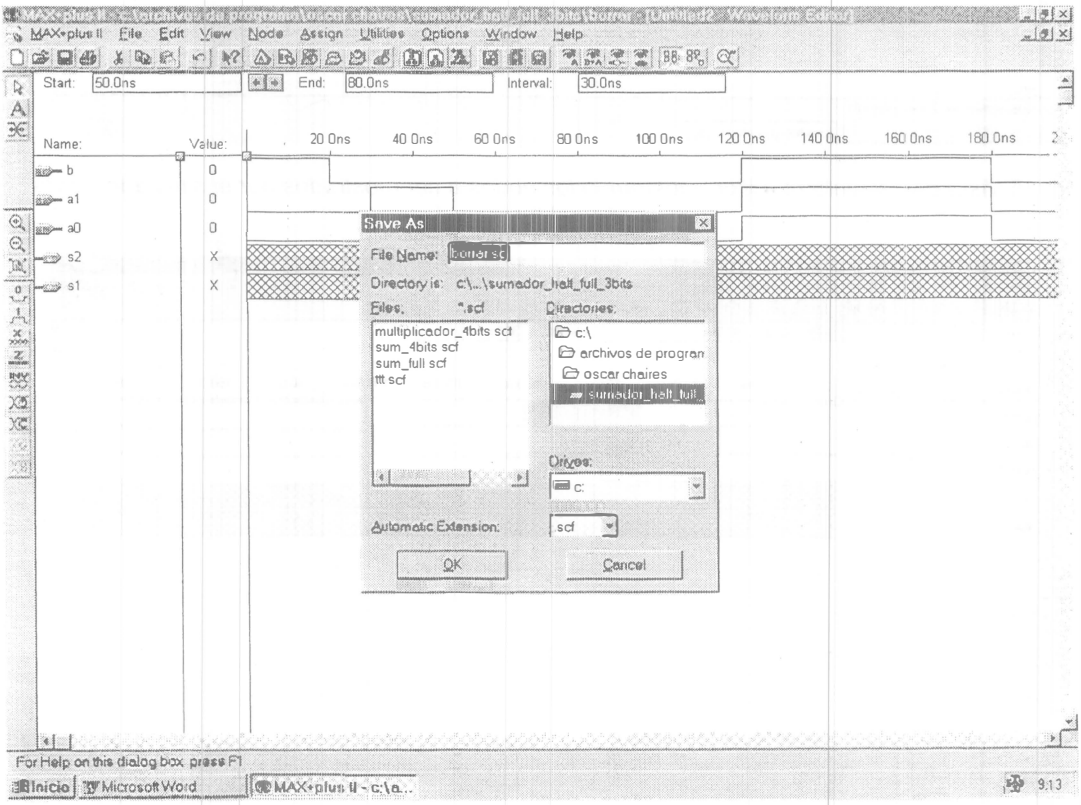


Fig 17 Vista del "waveform editor" mostrando la forma de guardar las excitaciones a los nodos para el diseño del Ejemplo 1

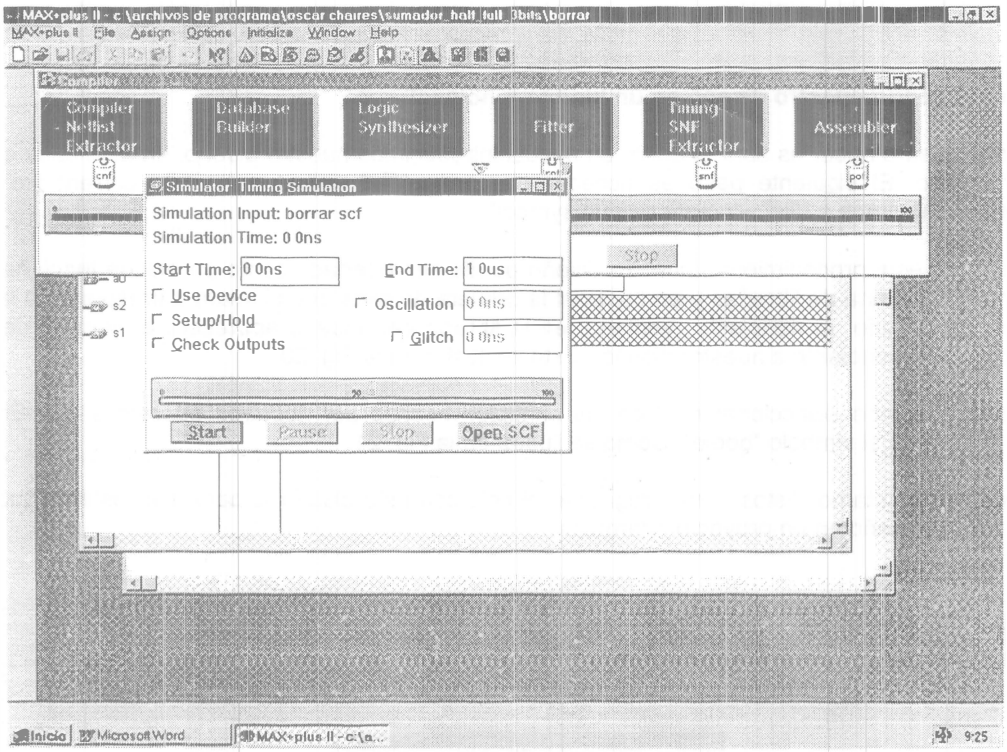


Fig 18 Vista del "waveform editor" mostrando la forma de iniciar la simulación para las excitaciones a los nodos en el diseño del Ejemplo 1

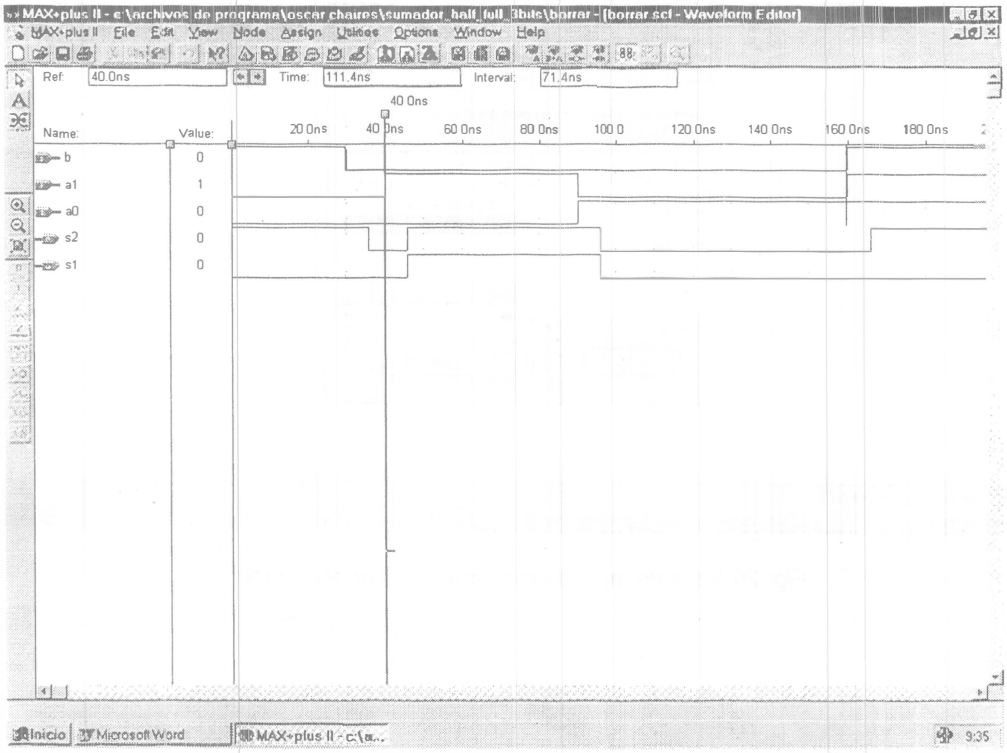


Fig 19 Vista del "waveform editor" mostrando los resultados después de ejecutar el proceso de simulación para las excitaciones a los nodos en el diseño del Ejemplo 1

5.5 Convirtiendo nuestro diseño en un bloque funcional.

El primer paso a hacer es abrir el editor en el cual introducimos nuestro diseño, en este caso el editor gráfico El siguiente paso es acceder el menú "file" – "set project to current file", posteriormente irnos a "file" –"create default symbol"

Después de este procedimiento, nuestro diseño puede ser insertado como un bloque funcional abriendo la ventana de librerías y en especial la primera de ellas que siempre corresponde a la librería del usuario Al abrir esta ventana, en la librería del usuario aparecerá el nombre del símbolo correspondiente a nuestro diseño Esto se ilustra en la Fig 20

Después debemos seleccionar cual de los símbolos hechos por nosotros queremos ver En nuestro caso es el símbolo "boole" Como se ilustra en la Fig 21

En este punto estamos listos para programar el chip con este diseño, ó para usar este bloque como un subdiseño en un proyecto jerárquico

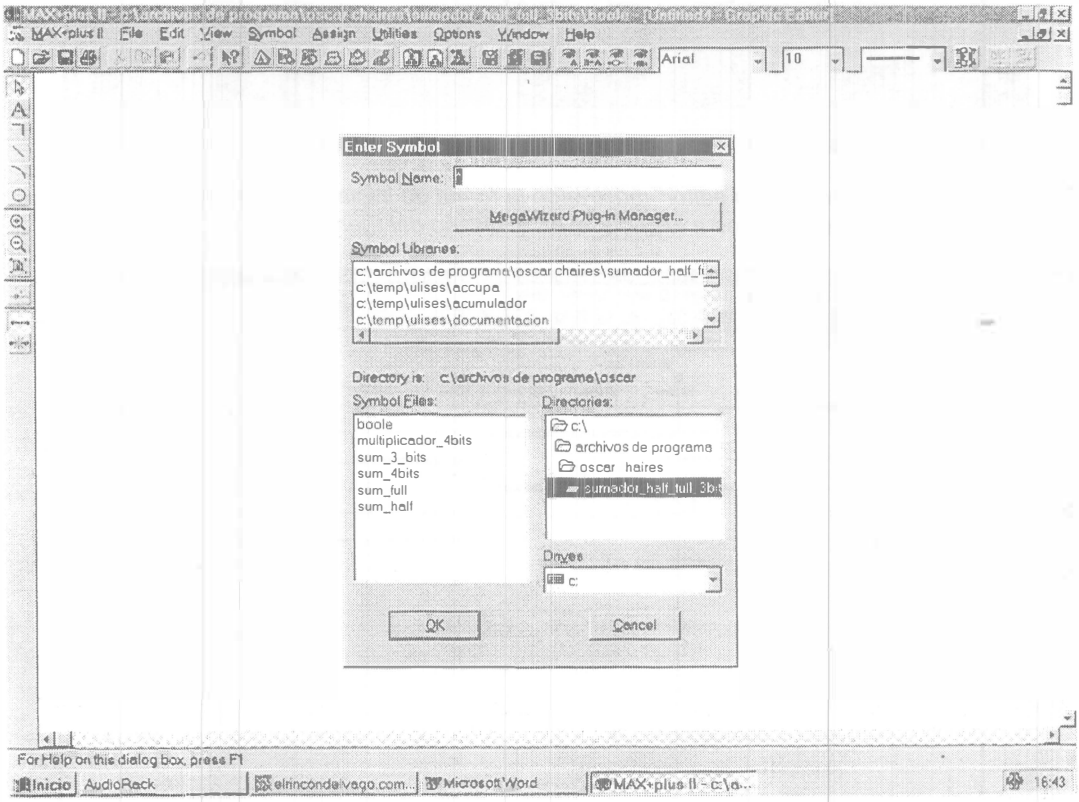


Fig 20 Vista de los símbolos creados por el usuario

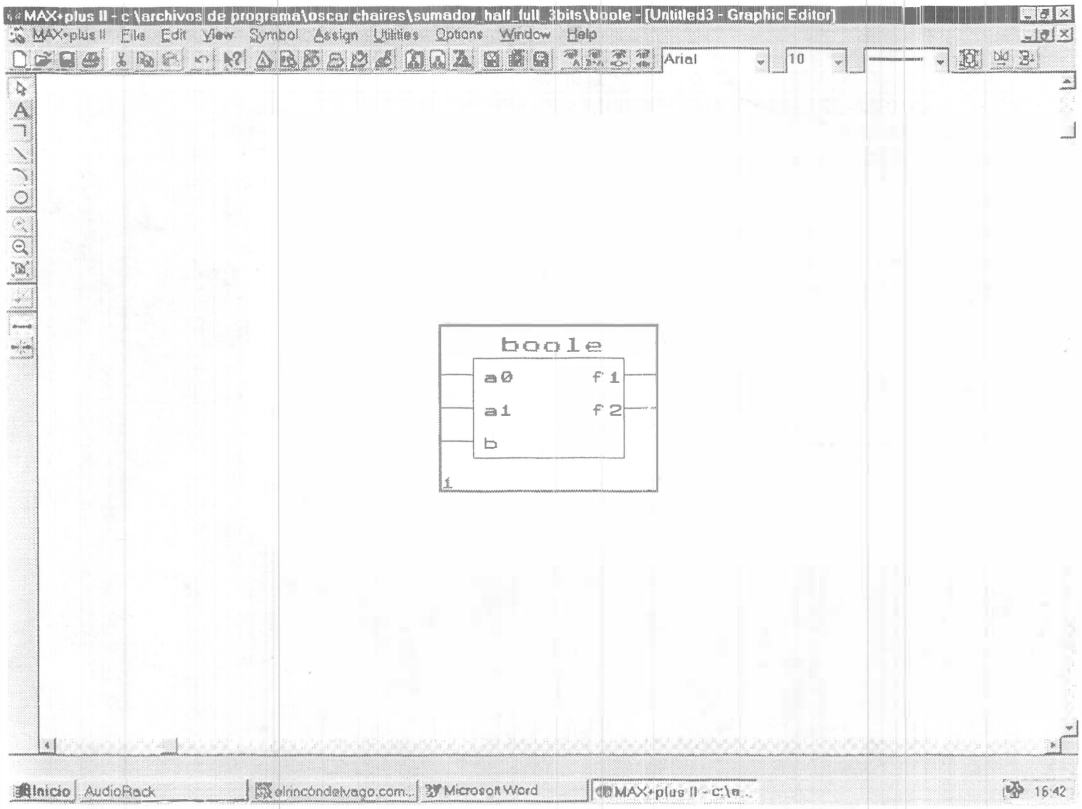


Fig 21 Vista del símbolo creados e insertado por el usuario

5.6 Programando el dispositivo

Si el caso es programar físicamente el chip entonces requerimos primero seleccionar el chip y la asignación de pines, si es que no lo hemos hecho, para ello se requiere asignar algún dispositivo y volver a compilar el proyecto incluyendo la simulación, como ya se ha descrito en las secciones anteriores

Para programar el dispositivo se requiere abrir la ventana del programador, ya sea desde el menú de las once aplicaciones mediante la selección de la aplicación que dice "programmer", o desde el compilador haciendo un click en el cilindro que dice " pof " En este momento se observará una ventana parecida a la ilustrada en la Fig 22

En este momento, se requiere que la tarjeta UPX ó UP1 se encuentren conectada en el puerto paralelo de nuestra computadora y debidamente polarizadas Finalmente solo se necesita hacer un click en donde dice "program", para que el dispositivo a usar sea grabado con la información de nuestro proyecto

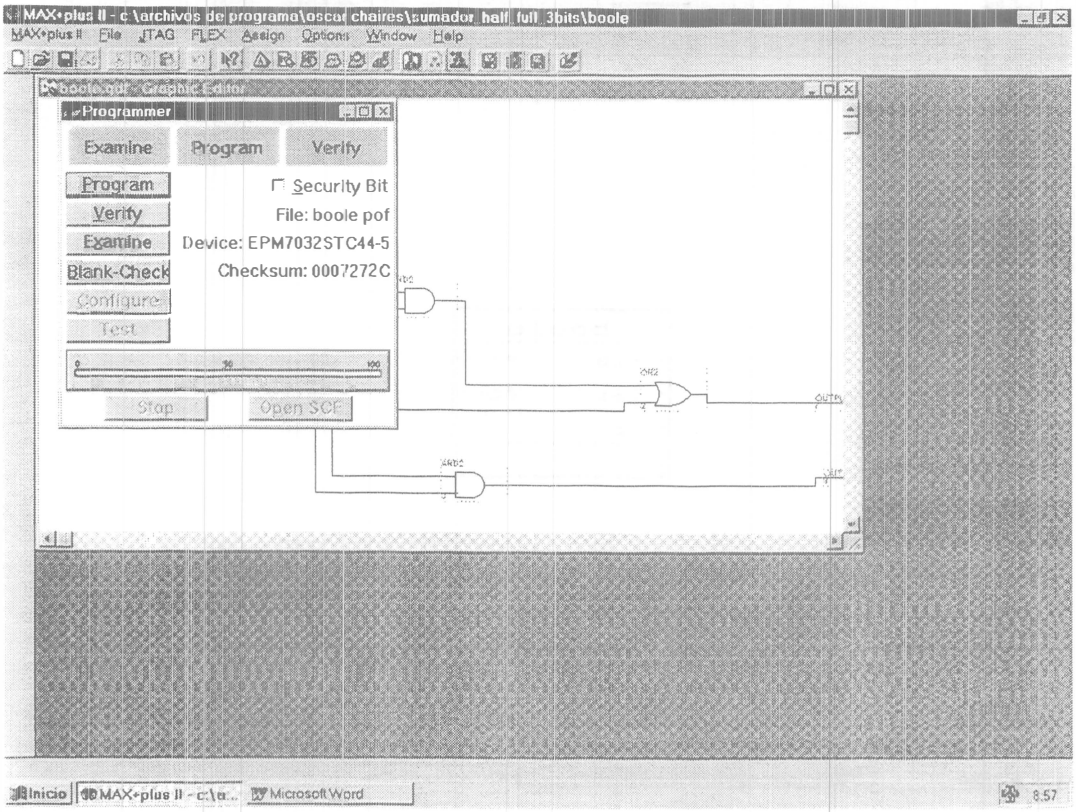


Fig 22 Vista de la ventana del programador

6. DISEÑANDO EN MAX+PLUS II CON EL EDITOR DE TEXTO.

Este editor sirve para introducir diseños que se encuentren descritos en algún lenguaje de descripción de hardware Existen varios lenguajes que pueden ser usados en este editor, que son el AHDL, VHDL, y VERILOG HDL Estos lenguajes también tienen plantillas (templates) para ver el formato o sintaxis requerido para la introducción de algún diseño También se pueden introducir diseños hechos en "Edif " ,"Xilinx" y "Palasm"

Para empezar lo que debemos hacer es abrir una ventana nueva del editor de texto, desde el menú "file" A continuación podemos introducir el diseño siguiendo la sintaxis del lenguaje de descripción de hardware que deseamos usar Para facilitar esta tarea podemos hacer uso de las plantillas que vienen incorporadas como parte del software Max+Plus II, para lo cual hacemos un click en donde dice "templates ", en donde se podrán seleccionar entre las distintas plantillas de "ahdl, vhdl, y verilog hdl" Este proceso se ilustra en la Fig 23

Nota: Para más información sobre estos lenguajes de descripción de hardware ver el manual de "ahdl" y "verilog hdl" de esta misma serie

Lo primero que se debe saber antes de introducir algún diseño de esta manera es que todo diseño que se quiera introducir aquí puede ser dividido en dos grandes partes

- Elementos del diseño
- Parte lógica

La primera parte debe tener un titulo o un include (cuando lo que se haga sea programar algo ya hecho en alguna biblioteca) y la interfase de nuestro diseño como son las entradas ,salidas pines bidireccionales ,etc La segunda parte es introducir el comportamiento que queremos que siga nuestro diseño

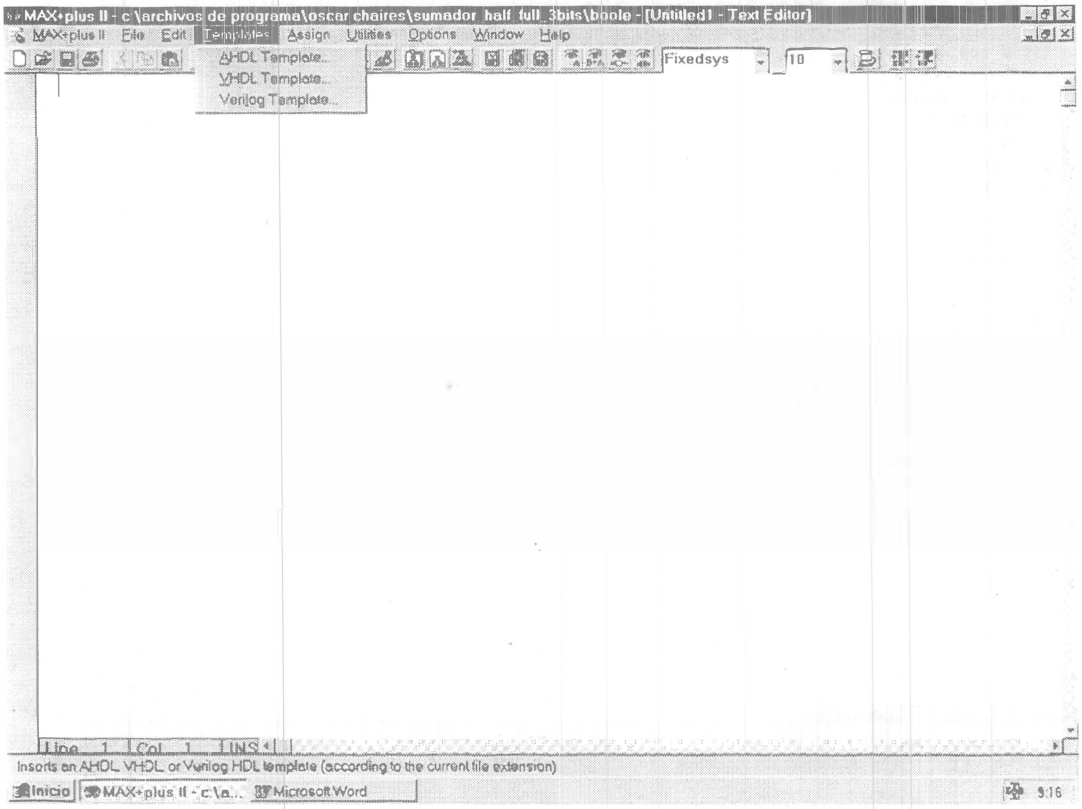


Fig 23 Templates para los lenguajes AHDL, Verilog y VHDL

Ejemplo 2

Si lo que deseamos introducir son unas funciones booleanas tal como F2= a0a1, y F1 = a0a1+b, lo primero que tenemos que hacer, ya estando en el editor de texto, es seleccionar el lenguaje a usar, y después escribir el titulo, datos y el diseño lógico El segundo paso a seguir será poner nombre a este proyecto Haciendo click en "file-project-name" Después volver a hacer un click en "file-set-project-to-current-file" Todo este proceso se ilustra en la Fig 24

Después de capturar el diseño el lenguaje de descripción de hardware elegido, el proceso de compilación, asignación de chip, asignación de pines, simulación y grabación es idénticamente igual que en el proceso hecho utilizando el editor gráfico Es decir aquí solo sustituimos la forma de captura utilizando el editor gráfico, por la forma de captura utilizando un lenguaje de descripción de hardware y por tanto el editor de texto

Nota: En manuales de la misma serie se estudia con detalle los lenguajes AHDL, Verilog HDL y VHDL

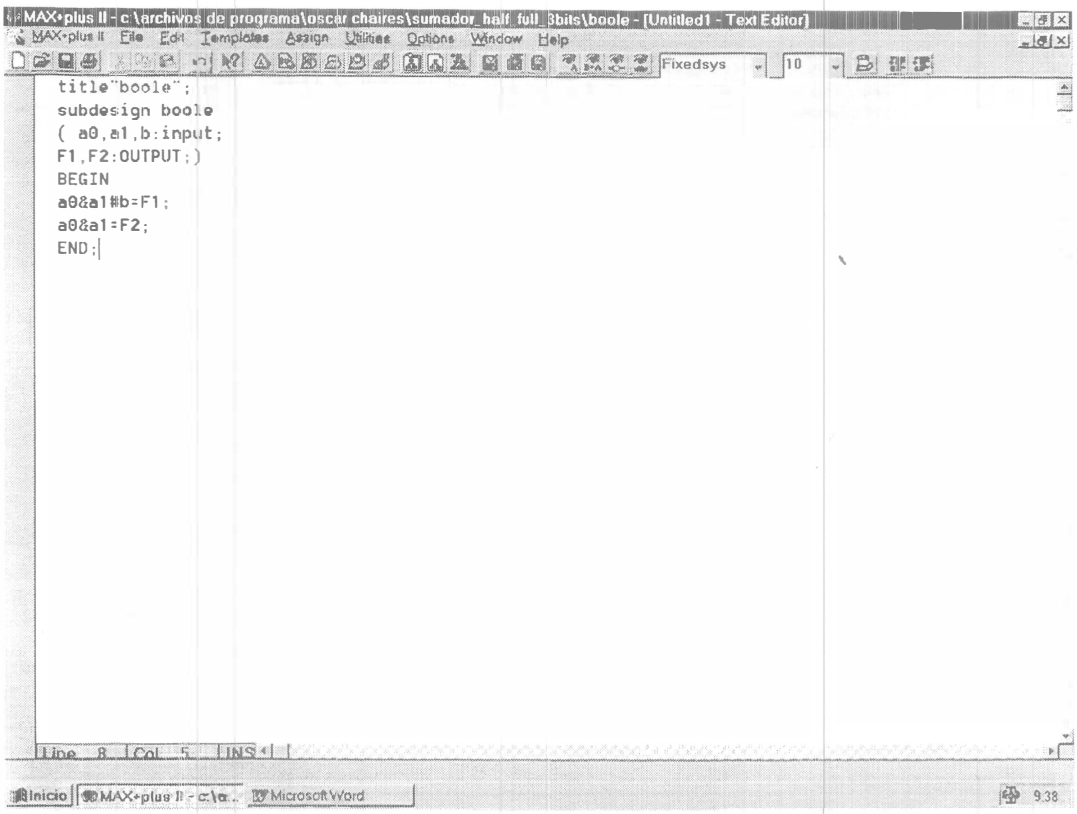


Fig 24 Captura del diseño del Ejemplo 2 usando los templates de AHDL

7 TARJETA DE DESARROLLO UP1.

El paquete de diseño de laboratorio de programa universitario de Altera, fue diseñado como una herramienta para el desarrollo de practicas de laboratorio utilizando la tecnología de Dispositivos Lógicos Programables Complejos (CPLD's)

La tarjeta de desarrollo UP1, ver Fig 25, tiene dos dispositivos CPLD's, el primero es el EPM7128S de 84 pines en encapsulado PLCC, el segundo es el EPF10K20 de 240 pines en encapsulado RQPF La tarjeta UP1 puede ser utilizada desde la versión 7 21 del software Max+Plus II

7.1 Descripción de la tarjeta UP1

La UP1 es una tarjeta para experimentar en el laboratorio y está basada en dos familias diferentes de CPLD's de Altera Max 7000 y Flex 10K

Dispositivo EMP7128S

En un dispositivo de mediana densidad dentro de la familia Max 7000S, se basa en tecnología EEPROM Tiene capacidad para 128 macroceldas sumando en total una capacidad aproximada de 2500 compuertas

Dispositivo EPF10K20

Es un dispositivo de alta integración perteneciente a la familia Flex 10K, está basado en tecnología SRAM Contiene 1152 elementos lógicos y seis bloques de memoria Cada

elemento lógico es una Look-up table de 4 entradas Cada bloque de memoria contiene 2048 bits Posee aproximadamente 20000 compuertas lógicas
La Fig 25 ilustra el diagrama a bloques de la tarjeta UP1, la cual será descrita más adelante

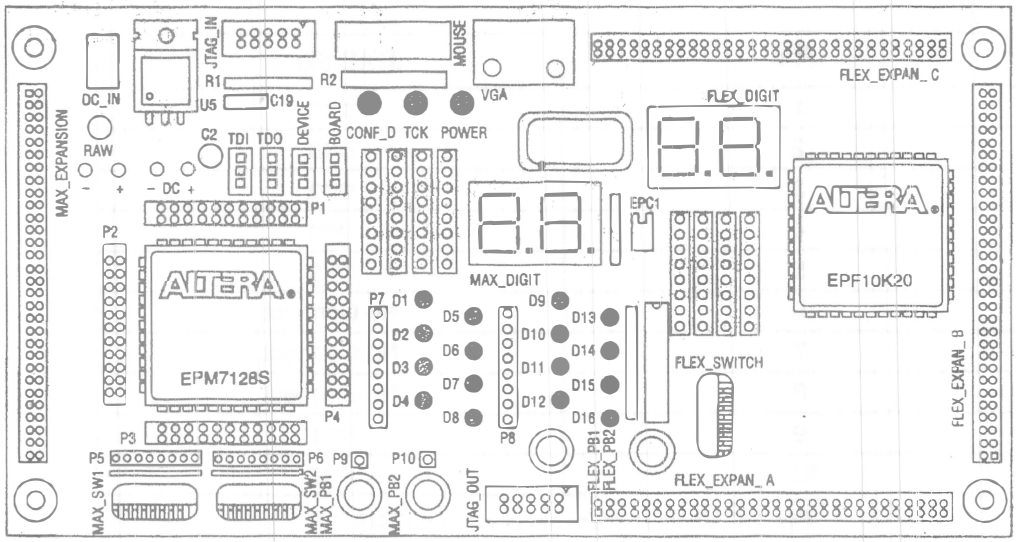


Fig 25 Diagrama a bloques de la tarjeta de desarrollo UP1

FACULTAD DE INGENIERIA

DC_IN & RAW Power Input

Este conector es el de alimentación, acepta entradas de DC de 7 hasta 12 volts con un mínimo de 250 mA El conector RAW consiste de dos orificios para conectar una fuente no regulada La UP1 trae un regulador interno para asegurar la alimentación de 5 V a los dispositivos Altera

Oscilador

La UP1 contiene un oscilador de cristal de 25 175 MHz La salida de este oscilador es una señal TTL que maneja las entradas de reloj global del EPM7128S (pin 83) y del EPF10K20 (pin 91)

JTAG_IN Header

Es el conector donde se conecta el Byte Blaster que se utiliza para transferir la información para la grabación del dispositivo desde el puerto paralelo hasta la UP1 La nomenclatura de este conector es ilustrada en la Fig 26

Jumpers.

G.- 612500

La UP1 contiene cuatro jumpers de tres pines cada uno (TDI, TDO, DEVICE y BOARD) los cuales pueden ser configurados por ejemplo para trabajar con el EPM7128S o el EPF10K20 o ambos, entre otras posibilidades La Fig 27 muestra las posiciones de los tres jumpers La Fig 28 muestra la configuración de los jumpers para una acción deseada

Pin	JTAG Signal
1	TCK
2	GND
3	TDO
4	VCC
5	TMS
6	No Connect
7	No Connect
8	No Connect
9	TDI
10	GND

Fig 26 Nomenclatura del conector JTAG_IN

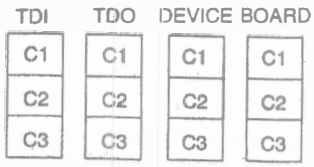


Fig 27 Posiciones de los tres jumpers

Desired Action	TDI	TDO	DEVICE	BOARD
Program EPM7128S device only	C1 & C2	C1 & C2	C1 & C2	C1 & C2
Configure EPF10K20 device only	C2 & C3	C2 & C3	C1 & C2	C1 & C2
Program/configure both devices, *Note (1)*	C2 & C3	C1 & C2	C2 & C3	C1 & C2
Connect multiple boards together, *Note (2)*	C2 & C3	OPEN	C2 & C3	C2 & C3

Fig 28 Acción según la posición de los jumpers

7.2 Uso del dispositivo EMP7128S

La UP1 provee los siguientes recursos para el dispositivo EPM7128S

- Socket para montar un dispositivo PLCC de 84 pines
- Pines de señales que son accesibles vía conectores hembras
- Conexión JTAG para el byte blaster
- Dos push-button
- Dos dipswitches de 8 bits cada uno
- 16 leds
- Dos displays de 7 segmentos
- Un oscilador de cristal de 25 175 MHz
- Un puerto de expansión con 42 pines de E/S y los pines dedicados globales CLR, OE1, OE2 y GCLK2

Conectores alrededor del EPM7128S

Estos conectores rodean al dispositivo EPM7128S y proveen acceso a todos los pines del chip La distribución de estos pines se detalla en la Fig 29, según la nomenclatura de la Fig 25

P1		P2		P3		P4	
Outside	Inside	Outside	Inside	Outside	Inside	Outside	Inside
75	76	12	13	33	34	54	55
77	78	14	15	35	36	56	57
79	80	16	17	37	38	58	59
81	82	18	19	39	40	60	61
83	84	20	21	41	42	62	63
1	2	22	23	43	44	64	65
3	4	24	25	45	46	66	67
5	6	26	27	47	48	68	69
7	8	28	29	49	50	70	71
9	10	30	31	51	52	72	73
11	X	32	X	53	X	74	X

Fig 29 Conectores alrededor del dispositivo EMP7128S

Push-Buttons MAX_PB1 & MAX_PB2

Estos son dos push buttons que proveen señales activas bajas, y son puestas arriba a través de resistores de 10 KΩ Las conexiones a estas señales se pueden hacer insertando cables en los orificios en la cercanía de los push buttons

Switches MAX_SW1 & MAX_SW2

Cada uno de ellos contiene ocho switches que proporcionan señales lógicas Estos switches son puestos arriba con resistencias de 10 KΩ Conexiones a estas señales son fácilmente hechas insertando cables en los orificios en las cercanías de los switches, y el otro lado del cable en el pin deseado del EPM7128S

LEDs D1 hasta D16

Son 16 LEDs que son puestos arriba a través de resistencias de 330 Ω Un LED es iluminado cuando un "0" lógico es aplicado al conector hembra asociado con el LED Las posiciones de la numeración de los LEDs se ilustra en la Fig 30

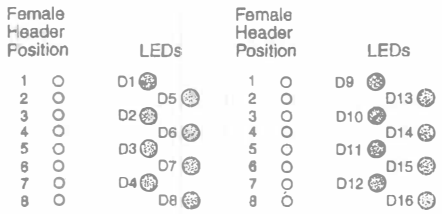


Fig 30 Posición de los LEDs

MAX_DIGIT Display

Es un display dual de siete segmentos conectado directamente a pines del dispositivo EPM7128S Cada segmento del display puede ser iluminado manipulando el pin respectivo al que está conectado en el dispositivo En la Fig 31 se ilustran los displays y en la Fig 32 las asignaciones entre segmentos del dígito y los pines del dispositivo

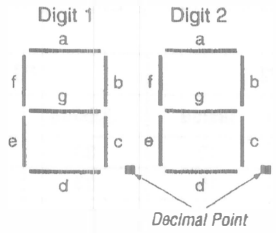


Fig 31 Nombres de los segmentos en los diplays

Display Segment	Pin for Digit 1	Pin for Digit 2
a	58	69
b	60	70
c	61	73
d	63	74
e	64	76
f	65	75
g	67	77
Decimal point	68	79

Fig 32 Conexiones de los pines del dispositivo hacia los displays

MAX_EXPANSION

Es un renglón dual de agujeros para acceder a las señales de E/S, señales globales, Vcc y Gnd del dispositivo EPM7128S La Fig 33 muestra la convención de números para los agujeros La Fig 34 lista los nombres de las señales contra los pines conectados a cada agujero

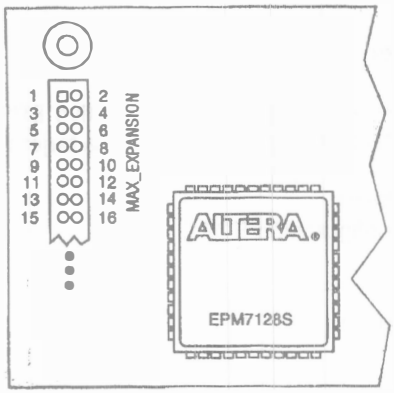


Fig 32 Convención de números para los agujeros

Hole Number	Signal/Pin	Hole Number	Signal/Pin
1	RAW	2	GND
3	VCC	4	GND
5	VCC	6	GND
7	No Connect	8	No Connect
9	No Connect	10	No Connect
11	No Connect	12	GCLRn/1
13	OE1/84	14	OE2/GCLK2/2
15	4	16	5
17	6	18	8
19	9	20	10
21	11	22	12
23	15	24	16
25	17	26	18
27	20	28	21
29	22	30	24
31	25	32	27
33	28	34	29
35	30	36	31
37	33	38	34

Hole Number	Signal/Pin	Hole Number	Signal/Pin
39	35	40	36
41	37	42	39
43	40	44	41
45	44	46	45
47	46	48	48
49	49	50	50
51	51	52	52
53	54	54	55
55	56	56	57
57	VCC	58	GND
59	VCC	60	GND

Fig 33 Nombres de las señales contra los pines conectados en cada agujero

7.3 Uso del dispositivo EPF10K20

La UP1 proporciona los siguientes recursos para el dispositivo EPF10K20 Los pines de este dispositivo están preasignados a los switches y LEDs sobre la tarjeta Estos recursos son

- Conexión JTAG para el ByteBlaster
- Socket para una EPROM de configuración ECP1
- Dos switches push buttons
- Un dipswitch octal
- Un display dual de siete segmentos
- Un oscilador de 25 175 MHz
- Puerto VGA
- Puerto para mouse
- Tres puertos de expansión, cada uno con 42 pines de I/O y siete pines globales

Push buttons FLEX_PB1 & FLEX_PB2

Estos son dos push buttons que proveen señales activas bajas, y son puestas arriba a través de resistores de 10 KΩ Las conexiones a estas señales para el FLEX_PB1 está conectada al pin 28 del dispositivo, y para el FLEX_PB2 está conectada al pin 29 del dispositivo

Switches FLEX_SW1.

Contiene ocho switches que proporcionan señales lógicas en "1" cuando el switch está abierto y "0" cuando el switch está cerrado La Fig 34 ilustra la asignación de pines a cada switch

Switch	EPF10K20 Pin
FLEX_SWITCH-1	41
FLEX_SWITCH-2	40
FLEX_SWITCH-3	39
FLEX_SWITCH-4	38
FLEX_SWITCH-5	36
FLEX_SWITCH-6	35
FLEX_SWITCH-7	34
FLEX_SWITCH-8	33

Fig 34 Asignación de pines para los switches FLEX_SW1

Display FLEX_DIGIT

Es un display dual de siete segmentos conectado directamente al dispositivo EPF10K20 Cada segmento puede ser iluminado poniendo un "0" lógico en el pin correspondiente del dispositivo La Fig 35 ilustra la asignación de pines para cada display

Display Segment	Pin for Digit 1	Pin for Digit 2
a	6	17
b	7	18
c	8	19
d	9	20
e	11	21
f	12	23
g	13	24
Decimal point	14	25

Fig 35 Asignación de pines para los displays de siete segmentos

FLEX_EXPAN_A, FLEX_EXPAN_B, FLEX_EXPAN_C.

Son columnas duales de agujeros para acceso a las señales de E/S, señales globales, de Vcc y Gnd de los pines del dispositivo EPF10K20 En la Fig 36 se muestra la convención de números para estos agujeros En la Fig 37, 38 y 39 se ilustran los pines conectados a cada agujero

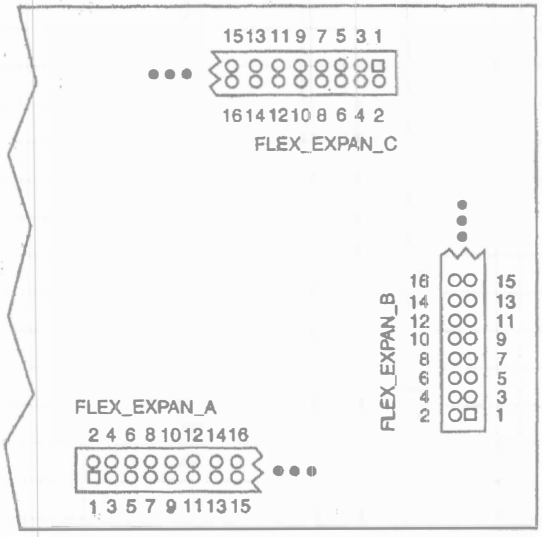


Fig 36 Convención de números para los agujeros de los conectores

Hole Number	Signal/Pin	Hole Number	Signal/Pin
1	RAW	2	GND
3	VCC	4	GND
5	VCC	6	GND
7	No Connect	8	DI1/99
9	DI2/92	10	DI3/210
11	DI4/212	12	DEV_CLR/209
13	DEV_OE/213	14	DEV_CLK2/211
15	45	16	46
17	48	18	49
19	50	20	51
21	53	22	54

Hole Number	Signal/Pin	Hole Number	Signal/Pin
23	55	24	56
25	61	26	62
27	63	28	64
29	65	30	66
31	67	32	68
33	70	34	71
35	72	36	73
37	74	38	75
39	76	40	78
41	79	42	80
43	81	44	82
45	83	46	84
47	86	48	87
49	88	50	94
51	95	52	97
53	98	54	99
55	100	56	101
57	VCC	58	GND
59	VCC	60	GND

Fig 37 Nombres de las señales contra los pines conectados en cada agujero del FLEX_EXPAN_A

Hole Number	Signal/Pin	Hole Number	Signal/Pin
1	RAW	2	GND
3	VCC	4	GND
5	VCC	6	GND
7	No Connect	8	DI1/99
9	DI2/92	10	DI3/210
11	DI4/212	12	DEV_CLR/209
13	DEV_OE/213	14	DEV_CLK2/211
15	109	16	110
17	111	18	113
19	114	20	115
21	116	22	117
23	118	24	119
25	120	26	126

Hole Number	Signal/Pin	Hole Number	Signal/Pin
27	127	28	128
29	129	30	131
31	132	32	133
33	134	34	136
35	137	36	138
37	139	38	141
39	142	40	143
41	144	42	146
43	147	44	148
45	149	46	151
47	152	48	153
49	154	50	156
51	157	52	158
53	159	54	161
55	162	56	163
57	VCC	58	GND
59	VCC	60	GND

Fig 38 Nombres de las señales contra los pines conectados en cada agujero del FLEX_EXPAN_B

Hole Number	Signal/Pin	Hole Number	Signal/Pin
1	RAW	2	GND
3	VCC	4	GND
5	VCC	6	GND
7	No Connect	8	DI1/99
9	DI2/92	10	DI3/210
11	DI4/212	12	DEV_CLR/209
13	DEV_OE/213	14	DEV_CLK2/211
15	175	16	181
17	182	18	183
19	184	20	185
21	186	22	187
23	188	24	190
25	191	26	192
27	193	28	194
29	195	30	196

Hole Number	Signal/Pin	Hole Number	Signal/Pin
31	198	32	198
33	200	34	201
35	202	36	203
37	204	38	206
39	207	40	208
41	214	42	215
43	217	44	218
45	219	46	220
47	221	48	222
49	223	50	225
51	226	52	227
53	228	54	229
55	230	56	231
57	VCC	58	GND
59	VCC	60	GND

Fig 39 Nombres de las señales contra los pines conectados en cada agujero del FLEX_EXPAN_C

8. TARJETA DE DESARROLLO UPx.

La ***UPx*** es una plaqueta experimental de muy bajo costo, para ser usada en las tareas de enseñanza sobre lógica programable En conjunto con el software para PC de ***ALTERA*** versión estudiantil ***MAX+PLUS II*** provee todas los recursos necesarios para crear y verificar diseños digitales de complejidad media

Características principales

- Apta para desarrollos con las siguientes EPLDs de ALTERA usando el **EPM7128SLC84/7128ELC84** 128 macroceldas en encapsulado PLCC de 84 patas
- Con un regulador de voltaje incorporado de 1 Ampere
- Con un generador de clock incorporado de 8MHz, que puede ser deshabilitado si así se desea
- Con un ***ByteBlaster*** incorporado, que sirve tanto para programar la 7032S, 7064S y la 7128S de la plaqueta, como para programar dispositivos colocados en otras plaquetas
- Con dos conectores de expansión que permiten tener acceso a todas las patas de los chips dentro de la plaqueta
- Con conexión directa a la PC a través de un cable de impresora paralela estándar tipo CENTRONICS

Sobre la EPM7128SLC84

La EPM7128SLC84 es un dispositivo lógico programable basado en términos producto, con programación tipo EEPROM Posee 128 macroceldas internamente agrupadas en 8 bloques (LABs) de 16 macroceldas cada uno, y una matriz de ruteado (PIA) que permite la plena interconexión del dispositivo Del total, 64 macroceldas están directamente asociadas a 64 patas del chip, para conectarse con el mundo externo, las restantes son útiles para la síntesis de lógica intermedia

La EPM7128SLC84 es un miembro de la familia MAX7000S, con una complejidad equivalente a 2 500 compuertas Lo revolucionario de estos dispositivos es que además de poder ser programados mediante el programador convencional de ALTERA, también pueden serlo mediante una interfase tipo JTAG (de ***Joint Test Action Group***), usando 4 patas dedicadas y sin necesidad de ser removidas del producto terminado esta característica se denomina ISP (por ***In-System-Programming***)

Sobre la interfase *ByteBlaster*

Para la programación de la EPM7128SLC84 mediante la interfase tipo JTAG, ALTERA ha desarrollado un circuito denominado ***ByteBlaster***, que por un lado se interconecta con la plaqueta donde está el dispositivo a programar a través de un conector de 10 pines denominado ISP Conector, y por el otro a la boca de impresora paralela de la PC donde se esté ejecutando el programa MAX+PLUS II

La plaqueta UPx ya incluye dentro de sí a los circuitos propios del ***ByteBlaster***, por lo que para programar sus EPLDs sólo requiere que el cable de impresora paralela de la PC sea conectado al conector Centronics de la UPx

Descripción de la UPx

La UPx se entrega en la forma de un KIT para armar, que NO contiene las EPLDs Este kit constituirá el diagrama a bloques ilustrado en la Fig 40

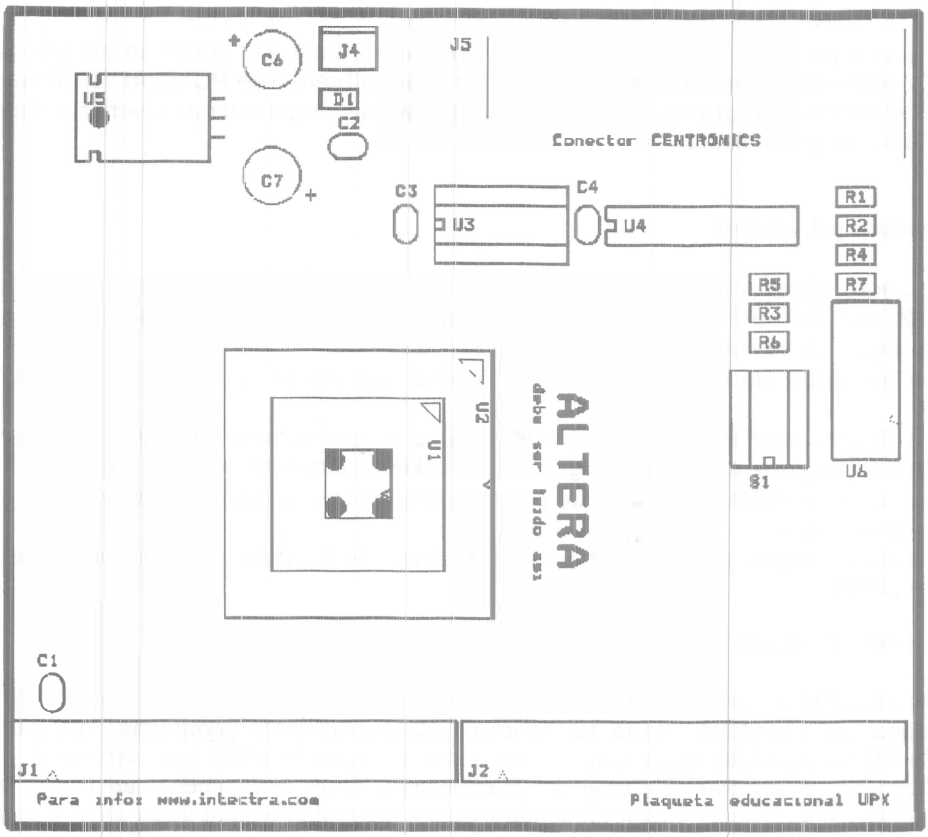


Fig 40 Diagrama a bloques de la tarjeta UPx

Sobre el conector Centronics (J5)

El conector CENTRONICS presente en la UPx permite su fácil conexión con la PC usando un cable estándar de impresora paralela Es importante notar que la **UPx NO TOMA** la energía que requiere para su operación de la PC sino **que requiere una fuente de alimentación externa** La distribución de los pines en el conector se detalla en la Fig 41

Conector Centronics	Función en la PC
2	DataBit 1
3	DataBit 2
7	DataBit 6
8	DataBit 7
9	DataBit 8
10	NACK
11	BUSY
12	PAPER END
13	SELECT
14	nAUTO_FEED
19-30,32	GROUND
1,4..6,15..18,31,33..36	sin conectar

Fig 41 Asignación de los pines del conector Centronics

Sobre el conector ISP (U6)

El conector ISP de la plaqueta UPx tiene por objeto permitir el uso de la UPx para programar plaquetas o circuitos externos a través de la interfase ISP (en otras palabras, usar a la UPx como ByteBlaster) En este caso los dip-switches 2 a 5 deben quedar abiertos para desconectar la EPLD residente en la UPx de la interfase ISP La distribución de los pines para este conector se detallan en la Fig 42

Pata	Señal JTAG
1	TCK
2	GND
3	TDO
4	Sin conectar (*)
5	TMS
6	Sin conectar
7	Sin conectar
8	Sin conectar
9	TDI
10	GND

Fig 42 Asignación de los pines del conector ISP

Sobre el conector de alimentación (J4)

El conector J4 es usado para suministrar la energía de operación a la UPx y a las eventuales plaquetas hija que se enchufen en los conectores de expansión Se detalla en la Fig 43

Pata	Función
1	GND
2	V+
3	GND

Fig 43 Asignación de los pines del conector de alimentación

La tensión de alimentación V+ debe ser mayor que +8 Volts y es recomendable que no supere los +12 Volts para evitar el excesivo calentamiento del regulador LM7805 (no es necesario que sea una tensión regulada) El consumo propio de la UPx operando a 8MHz, sin considerar el consumo adicional que pudieran generar plaquetas colocadas en los conectores de expansión, es menor a 100 miliamperes

Sobre los DIP-switches

La función de las 5 llaves tipo DIP es la siguiente

- **DSW1** cuando está ON conecta el Oscilador local de 8 MHz a la línea de clock de la EPLD local (GCK@43 en la 7064, GCK1@83 en la 7128S) y al pin 1 de los conectores para plaquetas hijas Cuando está abierto permite que el clock de la EPLD sea generado en esas plaquetas
- **DSW2** conecta la línea TDI/DATO del programador ISP/JTAG de la UPX a la línea TDI de la EPLD local (TDI@7 en la 7064, TDI@14 en la 7128S)
- **DSW3** conecta la línea TMS/nCNF del programador ISP/JTAG de la UPX a la línea TMS de la EPLD local (TMS@13 en la 7064, TMS@23 en la 7128S)
- **DSW4** conecta la línea TCK/DCLK del programador ISP/JTAG de la UPX a la línea TCK de la EPLD local (TCK@32 en la 7064, TCK@62 en la 7128S)
- **DSW5** conecta la línea TDO/DONE del programador ISP/JTAG de la UPX a la línea TDO de la EPLD local (TDO@38 en la 7064, TDO@71 en la 7128S)

Sobre los conectores de expansión

Los conectores J1 y J2 permiten conectar las patas de la 7128 a plaquetas hijas En la Fig 44 se ilustra la asignación de pines para el conector J1

IMPORTANTE: de usarse dispositivos sin ISP (familia MAX7000/7000E), o dispositivos con ISP pero con esta opción deshabilitada (MAX7000S), los pines indicados con (*) en las siguientes tablas quedan disponibles para su uso como líneas de entrada/salida de la EPLD En este caso los DIP Switches 2, 3, 4 y 5 deben ser abiertos para evitar que colisiones con los circuitos del ByteBlaster interno de la UPx

J1	Pin de 7128E/S	Función en la 7128E/S
1	83	GCK1
2	44	LabE-Macrocell 65
3	45	LabE-Macrocell 67
4	46	LabE-Macrocell 69
5	7,19,32,42	GND
6	47,59,72,82	GND
7	48	LabE-Macrocell 72
8	49	LabE-Macrocell 73
9	50	LabE-Macrocell 75
10	51	LabE-Macrocell 77
11	84	Input/OE1
12	2	Input/OE2/GCK2
13	52	LabE-Macrocell 80
14	1	Input/GCLR
15	73	LabH-Macrocell 115
16	74	LabH-Macrocell 117
17	75	LabH-Macrocell 118
18	76	LabH-Macrocell 120
19	77	LabH-Macrocell 123
20	79	LabH-Macrocell 125
21	3,13,26,38	VCC
22	43,53,66,78	VCC
23	80	LabH-Macrocell 126
24	81	LabH-Macrocell 128
25	33	LabD-Macrocell 64
26	34	LabD-Macrocell 61
27	35	LabD-Macrocell 59
28	36	LabD-Macrocell 57
29	37	LabD-Macrocell 56
30	39	LabD-Macrocell 53
31	40	LabD-Macrocell 51
32	41	LabD-Macrocell 49
33	12	LabA-Macrocell 3
34	11	LabA-Macrocell 5
35	10	LabA-Macrocell 6
36	9	LabA-Macrocell 8
37	8	LabA-Macrocell 11
38	6	LabA-Macrocell 13
39	5	LabA-Macrocell 14
40	4	LabA-Macrocell 16

Fig 44 Asignación de pines de J1 si la EPLD es una 7128E/S

J2	Pin de 7128E/S	Función en la 7128S	Función en la 7128E
1	83	GCK1	GCK1
2	20	LabB-Macrocell 21	LabB-Macrocell 21
3 (*)	14	TDI o LabB-Macrocell 32	LabB-Macrocell 32
4	21	LabB-Macrocell 19	LabB-Macrocell 19
5	7,19,32,42	GND	GND
6	47,59,72,82	GND	GND
7	17	LabB-Macrocell 25	LabB-Macrocell 25
8	15	LabB-Macrocell 29	LabB-Macrocell 29
9	18	LabB-Macrocell 24	LabB-Macrocell 24
10	16	LabB-Macrocell 27	LabB-Macrocell 27
11	84	Input/OE1	Input/OE1
12	2	Input/OE2/GCK2	Input/OE2/GCK2
13	22	LabB-Macrocell 17	LabB-Macrocell 17
14	1	Input/GCLR	Input/GCLR
15	68	LabG-Macrocell 105	LabG-Macrocell 105
16	70	LabG-Macrocell 109	LabG-Macrocell 109
17	69	LabG-Macrocell 107	LabG-Macrocell 107
18 (*)	71	TDO o LabG-Macrocell 112	LabG-Macrocell 112
19	67	LabG-Macrocell 104	LabG-Macrocell 104
20	65	LabG-Macrocell 101	LabG-Macrocell 101
21	3,13,26,38	VCC	VCC
22	43,53,66,78	VCC	VCC
23	64	LabG-Macrocell 99	LabG-Macrocell 99
24	63	LabG-Macrocell 97	LabG-Macrocell 97
25	61	LabF-Macrocell 94	LabF-Macrocell 94
26 (*)	62	TCK o LabF-Macrocell 96	LabF-Macrocell 96
27	60	LabF-Macrocell 93	LabF-Macrocell 93
28	54	LabF-Macrocell 83	LabF-Macrocell 83
29	57	LabF-Macrocell 88	LabF-Macrocell 88
30	56	LabF-Macrocell 86	LabF-Macrocell 86
31	58	LabF-Macrocell 91	LabF-Macrocell 91
32	55	LabF-Macrocell 85	LabF-Macrocell 85
33	30	LabC-Macrocell 37	LabC-Macrocell 37
34	31	LabC-Macrocell 35	LabC-Macrocell 35
35	28	LabC-Macrocell 40	LabC-Macrocell 40
36	29	LabC-Macrocell 38	LabC-Macrocell 38
37	25	LabC-Macrocell 45	LabC-Macrocell 45
38	27	LabC-Macrocell 43	LabC-Macrocell 43
39	24	LabC-Macrocell 46	LabC-Macrocell 46
40 (*)	23	TMS o LabC-Macrocell 48	LabC-Macrocell 48

Fig 45 Asignación de pines de J2 si la EPLD es una 7128E/S

9. BIBLIOGRAFÍA

A) Digital Design Principles and Practices John F Wakerly Ed Pretince Hall 2000
B) Logic and Computer Design Fundamentals Morris Mano Ed Pretince Hall 2000
C) Max+Plus II Getting Started Altera Corpotation 1997
D) Modern Digital Systems Design" John Y Cheung Ed West 1991
E) www.altera.com