

Capítulo 2

Revisión Bibliográfica

2.1. DSP

El título anterior se ha dejado como sigla debido a que existen dos definiciones para ésta: como sustantivo *Digital Signal Processor* (Procesador Digital de Señal); y, como verbo *Digital Signal Processing* (Procesamiento Digital de Señal) [3].

Un Procesador Digital de Señal es un computador rápido dentro de un chip, que ha sido optimizado para la detección, procesamiento y generación de señales del mundo real (como la voz, video, música, etc.). Las funciones descritas anteriormente son realizadas en tiempo real.

El Procesamiento Digital de Señal se refiere a la manipulación de señales que han sido generadas en el mundo real, donde estas señales son representadas como dígitos (números). Es decir, estas señales se digitalizan para su posterior procesamiento.

2.1.1. Características

La expresión matemática en la Ecuación 2.1 describe el proceso de filtrado digital:

Ecuación 2.1

$$y(x) = \sum_{k=0}^K b(k)x(i-k) - \sum_{m=1}^M a(m)y(i-m)$$

Sin profundizar mucho en la matemática de esta relación, se puede ver que este proceso requiere una cierta cantidad de operaciones: sumas; restas; y, multiplicaciones. Los valores $b(k)$ y $a(m)$ son dos tablas de coeficientes del filtro guardados en memoria. Estos coeficientes serán multiplicados por un vector de muestras de entradas $x(i-k)$ y salidas $y(i-m)$ anteriores, respectivamente. Cada vez que una nueva muestra ingresa al sistema esta toma el primer lugar en la tabla de muestras y la más antigua es eliminada, esto debido a que se trabaja con tablas de largo fijo. Lo mismo ocurre con la tabla de salidas. Este proceso indica que un dispositivo DSP debe ser diseñado para acceder y administrar eficientemente sus áreas de memoria y, simultáneamente, manejar el flujo de datos desde y hacia los puertos de datos. La Figura 2.1 ilustra este proceso.

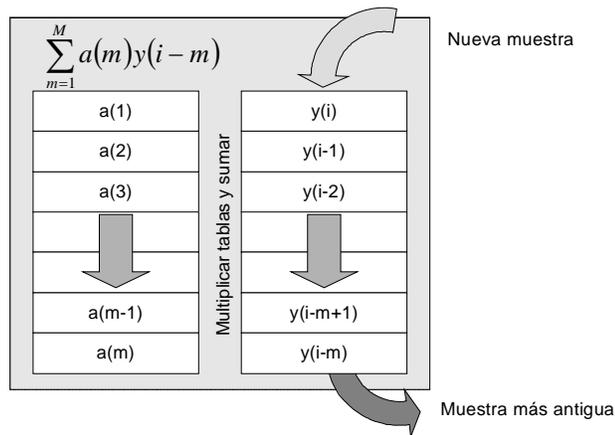


Figura 2.1 Coeficientes del Filtro y Muestras.

Otro ejemplo de algoritmos DSP es la Transformada Rápida de Fourier o FFT¹. La FFT usualmente es descrita por el número de muestras usadas en su cálculo, donde el número de muestras siempre es una potencia de 2 (por ejemplo: 128; 256; o, 512 puntos FFT). Se puede

¹ *Fast Fourier Transform.*

estimar la cantidad de multiplicaciones necesarias para cada punto de salida del resultado de la FFT según la Ecuación 2.2.

Ecuación 2.2

$$\text{Número de multiplicaciones} = \frac{N}{2} \log_2 N$$

Por ejemplo, si se tienen 128 muestras ($N=128$), el procesador DSP deberá realizar 448 multiplicaciones para obtener tan sólo un punto de la FFT.

Estos ejemplos se mencionan con el propósito de resaltar la naturaleza de los algoritmos DSP. En un típico sistema que usa un dispositivo DSP, las muestras arriban en un período regular de tiempo desde la entrada al DSP. En cada llegada de muestras se debe realizar todo el cálculo necesario que requiera cada algoritmo y el resultado debe ser presentado en la salida del DSP, todo esto antes de que llegue la próxima muestra de datos. Esta característica es esencial para cualquier dispositivo que requiera implementar exitosamente algoritmos DSP en tiempo real.

Los algoritmos básicos de procesamiento de señales mencionados serán usados sobre muestras de datos, calculando una suma (acumulación) de una serie de productos. Un DSP debe ser capaz de acumular series de resultados de multiplicaciones, conocida como *Multiply Accumulate* (MAC) o Multiplicación Acumulada.

Cabe destacar que el DSP está provisto de hardware dedicado para realizar las operaciones de multiplicación y MAC. El resultado obtenido de una multiplicación entre dos operandos usualmente requiere un ciclo de reloj del procesador DSP. En comparación con un típico microprocesador, este requiere de aproximadamente 80 ciclos de reloj para realizar una multiplicación de 16 bits².

En cuanto a la administración de la memoria, el DSP debe ser capaz de acceder a las áreas de memoria de la manera más eficiente posible de modo que el procesador no pierda tiempo

² *Binary Digit* o Dígito Binario.

esperando que los datos sean cargados³. Para el caso de la operación MAC, la cual se ejecuta reiteradamente por pequeños períodos de tiempo de manera muy rápida, es importante que una eficiente operación MAC no se “congele” mientras se espera que los datos sean cargados de la memoria. Los dispositivos DSP proveen una cierta cantidad de mecanismos para cargar los datos y determinar su dirección dentro de la memoria.

Otra característica importante en el diseño de muchos DSPs es la estructura o arquitectura de la misma memoria. Muchos microprocesadores son diseñados en base a la arquitectura Von Neumann, donde las instrucciones y los datos comparten el mismo espacio de memoria y son accedidos por los mismos buses de dirección y de datos.

La mayoría de los DSPs usan la arquitectura Harvard donde las instrucciones y los datos son guardados en sectores separados de memoria con accesos independientes. Esto es gracias a que las áreas de programa (instrucciones) y de datos poseen su propio bus de direcciones y de datos. Esta arquitectura habilita al DSP a acceder a instrucciones y datos simultáneamente, entregando una gran ventaja en cuanto a rapidez sobre un microprocesador convencional [4]. Además, se tiene la libertad de que el ancho (tamaño en bits) de los buses sea diferente, lo que permite que el ancho de los datos sea distinto que el de las instrucciones.

2.1.2. Interfaces

En la parte anterior se habló sobre la operación del DSP en cuanto a la parte central del mismo. A este se le denomina *Core* o núcleo. Las interfaces o periféricos del DSP permiten la comunicación entre lo que está fuera del DSP con el núcleo. La Figura 2.2 un dispositivo hipotético con algunas de las funciones o módulos típicos de un DSP.

³ Ciclo *Fetch* o Captura de una Instrucción.

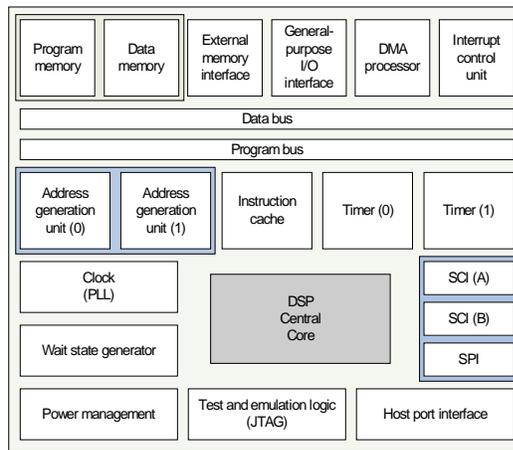


Figura 2.2 Dispositivo DSP hipotético.

Como se puede ver en la Figura 2.2 el núcleo se encuentra en el centro del dispositivo, rodeado por los componentes que están fuera de este pero que pertenecen al DSP. La mayoría de estos componentes o módulos están capacitados para ejecutarse en paralelo con el núcleo, tomando un pequeño tiempo del ciclo de instrucción del núcleo DSP. Un ejemplo de esto es un *on-chip timer* (temporizador interno), el cual puede ser inicializado para interrumpir al DSP cada un cierto tiempo predefinido.

Los componentes existentes dentro de un DSP son muchos y varían de acuerdo a la marca, familia o propósito de cada DSP.

Memoria

La mayoría de los dispositivos DSP están provistos con una cantidad limitada de *On-chip Read-Only Memory* o memoria sólo lectura interna (ROM) y *Random Access Memory* o memoria de acceso aleatorio (RAM). Además, existen varios tipos de memorias RAM como: *Dual-Access RAM* o RAM de doble acceso; y, *Single-Access RAM* o RAM de acceso simple.

La *On-chip ROM* es parte del espacio de memoria del programa y, en algunos casos, forma parte del espacio de memoria de los datos. En dispositivos con una pequeña cantidad de ROM,

esta contiene un *boot loader*⁴ el cual sirve para cargar de una ROM externa, por ejemplo, durante la secuencia de inicio. Además, el *boot loader* permite que el código de la aplicación o programa pueda ser cargado desde una interfaz serial o a través del uso de JTAG⁵.

La *On-chip Dual-Access RAM* (DARAM) recibe este nombre ya que cada bloque de esta memoria puede ser accedida dos veces por cada ciclo de máquina, es decir, puede leer y escribir en un bloque de DARAM en el mismo ciclo.

En la *On-chip Single-Access RAM* (SARAM) cada bloque es accesible una vez solamente por cada ciclo de máquina, siendo de escritura o de lectura. En ambos casos, las memorias siempre son mapeadas y su principal función es la de almacenar valores de datos.

Interfaz de E/S de Propósito General

Se conoce como *General Purpose Input/Output* (GPIO) o Entrada/Salida de Propósito General. Esta interfaz contiene puertos de señales digitales de entrada y/o salida. Generalmente se trata de un puerto paralelo el cual se puede configurar como entrada o salida. Cuando es configurado como salida, se puede escribir en un determinado registro para que controle la salida de cada pin del puerto. Cuando es configurado como entrada, se puede detectar el estado del pin de entrada leyendo el estado de un registro interno asociado al puerto.

Puertos Seriales

Un puerto serie es una interfaz entre distintos dispositivos (microcontroladores, computadores, periféricos, etc.) donde la información es enviada bit a bit, pudiendo enviar un sólo un bit a la vez. Dentro de un dispositivo DSP los puertos seriales más comunes son: *Serial Communication Interface (SCI)* o Interfaz de Comunicación Serial y *Serial Peripheral Interface (SPI)* o Interfaz Serial de Periféricos.

⁴ Cargador.

⁵ *Join Test Action Group*.

El módulo SCI utiliza una comunicación asíncrona. Debido a esto sólo necesita dos líneas de comunicación: la línea de transmisión (TX) y de recepción (RX). Generalmente, este módulo se utiliza para comunicar el DSP con un computador personal (PC).

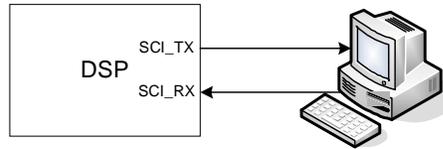


Figura 2.3 Comunicación mediante SCI.

El módulo SPI es un estándar de comunicaciones usado principalmente para la transferencia de información entre dispositivos electrónicos. Se caracteriza por utilizar una comunicación síncrona en modo Maestro-Esclavo (*Master-Slave*), donde sólo un dispositivo Maestro puede controlar varios dispositivos Esclavos. Esto es gracias a que este estándar es de topología Bus. Incluye: una línea de reloj (*Clock SCLK*); una línea de datos de entrada (*Master Input - Slave Output MISO*); una línea de datos de salida (*Master Output - Slave Input MOSI*); y, un pin de selección (*Chip Select CS*) que conecta y desconecta la operación del dispositivo con el que se desea comunicar. Se deberán utilizar tantos pines CS como dispositivos esclavos se tengan.

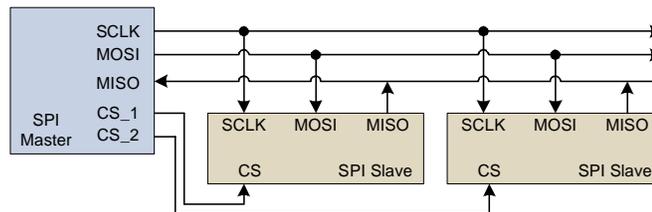


Figura 2.4 Comunicación mediante SPI.