

MÓDULO ANALIZADOR LÓGICO PARA LA TARJETA UP1 DE ALTERA

Antonio Cebrián¹, Rafael Gadea², José Manuel Blasco³

¹*Universidad Politécnica de Valencia, ETSIT, acebrian@eln.upv.es*

²*Universidad Politécnica de Valencia, ETSIT, rgadea@eln.upv.es*

³*Universidad Politécnica de Valencia, ETSIT, jobabluc@teleco.upv.es*

RESUMEN

La tarjeta de evaluación UP1 (*University Program 1*) de Altera se utiliza junto con el entorno de desarrollo integrado MAX+PLUS II en las prácticas de la asignatura Diseño de Circuitos y Sistemas Electrónicos impartida en la E.T.S.I de Telecomunicación de la Universidad Politécnica de Valencia.

Por un lado, el MAX+PLUS II integra todos los pasos de desarrollo para dispositivos programables de Altera: entrada de diseño, síntesis, simulación y programación. Por otro lado, la tarjeta de evaluación UP1 permite verificar el funcionamiento real del diseño realizado.

Para completar cada puesto de prácticas, se ha desarrollado un módulo analizador lógico para la tarjeta de evaluación UP1 que permite visualizar en el propio entorno MAX+PLUS II las entradas y salidas reales del diseño. Lo que permite al alumno enfrentarse a diseños que no habiendo presentado problemas de simulación no funcionan correctamente en la tarjeta de evaluación.

1. INTRODUCCIÓN

La tarjeta de evaluación UP1 (*University Program 1*) de Altera se utiliza junto con el entorno de desarrollo integrado MAX+PLUS II en las prácticas de la asignatura Diseño de Circuitos y Sistemas Electrónicos impartida en la E.T.S.I de Telecomunicación de la Universidad Politécnica de Valencia.

Por un lado, el MAX+PLUS II integra todos los pasos de desarrollo para dispositivos programables de Altera: entrada de diseño, síntesis, simulación y programación. Por otro lado, la tarjeta de evaluación UP1 permite verificar el funcionamiento real del diseño realizado.

Cada puesto de prácticas está dotado con un PC con el entorno de desarrollo integrado MAX+PLUS II y con una tarjeta de evaluación UP1. En cada práctica el alumno realiza el diseño propuesto, lo simula y pasa a comprobar su funcionamiento en la tarjeta de evaluación UP1.

El problema aparece cuando un diseño que no ha presentado problemas en la simulación, no funciona correctamente en la tarjeta de evaluación UP1. La mejor solución para poder abordar esta situación pasa por visualizar las entradas y salidas reales del diseño en la propia tarjeta de evaluación UP1. Para ello, sería necesario dotar a cada puesto de prácticas con un analizador lógico. Evidentemente, esta solución es económicamente inviable.

Buscando una solución de compromiso, se ha desarrollado un módulo analizador lógico de bajo coste para la tarjeta de evaluación UP1 que se ajusta a las necesidades de las prácticas. El módulo analizador lógico desarrollado capturará las entradas y salidas del diseño durante su verificación y permitirá su importación desde MAX+PLUS II para su posterior visualización, estudio y localización del problema por parte del alumno.

2. ENTORNO DE DESARROLLO INTEGRADO MAX+PLUS II

El entorno de desarrollo integrado MAX+PLUS II para dispositivos lógicos programables de Altera incluye un completo flujo de diseño y un entorno gráfico de trabajo intuitivo [1].

El flujo de diseño (figura 1) del MAX+PLUS II está dividido en cuatro fases claramente diferenciadas: entrada del diseño, compilación, verificación (simulación) y programación del dispositivo programable.

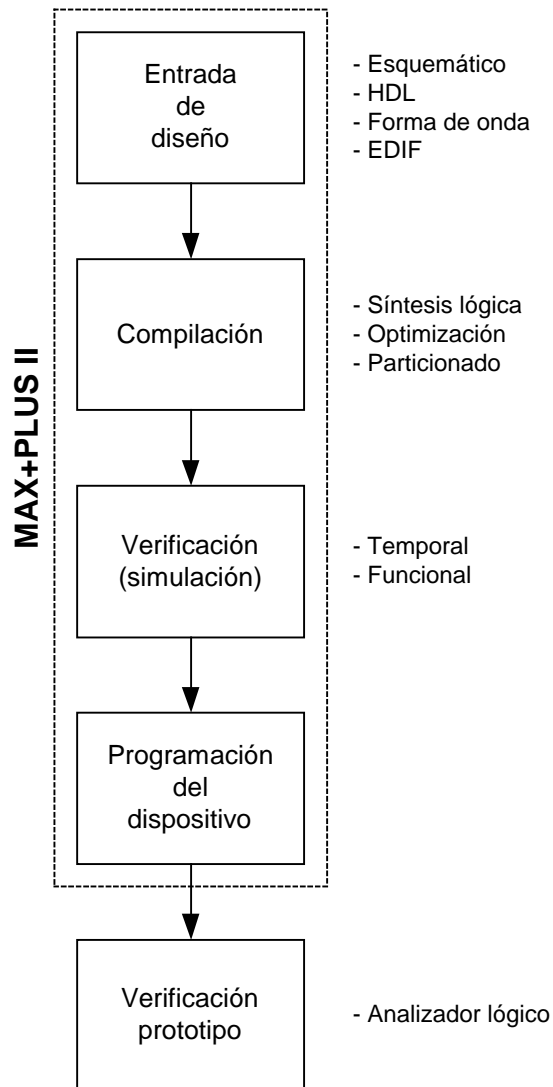


Figura 1: Flujo de diseño con MAX+PLUS II

- Entrada de diseño: Constituida por la introducción del diseño (parte lógica) y por la asignación y configuración del dispositivo (parte física).

La introducción del diseño se puede realizar mediante diversos formatos:

- Esquemático: mediante un editor gráfico integrado que se complementa con una serie de librerías con las primitivas y las funciones estándar más usuales (*primitive symbols*, *LPM functions*, *MegaCore functions*).
- Lenguaje de descripción hardware: además de permitirse la entrada de diseño mediante el uso del lenguaje de programación hardware de Altera (AHDL), también es posible el uso de los lenguajes estándar Verilog y VHDL.

- Forma de onda: mediante un editor gráfico integrado de formas de onda que permite crear archivos de entrada de diseño así como vectores de entrada para la simulación.
- Estándar EDIF: es posible incorporar partes del diseño realizadas con otras herramientas de desarrollo gracias al soporte del formato estándar EDIF, soportado por compañías como Cadence, Mentor Graphics y Viewlogic.

La asignación y configuración del dispositivo permite seleccionar cualquiera de los dispositivos programables de Altera y hacer la asignación de pines de nuestro diseño sobre el dispositivo seleccionado.

- **Compilación:** Cuando la entrada de diseño se ha completado, es necesario compilar el diseño. En el proceso de compilación se realiza en primer lugar la síntesis lógica del diseño optimizándola para la arquitectura del dispositivo programable utilizado. Tras la síntesis lógica, se realiza el particionado del diseño en uno o varios dispositivos programables (si el diseño completo no cabe en un solo dispositivo programable). En último lugar se generan los archivos de análisis temporal y de simulación del diseño así como los archivos de programación del dispositivo programable.
- **Verificación:** Los archivos de análisis temporal y de simulación del diseño obtenidos en la fase de compilación, permiten la verificación temporal y funcional del diseño. El análisis temporal permite determinar los requerimientos en los tiempos de *setup* y de *hold* de las señales así como la máxima frecuencia del reloj soportada por nuestro diseño. También es posible determinar el tiempo de propagación entre dos puntos del dispositivo programable y localizar los caminos críticos de nuestro diseño.
Mediante el uso del simulador integrado y con la ayuda del editor de formas de onda, es posible simular la respuesta del diseño ante una serie de estímulos de entrada generada por el usuario, de esta forma es posible verificar el correcto funcionamiento del diseño antes de la programación física del dispositivo programable.
- **Programación del dispositivo:** Tras la verificación del diseño en la cual se ha comprobado su correcto funcionamiento es necesario transferir el diseño al dispositivo programable. El proceso se realiza mediante el programador de dispositivos programables integrado, el cual permite programar, verificar y borrar los dispositivos programables utilizados.
En nuestro caso la programación del dispositivo programable se realizará sobre la tarjeta de evaluación UP1 que será considerada como el prototipo hardware del diseño realizado y que tendrá por tanto que ser verificado.

3. TARJETA DE EVALUACIÓN UP1

La tarjeta de evaluación UP1 de Altera (figura 2) incorpora dos dispositivos programables de las familias FLEX10K y MAX7000, respectivamente [2]. Además dispone de una serie de elementos de entrada y salida como pulsadores, microinterruptores, leds y visualizadores de 7 segmentos que pueden ser utilizados para generar las entradas y salidas del diseño realizado.

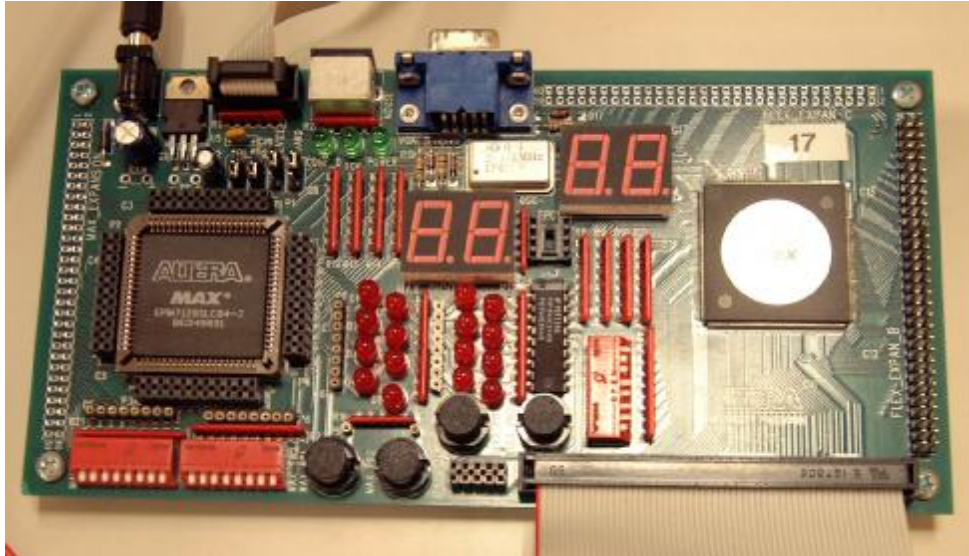


Figura 2: Tarjeta de evaluación UP1

La tarjeta de evaluación UP1 dispone de cuatro conectores de expansión de 60 pines: MAX_EXPANSION, FLEX_EXAN_A, FLEX_EXAN_B y FLEX_EXAN_C. Dichos conectores permiten acceder a los pines de entrada/salida de los dispositivos programables de las familias MAX7000 (el primero) y FLEX10K (los tres restantes), así como a las señales globales, la alimentación y la masa de la tarjeta de evaluación UP1.

Estos conectores permiten el uso de módulos de expansión de la tarjeta de evaluación UP1 que la doten de nuevas capacidades como es el caso del analizador lógico desarrollado.

4. MÓDULO ANALIZADOR LÓGICO

Se ha desarrollado un módulo analizador lógico de estados de bajo coste para la tarjeta de evaluación UP1 que permite realizar la fase de verificación del diseño sobre el prototipo real (en nuestro caso la tarjeta de evaluación UP1). El módulo analizador lógico se ha ajustado a las necesidades de las prácticas convirtiéndolo en un sustituto de bajo coste del analizador lógico de laboratorio. Es posible muestrear 16 entradas digitales con una frecuencia de muestreo de 2 MHz y con capacidad para almacenar 10000 estados (transiciones simples o múltiples de las señales de entrada).

Para facilitar la integración con la tarjeta de evaluación UP1 y su uso por parte del alumno, la tarjeta de circuito impreso del módulo analizador lógico desarrollado se ha diseñado como una tarjeta de expansión que se superpone a la tarjeta de evaluación UP1 utilizando el conector de expansión FLEX_EXPAN_A (figura 3). A través de este dicho conector de expansión, el módulo analizador lógico desarrollado toma la alimentación, la masa y las señales de entrada/salida utilizadas habitualmente en el desarrollo de las prácticas (las señales que interesa muestrear).

Gracias a la utilización de un conector *stackthrough* en el módulo analizador lógico, el conector de expansión de la tarjeta de evaluación UP1 sigue siendo accesible tras la conexión de dicho módulo. De esta forma se ha evitado que el módulo analizador lógico desarrollado limite de alguna forma la funcionalidad de la tarjeta de evaluación UP1.

El módulo analizador lógico desarrollado capturaré las entradas y salidas del diseño durante su verificación y permitirá su importación desde MAX+PLUS II para su posterior visualización por parte del alumno.

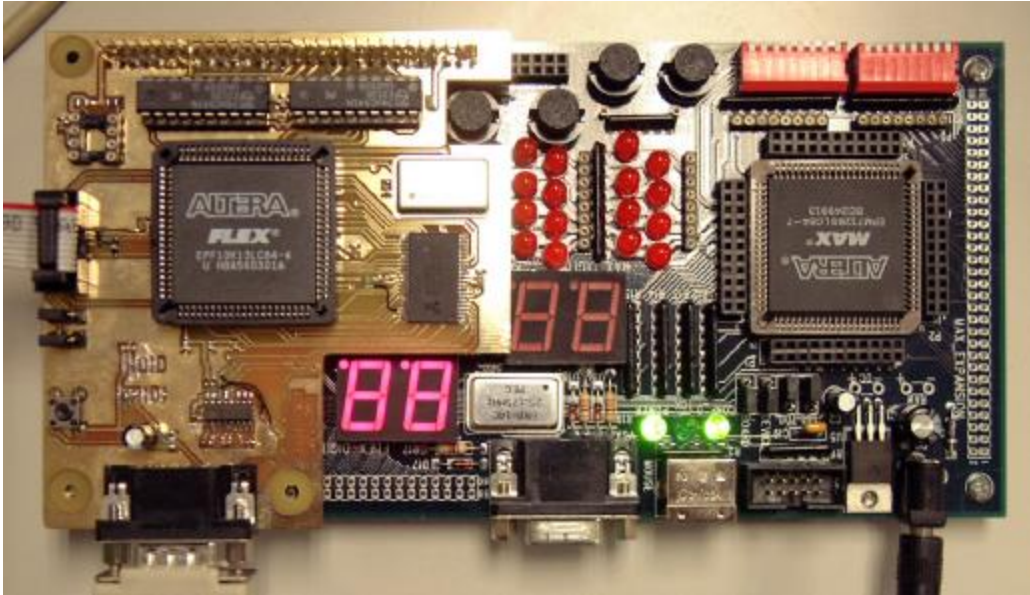


Figura 3: Módulo analizador lógico para tarjeta de evaluación UP1

4.1. Diagrama de bloques

El módulo analizador lógico (figura 4) está basado en un dispositivo programable (FPGA) de la familia FLEX10K de Altera que implementa el muestreo digital, la UART para la comunicación serie con el PC y el control de todo el sistema.

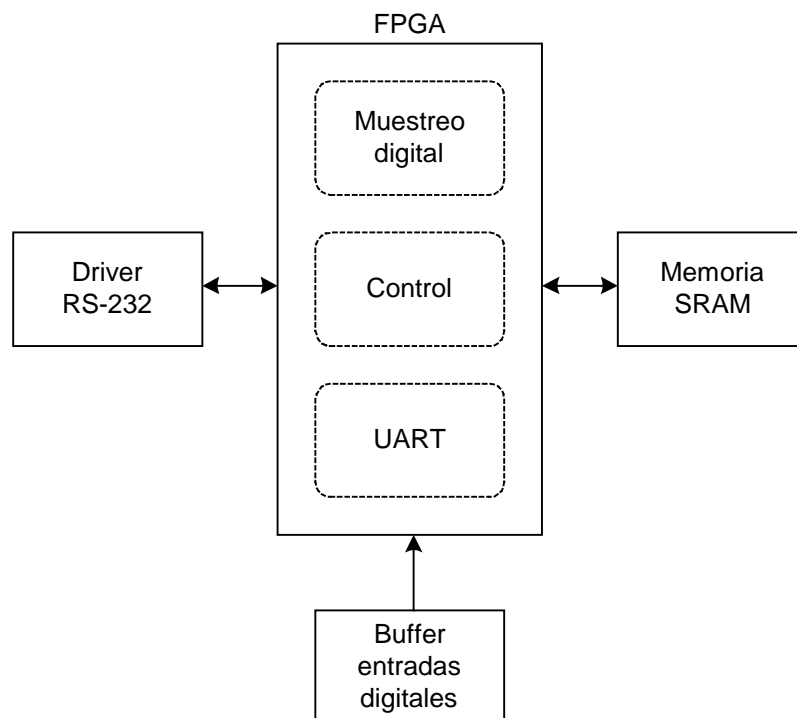


Figura 4: Diagrama de bloques del módulo analizador lógico

- Dispositivo programable (FPGA):
Dispositivo programable de la familia FLEX10K de Altera sobre el cual se han implementado los siguientes elementos:

- Muestreo digital: se ha implementado un analizador lógico de estados [3]. El analizador lógico implementado almacena el estado de las entradas digitales junto con su duración cada vez que se produce un cambio en las mismas. De esta forma la memoria de almacenamiento contiene una serie de parejas estado-duración que refleja la evolución temporal de las entradas digitales muestreadas. Dicha implementación del analizador lógico de estado se puede conseguir utilizando un sencillo algoritmo:
 1. Lee el estado actual (de las señales de entrada).
 2. Compara el estado actual con el estado anterior.
 3. Si se ha producido un cambio de estado:
 - 3.1. Almacena en la memoria de datos el estado anterior.
 - 3.2. Almacena en la memoria de datos el contador de tiempo transcurrido.
 - 3.3. Borra el contador de tiempo transcurrido.
 4. Actualiza el estado anterior con el estado actual.
 5. Incrementa el contador de tiempo transcurrido.
 6. Vuelve al paso 1.

- Control: máquina de estados encargada de controlar el muestreo digital así como la comunicación con el PC a través de la UART. Desde el PC se enviarán comandos para iniciar/parar el muestreo digital, para borrar/leer la memoria de almacenamiento, para configurar la máscara a aplicar a las entradas digitales y para configurar la condición de inicio del muestreo.

- UART: implementación de una UART para proporcionar una comunicación *full duplex* con el PC a 57600, 8, N, 1. Dicha comunicación permitirá el envío de comandos desde el PC así como la transmisión del contenido de la memoria de almacenamiento desde el analizador lógico desarrollado hasta el PC.

- Memoria de almacenamiento (SRAM):
 Memoria estática de 65,536 palabras de 16 bits utilizada para el almacenamiento de los datos del muestreo digital. Su contenido estará formado por una serie de parejas estado-duración que reflejarán la evolución temporal de las entradas digitales muestreadas.

- Buffer de entrada digital:
 Buffer de entrada utilizado para evitar que el dispositivo programable quede expuesto a señales de niveles incorrectos que pudieran causarle daños irreversibles.

- Driver RS-232:
 Encargado de la conversión de los niveles TTL de la UART a los niveles RS-232 utilizados en la conexión serie con el PC.

5. INTEGRACIÓN CON EL MAX+PLUS II

El analizador lógico para la tarjeta de evaluación UP1 utiliza una conexión serie con el PC. A través de la cual se reciben comandos y se transfiere el contenido de la memoria de almacenamiento. Para gestionar esta comunicación serie desde el PC se ha desarrollado una sencilla aplicación de control (figura 5) que permite, en primer lugar, el envío de los comandos de inicio y paro del muestreo digital y de la configuración de las máscaras de

muestreo e inicio del muestreo y en segundo lugar, la recepción de la memoria de almacenamiento del analizador lógico.

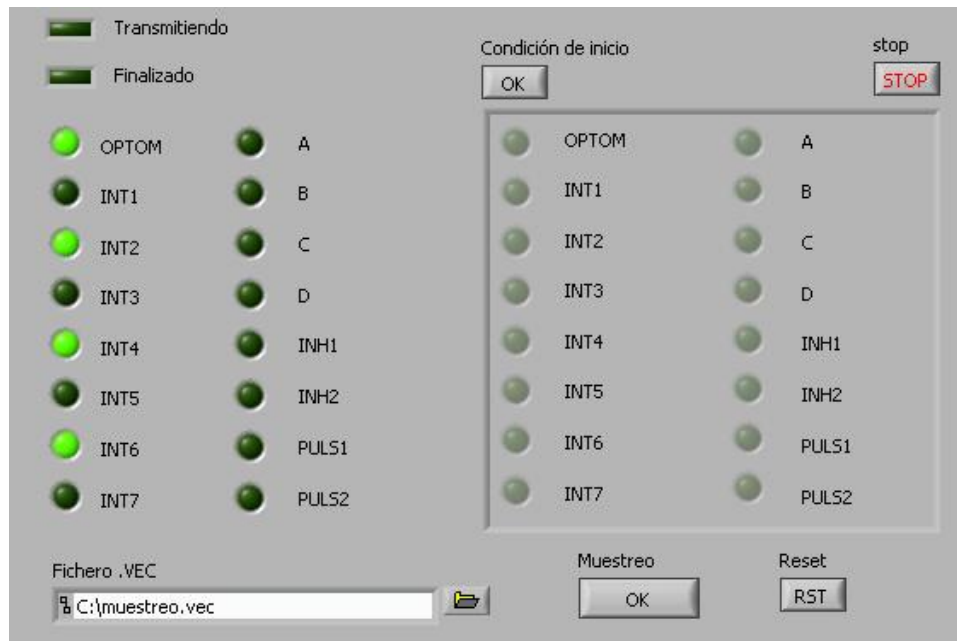


Figura 5: Aplicación de control

La memoria de almacenamiento contiene una serie de parejas estado-duración que reflejan la evolución temporal de las señales digitales de entrada muestreadas. Por tanto, el problema de la integración con el MAX+PLUS II se reduce a poder visualizar dicha serie de parejas estado-duración recibidas dentro del entorno de desarrollo MAX+PLUS II, concretamente utilizando el editor de formas de onda. Para ello, se ha utilizado el formato vector (.vec) reconocido por el MAX+PLUS II y que puede ser importado directamente desde el editor de formas de onda. Dicho formato está concebido para realizar entradas de estímulos en la simulación y en el test funcional y para realizar entradas de diseño a través del editor de formas de onda.

El archivo vector no es más que un archivo ASCII que determina mediante una serie de vectores el estado lógico de las señales así como el momento en el que se producen dichos estados. Se ha incluido un fragmento de un archivo vector en el cual se puede apreciar como, tras definir la unidad de tiempo utilizada y el nombre de las señales que componen los vectores, se describe el estado lógico de las señales y el momento en el que se producen dichos estados lógicos:

```
UNIT ns;
INPUTS OPTOM INT1 INT2 INT3 INT4 INT5 INT6 INT7 A B C D INH1 INH2 PULS1
PULS2;
PATTERN
000000000000> 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0
000000000434> 1 1 0 0 0 0 0 1 0 0 0 0 0 0 0 0
000000194432> 0 0 1 0 0 0 0 1 0 0 0 0 0 0 0 0
000000574182> 1 0 1 0 0 0 0 1 0 0 0 0 0 0 0 0
000000953932> 0 1 1 0 0 0 0 1 0 0 0 0 0 0 0 0
000001333682> 1 1 1 0 0 0 0 1 0 0 0 0 0 0 0 0
000001713432> 0 0 0 1 0 0 0 1 0 0 0 0 0 0 0 0
000002093182> 1 0 0 1 0 0 0 1 0 0 0 0 0 0 0 0
000002472932> 0 1 0 1 0 0 0 1 0 0 0 0 0 0 0 0
000002852682> 1 1 0 1 0 0 0 1 0 0 0 0 0 0 0 0
000003232432> 0 0 1 1 0 0 0 1 0 0 0 0 0 0 0 0
```

```

000003612182> 1 0 1 1 0 0 0 1 0 0 0 0 0 0 0 0
000003991932> 0 1 1 1 0 0 0 1 0 0 0 0 0 0 0 0
000004371682> 1 1 1 1 0 0 0 1 0 0 0 0 0 0 0 0
000004751432> 0 0 0 0 1 0 0 1 0 0 0 0 0 0 0 0
000005131182> 1 0 0 0 1 0 0 1 0 0 0 0 0 0 0 0
000005510932> 0 1 0 0 1 0 0 1 0 0 0 0 0 0 0 0
000005890682> 1 1 0 0 1 0 0 1 0 0 0 0 0 0 0 0
000006270432> 0 0 1 0 1 0 0 1 0 0 0 0 0 0 0 0
000006650182> 1 0 1 0 1 0 0 1 0 0 0 0 0 0 0 0
000007029932> 0 1 1 0 1 0 0 1 0 0 0 0 0 0 0 0
000007409682> 1 1 1 0 1 0 0 1 0 0 0 0 0 0 0 0
000007789432> 0 0 0 1 1 0 0 1 0 0 0 0 0 0 0 0
000008169182> 1 0 0 1 1 0 0 1 0 0 0 0 0 0 0 0
000008548932> 0 1 0 1 1 0 0 1 0 0 0 0 0 0 0 0
000008928682> 1 1 0 1 1 0 0 1 0 0 0 0 0 0 0 0
000009308432> 0 0 1 1 1 0 0 1 0 0 0 0 0 0 0 0
000009688182> 1 0 1 1 1 0 0 1 0 0 0 0 0 0 0 0
000010067932> 0 1 1 1 1 0 0 1 0 0 0 0 0 0 0 0
000010447682> 1 1 1 1 1 0 0 1 0 0 0 0 0 0 0 0
000010827432> 0 0 0 0 0 1 0 1 0 0 0 0 0 0 0 0
000011207182> 1 0 0 0 0 1 0 1 0 0 0 0 0 0 0 0
000011587366> 0 1 0 0 0 1 0 1 0 0 0 0 0 0 0 0
000011967116> 1 1 0 0 0 1 0 1 0 0 0 0 0 0 0 0
000012346866> 0 0 1 0 0 1 0 1 0 0 0 0 0 0 0 0
000012726616> 1 0 1 0 0 1 0 1 0 0 0 0 0 0 0 0
000013106366> 0 1 1 0 0 1 0 1 0 0 0 0 0 0 0 0
000013486116> 1 1 1 0 0 1 0 1 0 0 0 0 0 0 0 0
000013865866> 0 0 0 1 0 1 0 1 0 0 0 0 0 0 0 0
000015005116> 1 1 0 1 0 1 0 1 0 0 0 0 0 0 0 0
000015384866> 0 0 1 1 0 1 0 1 0 0 0 0 0 0 0 0
000015764616> 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
;

```

Tras importar el archivo vector desde el editor de formas de onda del MAX+PLUS II se consigue una representación gráfica mucho más amigable de los resultados obtenidos en el muestreo digital (figura 6).

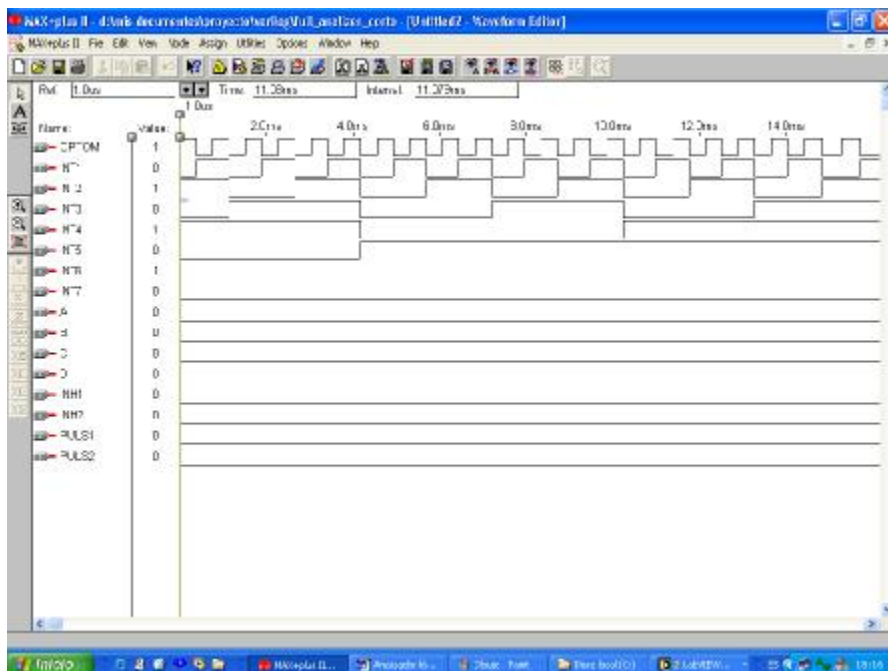


Figura 6. Archivo vector importado en el editor de formas de onda

6. CONCLUSIONES

Se ha desarrollado un módulo analizador lógico de estados de bajo coste para la tarjeta de evaluación UP1 de Altera utilizada en prácticas.

Se ha resuelto la integración del analizador lógico en el entorno de desarrollo MAX+PLUS II mediante el desarrollo de una aplicación de control que permite el gobierno del analizador lógico y la conversión de la memoria de muestreo en un formato importable desde el editor de formas de onda del MAX+PLUS II.

Se ha conseguido integrar todas las fases del flujo de diseño en un mismo entorno de trabajo enriqueciendo notablemente el contenido de las prácticas realizadas por el alumno con un coste reducido.

7. BIBLIOGRAFÍA

[1] Altera Corporation, "MAX+PLUS II Programmable Logic Development System & Software", A-DS-MPLUS2-08, January 1998

[2] Altera Corporation, "University Program UP1 Development Kit", A-UG-UP1-3.0, July 2003

[3] J.M. Grima, J.M. Andrés, R. Capilla, "ANALIZADORES LÓGICOS: ESTADOS, TIEMPOS, FIRMAS", SPUPV-93.028