

# Verificación Funcional de un Microcontrolador 8051 en Tecnología 3D

Osterman Niria, Di Federico Martín, Julián Pedro

*Departamento de Ingeniería Eléctrica y de Computadoras, Universidad Nacional del Sur,  
Bahía Blanca, Buenos Aires, Argentina.*

**Resumen**—En este informe se expone el desarrollo de un proyecto final de carrera el cual se realizó sobre la base de un sensor de imágenes fabricado en tecnología 3D sobre un proceso de 130 nm. El objetivo consistió en verificar el funcionamiento de un microcontrolador 8051 que se encuentra dentro del Imager y tiene la posibilidad de controlarlo en su totalidad.

**Index Terms**—Microcontrolador 8051, UART, validación.

## I. INTRODUCCIÓN

LA tecnología 3D interconecta varias obleas de silicio en forma vertical, utilizando conexiones metálicas entre ellas, o vías, cuyos diámetros son del orden de las centenas de micrómetros. La tecnología 3D posibilita incrementar la densidad de integración al permitir construir un chip múltiple con diferentes pisos, denominados "tiers".

Dentro de las ventajas de esta tecnología se pueden mencionar la mayor cantidad de transistores por unidad de volumen, la posibilidad de interconectar obleas de distintos tipos de tecnología y la reducción en las interconexiones, lo cual permite una disminución de las capacidades parásitas y un incremento en la velocidad de operación.

El Imager se encuentra fabricado en esta tecnología y está formado por un arreglo de 48x36 celdas, donde cada celda posee dos registros, un contador, un elemento fotosensor y un convertor A/D. Además de las celdas se tienen varias estructuras adicionales que realizan cálculos matemáticos y un microcontrolador 8051, que posibilita controlar el circuito completo.

El chip también cuenta con un puerto de comunicaciones RS232 para enviar señales hacia y desde el exterior.

En este trabajo se detallan los pasos seguidos para realizar la verificación del funcionamiento del microcontrolador 8051 dentro del procesador de imágenes. Comenzando por el diseño del microcontrolador implementado y del Imager 3D, luego, en la Sección III se explica la etapa de verificación y de la interfaz de comunicación serie. Por último, en la Sección IV se explica la validación del microcontrolador 8051 en el Chip 3D.

## II. DISEÑO

### II-A. Microcontrolador e Interfaz de Comunicación Serie

En la Fig. 1 se muestra el diagrama en bloques de la instancia que describe el comportamiento del circuito compuesto por el microcontrolador y la interfaz de comunicación serie. Se

muestran los puertos de entrada y de salida los cuales permiten la interacción del microcontrolador con el exterior.

Las señales de la interfaz de comunicación serie se mapearon a puertos del 8051. A continuación se describe su interconexión:

$INP \rightarrow P1\_out$   
 $EOR \rightarrow P0\_in(0)$   
 $OUTP \rightarrow P1\_in$   
 $EOT \rightarrow P0\_in(1)$   
 $READY \rightarrow P0\_out(2)$   
 $WR \rightarrow P0\_out(0)$



Figura 1. Diagrama en bloque de la instancia microcontrolador e interfaz de comunicación serie.

**II-A1. Diseño del Microcontrolador 8051:** El microcontrolador implementado cuenta con una pequeña memoria interna de 32 bytes y el programa se almacena en una memoria externa. El puntero de instrucciones se encuentra conectado al exterior por medio de 12 pines, logrando direccionar hasta 4 Kbytes de memoria de programa. Las conexiones con el Imager se realizan por medio de ocho puertos de ocho bits denominados  $p0-p7$ . Estos puertos son los últimos ocho bytes de la memoria interna del microcontrolador.

El 8051 permite tres tipos de direccionamiento: directo, indexado, inmediato y por registros específicos. Cuenta con un conjunto de 70 instrucciones.

El microcontrolador no maneja interrupciones y permite que el Imager pueda, agregando solo una memoria externa donde se almacena el programa, ejecutar todas las tareas necesarias para: captar la imagen, procesarla, trabajar en paralelo con otros Imagers y entregar información al exterior.

**II-A2. Interfaz de comunicación serie.:** La UART (Universal Asynchronous Receiver/Transmitter) implementada con-

tiene dos relojes independientes para el receptor y para el transmisor permitiendo que la comunicación sea del tipo full-duplex.

Sus puertos de entrada y de salida se conectan con los del microcontrolador 8051, permitiendo controlar la comunicación serie programando la memoria ROM externa del mismo.

## II-B. Diseño del Chip 3D

El Imager bajo el cual se realizó la verificación del microcontrolador está conformado por los bloques mostrados en la Fig. 2. El chip completo abarca un área de 2 mm. x 2,5 mm. y tiene 70 pads, con un voltaje de trabajo de 3,3 V. Se encuentra dividido en dos pisos. Todas las señales del exterior llegan al piso superior y se envían al piso inferior a través de conexiones internas.

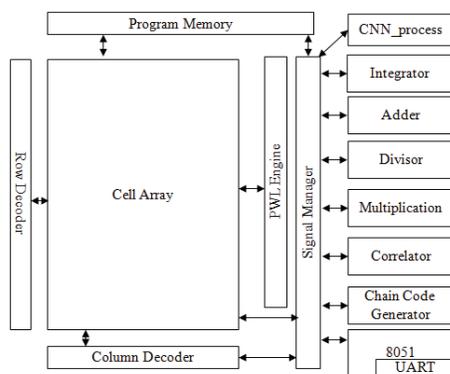


Figura 2. Diagrama en bloques del chip 3D.

Los bloques extras con los que cuenta el Imager están separados en dos grupos: en el piso inferior están colocados aquellos que tienen acceso directo a las celdas. Los que comparten información a través de un bus interno, están colocados en el piso superior. Este piso recibe todas las señales y la alimentación, muchas de las cuales son enviadas al piso inferior, en el cual se encuentra el microcontrolador.

El microcontrolador 8051 interactúa con el exterior a través de pads. Entre ellos se pueden mencionar: la señal *MICRORO* que controla si el microcontrolador posee el mando del Imager, lo cual sucede si se encuentra en nivel lógico alto. *CLK\_M* y *RST\_M* son la entrada de reloj al microcontrolador y su señal de reinicio respectivamente.

*RXD* y *TXD* corresponden a la entrada y salida de la UART del microcontrolador 8051 permitiendo tener comunicación hacia y desde el interior a través del puerto serie. Siete pines de entrada (*BUSH*), dan la posibilidad de ingresar un valor hacia el microcontrolador, si éste está seleccionado como el que controla el circuito. Además se tiene un bus de salida (*BUS\_OUT7 – BUS\_OUT0*), que según el valor de una señal de selección (*selBusOut*), permite que el bloque seleccionado lo escriba. Por último se encuentran las señales *ADDR*, *DATAIN* y *ROMRD* las cuales interactúan con la memoria de programa.

Todas las partes internas del circuito se conectan a un bus interno y tienen acceso de lectura y escritura. El bloque que

escribe en el bus interno es controlado por una señal de selección denominada *SelBusInt*. Los bloques que son enviados al bus de salida son controlados por la señal denominada *selBusOut*.

Como se dijo anteriormente, el microcontrolador toma el control del circuito completo en el momento en que la señal *MICRORO* se encuentra en nivel lógico alto. A partir de este momento, los ocho puertos de salida del 8051, *p0 – p7*, se conectan con todas las señales necesarias para controlar el chip y tener acceso a todos los registros internos. Dichas señales se muestran en la tabla I.

EL microcontrolador envía señales al chip a través de sus puertos los cuales se escriben realizando un programa en lenguaje ensamblador el cual se almacena en la memoria ROM.

Bits	7	6	5	4	3	2	1	0
p0	Bus Externo							
p1	COL Externo							
p2	ROW Externo							
p3	Init Proc	Init cnt	Init pwm	Init chain		Load1	Load2	Cbusb
p4	LX	LU	LP	CCLK	CRST	RST	CLKPR	Clk Pxl
p5	BorderU	BordeX	BordeX B	BordeX R	MASK	SelVecF		Fo Gin
p6		EnSel				Rst PxlB	Rst Sample	Sample
p7	SELBUS OUT				SELBUS INT			

Cuadro I  
SEÑALES DE CONTROL DEL IMAGER.

## III. VERIFICACIÓN

### III-A. Simulaciones

Las simulaciones se realizaron utilizando el software ModelSim PE Student Edition 10.1d. La memoria ROM se generó mediante un software emulador del microcontrolador 8051 llamado MCU 8051 IDE v 1.4.5. En éste se escribe el programa en lenguaje ensamblador, luego genera un archivo .hex el cual es tomado por un script realizado en Matlab que se encarga de producir un archivo VHDL correspondiente a la memoria.

En primer se realizó un programa que escribe valores en cada puerto de salida del microcontrolador y en cada dirección de memoria RAM. Se realizó la simulación y se verificó el correcto funcionamiento.

Luego se agregó la descripción de la interfaz serie, se realizó un programa que genera un eco y se simuló el correcto funcionamiento del conjunto.

## IV. VALIDACIÓN

La validación se dividió en dos etapas: en primer lugar se realizó la validación del microcontrolador en una FPGA Spartan 3 utilizando el sintetizador Design Suit 14.4 de Xilinx (versión WebEdition). Una vez asegurados del funcionamiento en la FPGA, se realizó la verificación del chip 3D utilizando una plaqueta que lo contiene y que consta de pines que dan la posibilidad de conectarlo a la FPGA.

Para realizar la validación, se sintetizó en la FPGA un microcontrolador Picoblaze y la memoria de programa. Todas las funciones realizadas en Matlab se escribieron enviando las directivas hacia el Picoblaze y éste hacia el microcontrolador 8051.

El Picoblaze tiene siete puertos de salida,  $PoF0 - PoF7$ , que se conectaron a pines del chip y dos puertos de salida  $PiF0$  y  $PiF1$ . El puerto  $PoF0$  se conecta a los pines de entrada  $BUSIN$  siempre que el microcontrolador 8051 no tenga el mando, en caso contrario, el  $BUSIN$  se conecta con el dato de salida de la memoria ROM. La señal  $MICRORO$  está conectada con el bit 7 del puerto  $PoF4$  del Picoblaze.

El puerto  $PoF1$  está conectado al  $BUSH$ . Este se utiliza para ingresar datos desde el exterior.

El puerto  $PoF2$  se conecta con los pines del chip denominados  $CLKs$ , estos son señales de habilitación para cargar datos en distintos registros.

El puerto  $PoF3$  está conectado a pines que se utilizan para escribir registros internos, como por ejemplo los registros que seleccionan quien escribe o lee en el bus interno o en el bus de salida. El puerto  $PoF3(0)$ , se encuentra conectado a la señal  $LOAD$ , se necesita un pulso de esta señal para cargar un valor externo en el  $BUSH$ .

Las señales hacia el microcontrolador son  $CLK_M$ ,  $RST_M$ ,  $RXD$ ,  $CLK_BRD$  y  $TXD$  y están conectadas a los puertos del Picoblaze  $PoF4(0)$ ,  $PoF4(1)$ ,  $PoF4(2)$ ,  $PoF4(3)$  y  $PiF1(0)$  respectivamente.

Como se dijo anteriormente, el Picoblaze se comunica con Matlab a través del puerto de comunicación serie. La memoria de programa del Picoblaze contiene todas las rutinas para testear el chip. Se utilizaron aquellos comandos que realizan la escritura de un puerto indicado o la lectura del bloque seleccionado en el bus de salida.

#### IV-1. Procedimiento realizado:

- **Selección de fila:** Se escribió un programa en assembler para la memoria ROM. El programa consistió en escribir valores de 00h a FFh en un registro del microcontrolador seleccionando la fila a mostrarse en el bus de salida.

Luego se generó el archivo VHDL de la ROM y se lo sintetizó junto con el microcontrolador Picoblaze.

La función realizada en Matlab envía el caracter Z y lo recibe para constatar que se encuentra establecida una comunicación. Envía al puerto  $PoF4$  del Picoblaze, un 128 (80h). Este valor activa la señal  $MICRORO$ , dándole el control al 8051. Luego se realiza un pulso de reloj hacia el microcontrolador y se establece en nivel lógico alto la señal de reinicio ( $RST$ ).

Ante un pulso en la señal  $RST$ , el puntero de instrucciones se sitúa en la dirección 00h de la memoria de programa. Desde Matlab se generan los pulsos de reloj necesarios para que se ejecuten todas las líneas de código escritas en la memoria ROM. Luego se envía un comando que permite leer el valor del bus de salida en relación al puerto seleccionado, en este caso se selecciona la fila. Por último se deshabilita el microcontrolador 8051.

- **Escribir dato en puerto del 8051 y enviarlo al bus de salida:** Se ingresa un valor en el  $BUSH$ , con el 8051 habilitado y luego se realiza un pulso en la señal de habilitación ( $LOAD$ ), el valor ingresa al puerto  $p2_{in}$  del microcontrolador.

El programa escrito en assembler, selecciona en el bus interno el registro que contiene el valor del puerto de entrada del 8051 ( $p2_{in}$ ) y en el bus de salida el registro

que contiene el puerto de salida del microcontrolador ( $p3_{out}$ ). El valor recibido desde el  $BUSH$  se almacena en el registro acumulador, y luego se envía por el puerto  $p3_{out}$  del microcontrolador.

Desde Matlab, se lee el valor del dato enviado en el bus de salida.

Una vez asegurado que es posible comunicar el exterior con el 8051, se procedió a realizar la verificación de sus instrucciones. Para cada instrucción a verificar se escribió un programa distinto en la ROM.

La rutina realizada en Matlab ejecuta la siguiente función: En el caso que se necesite ingresar un valor por el  $BUSH$  se habilita el Microcontrolador colocando en nivel lógico alto la señal  $MICRORO$ , enviando un 80h al puerto  $PoF4$ . Luego se generan pulsos de reloj ( $CLK_M$ ) y se levanta la señal de reset ( $RST_M$ ).

A continuación se escribe un valor en el puerto  $BUSH$  y se genera un pulso escribiendo en el puerto  $PoF3(0)$ , correspondiente a la señal  $LOAD$ . Estas líneas se omitieron en el caso que la instrucción a verificar no necesitara un valor externo. Luego se generan los pulsos de reloj para que se ejecute el programa en la memoria y se lee el valor del bus de salida, que en este caso contiene el puerto de salida  $p3_{out}$ .

Por último se deshabilita el microcontrolador, colocando un cero en el puerto  $PoF4$ .

Se realizaron rutinas independientes para testear cada instrucción del microcontrolador 8051. Estas rutinas se basaron en recibir un dato desde el exterior, ejecutar la instrucción a verificar y devolver el resultado por el puerto  $p3_{out}$ .

Se verificó el conjunto de instrucciones y se comprobó su correcto funcionamiento.

## V. CONCLUSION

En este informe se presentó el desarrollo de un proyecto final de carrera realizado sobre la base de un sensor de imágenes fabricado en tecnología 3D. El proyecto consistió en realizar la verificación funcional de un microcontrolador 8051 que forma parte del diseño del Imager.

## REFERENCIAS

- [1] Di Federico Martín, "Circuitos integrados de alto desempeño para visión con procesamiento basado en redes celulares", Tesis Doctor en Ingeniería, 2011.