

# CAPÍTULO 6: CONCLUSIONES Y LÍNEAS FUTURAS

En este último capítulo del presente proyecto se exponen las conclusiones obtenidas tras la realización del mismo, tanto en el aspecto general del tema tratado, como en el aspecto particular del trabajo realizado.

Y como punto final de esta memoria, se incluyen las posibles líneas futuras y tendencias que puede tomar la realización de simuladores de DSP, así como aplicaciones futuras al simulador desarrollado a lo largo de este proyecto.

## 6.1. CONCLUSIONES

Tras la finalización del proyecto, se ha conseguido realizar un simulador del DSP con las siguientes características:

- Permite cargar y simular un programa ensamblador del DSP TMS320C30.
- Emula características del compilador y linker de TI como la ubicación en memoria de secciones de datos y código.
- Realiza una simulación para una franja temporal elegida.
- Emula el comportamiento de los distintos componentes hardware del DSP TMS320C30 EVM: registros, memoria, CPU, puerto serie, AIC (dentro del mismo los conversores ADC y DAC), etc.
- Es capaz de realizar un tratamiento de una señal de entrada muestreada ubicada en un fichero de valores de entrada.
- Crea un fichero de muestras de salida con valores similares a los que produce el DSP TMS320C30 EVM bajo las mismas condiciones. Este fichero de salida permite la construcción de gráficas de representación de señal mediante el programa Matlab.
- Permite la elección de la frecuencia de muestreo de la señal de entrada, emulando así el comportamiento del ADC.
- Posee características del depurador del DSP TMS320C30, entre ellas las siguientes:
  - o Permite visualizar y modificar el contenido de los registros del DSP.
  - o Permite el acceso, visualización y modificación el contenido de las distintas posiciones del mapa de memoria del DSP.
  - o Es capaz de asignar y visualizar el contador de programa correspondiente a cada línea de código.

- Posee distintos modos de ejecución de código:
  - § Modo de ejecución continua: el programa se ejecuta hasta la finalización de la franja temporal elegida.
  - § Modo de ejecución paso a paso: es similar al modo de ejecución continua excepto que se ejecuta sólo una línea de código cada vez y se puede ver el contenido de registros y memoria.
  - § Modo de ejecución de bloque: es idéntico al modo de ejecución paso a paso pero ejecuta los bloques de manera continua.
  - § Modo de ejecución hasta el cursor: ejecuta el código de manera continua hasta la posición que se haya situado el cursor.
- Es capaz de comenzar una nueva ejecución de un programa cuando el usuario lo desee.
- Permite ver la ubicación de los registros del puerto serie para la correcta identificación de sus posiciones en el mapa de memoria.
- El simulador es capaz de interpretar la mayoría de instrucciones y algunas directivas de ensamblador y de todos los modos de direccionamiento excepto el de bit invertido.
- Emula y visualiza el flujo de control del programa del DSP durante una ejecución: muestra la instrucción siguiente a ejecutar y en todo momento el contenido de los registros y memoria de dicha ejecución.

## 6.2. LÍNEAS FUTURAS

La simulación de procesadores digitales de señal es realizable desde el punto de vista funcional, como se ha demostrado en el desarrollo del proyecto. Al interpretar cada línea de código ensamblador se consigue una simulación fiel para una franja temporal elegida. Con ello se ha conseguido el objetivo final del proyecto, que era realizar un simulador desde el punto de vista de la funcionalidad. Nunca se aspiró a realizar un programa que fuera fiel desde el punto de vista temporal.

Para conseguir un programa que simule el comportamiento temporal del DSP se tendrían que tener en cuenta los retardos producidos por cada instrucción, además de cómo se comportan las diferentes instrucciones en el '*pipeline*' (las instrucciones de ejecución paralela tienen un comportamiento temporal diferente ya que hacen varias operaciones

simultáneas). No obstante, para conseguir un rendimiento óptimo, se aconseja en este caso optar por un compilador, en lugar del intérprete que se ha realizado.

Otra línea futura es la realización de las interrupciones del DSP, además de emular completamente el comportamiento de los puertos serie y del controlador DMA. En el simulador se ha implementado la del puerto serie 0, pero no se ha implementado el resto por no considerarlo necesario para el objetivo final del proyecto.

Si se utiliza la tarjeta de sonido del PC, que posee conversores analógico-digitales y digital-analógicos, puede emularse el comportamiento del AIC programando adecuadamente desde el simulador la frecuencia de muestro. Así se puede conseguir una simulación virtual del DSP. Lógicamente, el simulador tendría que tratar las muestras de entrada en tiempo real o capturar una franja temporal, tratarla y luego presentar la salida correspondiente a esa franja temporal.

También se recomienda para un futuro, implementar el direccionamiento a bit invertido, que es particularmente interesante en la implementación de algoritmos como la transformada rápida de Fourier (FFT, *Fast Fourier Transform*).

Para que el programa simulador sea totalmente fiel a la realidad en cuanto al juego de instrucciones, se deben realizar las siguientes que no han sido implementadas: RND, NORM, TRAP, SWI, instrucciones con interbloqueo (que es otro objetivo futuro), SUBRB, SUBRF y SUBRI.

Una mejora respecto a la ubicación en memoria consiste en que el programa simulador permita ubicar cada sección de memoria (definida con la directiva .sect) en una zona elegida por el usuario, y no hacer esto de manera automática. También el simulador puede mejorar incrementando el número de directivas de ensamblador que sea capaz de interpretar (utilización de bloques de código, manejo de ficheros, etc...).

Otros programas de simulación como el '*Code Composer*' de TI, incluyen la capacidad de representar gráficamente las señales desde el mismo entorno de simulación. Esto podría hacerse en el simulador haciendo una llamada al programa Matlab, dando un formato previo adecuado a las muestras antes de representarlas gráficamente.

Los pasos que se han dado en la realización de este simulador pueden repetirse para otro DSP de distintas características. Dado el coste de cada módulo de evaluación de DSP, se hace necesario la implementación de simuladores educativos para el aprendizaje y primeros pasos antes de realizar pruebas importantes.

La motivación de realizar el simulador del DSP fue debida a que el número de alumnos por módulo hardware suele ser alto, y el tiempo escaso para realizar un aprendizaje

adecuado. La realización de simuladores de hardware se hará necesaria en un futuro para evitar costes y para una mejor preparación de los alumnos de cara a pruebas reales con el hardware en cuestión.