

APÉNDICE D: DESCRIPCIÓN DEL DSP56002 DE MOTOROLA.

D.1) Introducción.

El DSP56002 es un procesador digital de señal de propósito general de 24 bits, con formato numérico de punto fijo, perteneciente a la familia de procesadores DSP56K de Motorola. Estos procesadores se caracterizan por estar especialmente diseñados para implementar aplicaciones de procesamiento de señal como es el caso de la cancelación de ecos acústicos. Más adelante se verán las características más destacadas de este procesador y los detalles que le hacen idóneo para su aplicación a nuestro diseño.

El DSP56002 se incluye en un sistema de desarrollo hardware, conocido como ADS (*Application Development System*), que se emplea para el diseño, la depuración y la evaluación de prestaciones del sistema en tiempo real. Aparte de las facilidades hardware, con el procesador se incluye una serie de programas, entre los que destaca el simulador, el ensamblador y el compilador cruzado de lenguaje C.

D.2) El procesador DSP56002.

El procesador DSP56002, al igual que los pertenecientes a la familia DSP56K de Motorola, se caracteriza por:

- **Velocidad:** el procesador realiza 20 millones de instrucciones por segundo, con un ciclo instrucción de 50 nseg, con un reloj de 40 MHz. Esta elevada velocidad, así como otras características de su arquitectura hacen que sea muy eficiente en aplicaciones de tiempo real.
- **Precisión:** los datos son de 24 bits, lo que proporciona un margen dinámico de 144 dB; los resultados intermedios en las operaciones que se realizan en el procesador pueden llegar a los 56 bits, con un margen de 336 dB.
- **Procesos paralelos:** el diseño del chip facilita la utilización de diversos buses simultáneamente, siendo, por tanto, posible, en un único ciclo de instrucción, el cálculo de una operación y uno o dos movimientos de datos o actualizaciones de punteros.
- **Lenguaje ensamblador:** el conjunto de instrucciones del DSP es muy similar a los empleados por otros microprocesadores más populares, como son los de la familia 68K de Motorola. Esto hace que la programación del DSP sea muy sencilla para quien esté familiarizado con la programación en ensamblador.
- **Bajo consumo:** esto se debe a que el DSP56002 está fabricado en tecnología HCMOS. Además existen instrucciones que permiten estados de espera en los que el gasto es mínimo.

En los apartados siguientes se expondrán las principales características de la arquitectura del DSP, poniendo un mayor énfasis en los formatos numéricos, instrucciones y modos de direccionamiento.

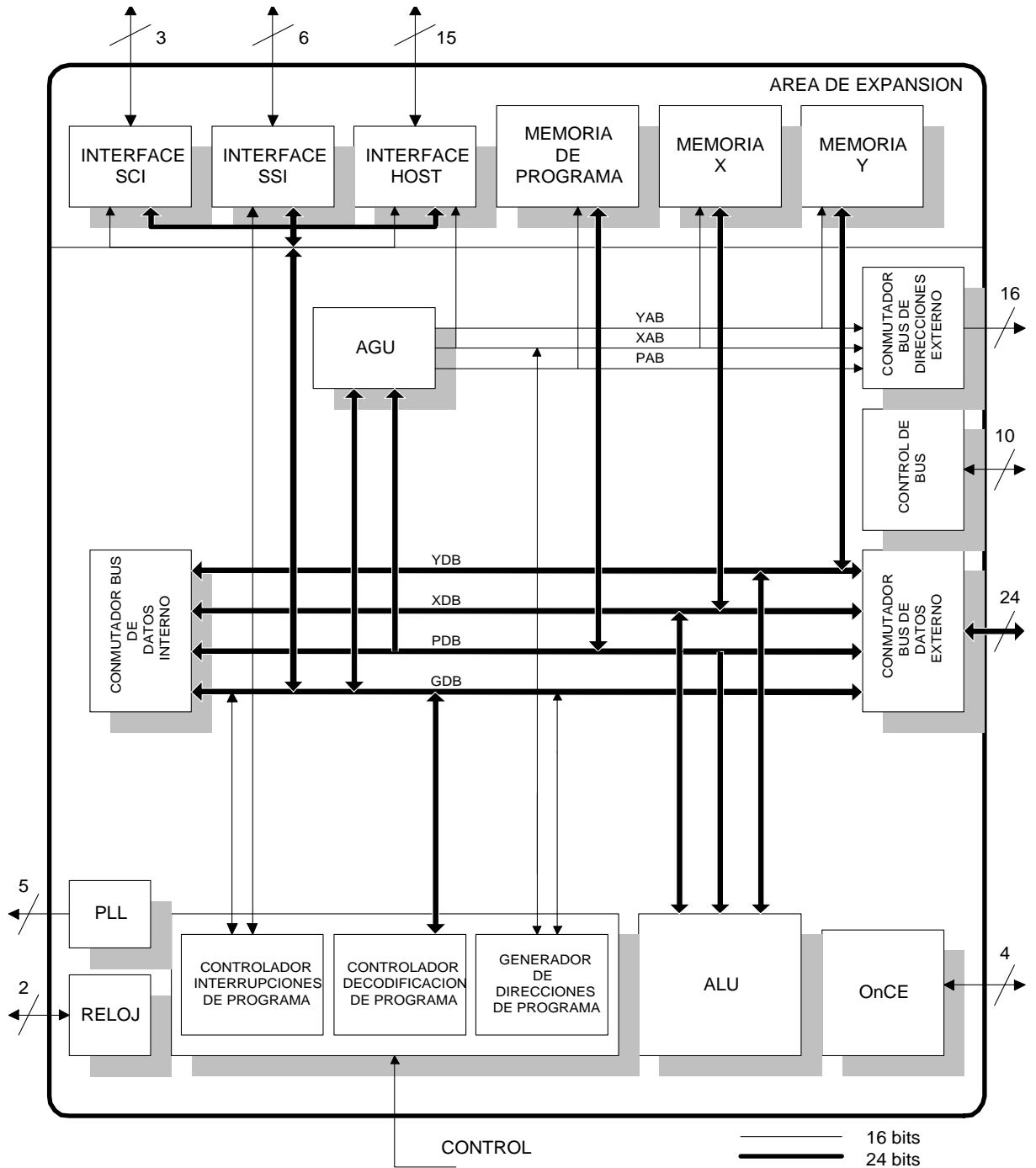


Figura D.2.1. Arquitectura del DSP56002.

D.2.1) Arquitectura del DSP56002.

Los procesadores de la familia DSP56K se encuentran en un módulo central de procesos. A su alrededor, en el área de expansión, el chip puede soportar varias configuraciones de memoria y módulos periféricos. En el módulo central se pueden distinguir las siguientes partes:

- Buses de datos.
- Buses de direcciones.
- Unidad aritmético-lógica de datos (ALU).
- Unidad generadora de direcciones (AGU).
- Unidad de control de programa (PCU).
- Expansión de memoria (Puerto A).
- La herramienta de depuración *On-Chip Emulator* (OnCE).
- Circuitería de reloj basada en PLL.

En la figura D.2.1 puede observarse la distribución de los elementos anteriores en el DSP.

D.2.2) Buses de datos.

El módulo de proceso central está organizado alrededor de los registros de unidades independientes: PCU, AGU y ALU. El movimiento de datos entre estas tres unidades se produce a través de cuatro buses bidireccionales de 24 bits. Éstos son:

- XDB: Bus de datos de la memoria X.
- YDB: Bus de datos de la memoria Y.
- PDB: Bus de datos de la memoria de programa.
- GDB: Bus de datos global.

Las transferencias entre memorias X e Y y la ALU se realizan a través de XDB e YDB. Para las transferencias entre memoria de programa y PCU se utiliza el PDB. EL GDB se emplea en otras transferencias, como en operaciones de entrada y salida con periféricos.

D.2.3) Buses de direcciones.

Existen tres buses de direcciones de 16 bits:

- XAB: Bus de direcciones de la memoria X.
- YAB: Bus de direcciones de la memoria Y.
- PAB: Bus de direcciones de la memoria de programa.

Las llamadas a memoria interna se realizan a través de estos tres buses. Las llamadas a memoria externa se realizan mediante un único bus de 16 bits que, a través del conmutador de bus de direcciones externo, se conecta con una de los tres buses internos. Sólo se permite un acceso a memoria externa por ciclo, mientras que, también por ciclo, es posible realizar simultáneamente un acceso a memoria X, un acceso a memoria Y y una operación de cálculo o de movimiento de registros en el interior de la ALU.

D.2.4) Unidad Aritmético-Lógica (ALU).

La ALU (Fig. D.2.2) es la unidad del procesador que se encarga de la realización de las operaciones aritméticas y lógicas. La ALU está especialmente diseñada para facilitar las operaciones más usuales en el tratamiento de la señal como son sumas, restas, productos y combinaciones de las anteriores.

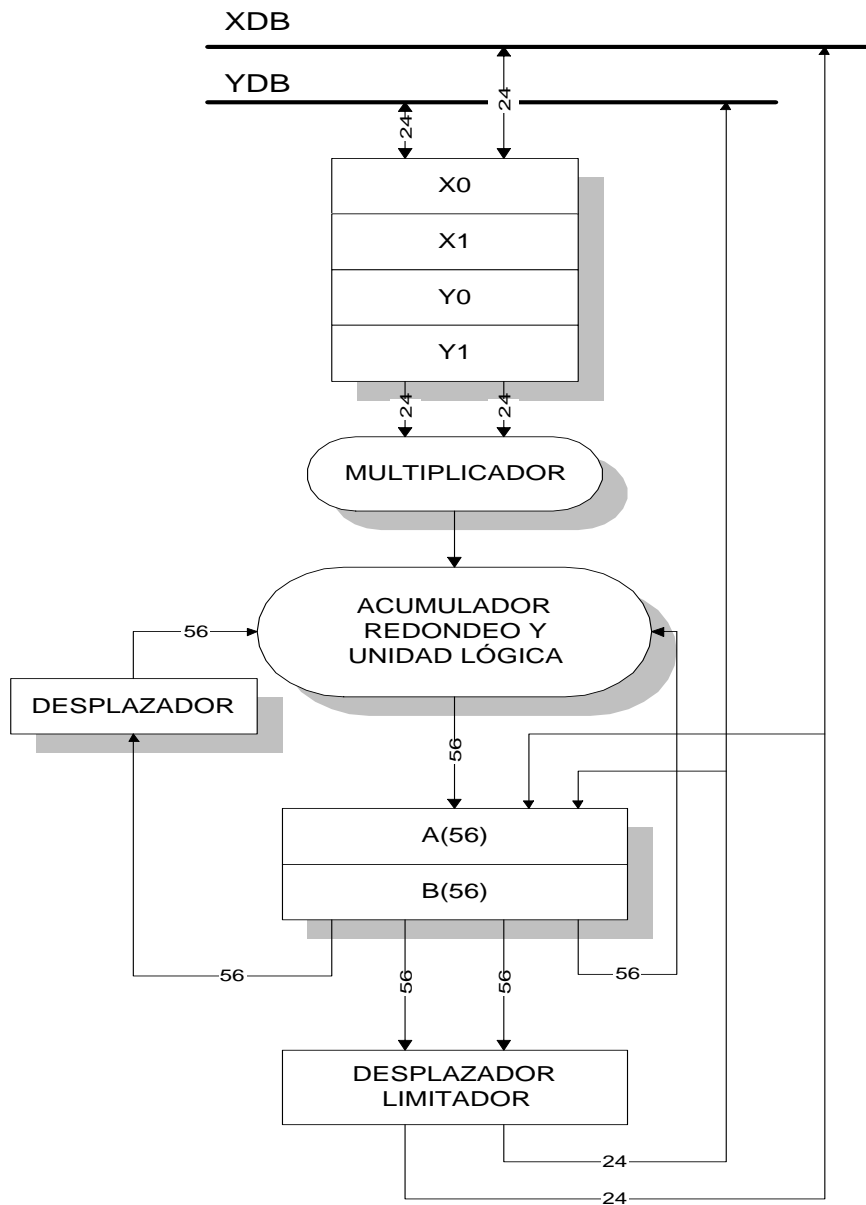


Figura D.2.2. Unidad aritmético-lógica.

Los componentes de la ALU son:

- Cuatro registros de 24 bits (X0, X1, Y0, Y1) con la posibilidad de ser utilizados como dos únicos registros de 48 bits (X=X1:X0, Y=Y1:Y0).
- Un multiplicador-acumulador (MAC).

Dos registros de 56 bits (A, B) llamados acumuladores, cada uno desglosable en tres registros: dos registros de 24 bits y un registro de extensión de 8 bits (A=A2:A1:A0, B=B2:B1:B0).

- Un desplazador de los acumuladores.
- Dos circuitos desplazadores/limitadores del bus de datos.

Los registros de entrada (X0, X1, Y0, Y1) contienen los operandos que se pasarán al MAC para realizar el cálculo. También pueden utilizarse como meros registros de almacenamiento mientras no se necesiten para realizar alguna operación. Nunca es posible realizar una operación cuyos operandos estén en posiciones de memoria. El resultado quedará almacenado en alguno de los acumuladores (A, B).

Las palabras de 24 bits proporcionan un margen dinámico de :

$$MD(24) = 20 \cdot \log_{10}(2^{24}) = 144dB$$

Este margen es suficiente pues la mayoría de los convertidores de datos trabajan, como mucho, a 16 bits (como es el caso de los empleados en nuestro diseño de cancelador). Los acumuladores, de 56 bits, proporcionan un margen de:

$$MD(56) = 20 \cdot \log_{10}(2^{56}) = 336dB$$

Con este margen no es previsible que se produzca una pérdida de precisión en operaciones intermedias. En el caso de que se produjesen saturaciones la ALU está preparada para limitar los datos e intentar minimizar en lo posible los efectos de estos desbordamientos.

La ALU puede realizar, en un único ciclo de instrucción, las operaciones siguientes (D indica un acumulador de destino y S1, S2 registros de entrada):

- Multiplicación. (D=S1*S2).
- Multiplicación y acumulación, con signo positivo o negativo. (D=D+S1*S2, D=D-S1*S2).

- Redondeo. ($D=D+r$).
- Multiplicación y acumulación, con signo positivo o negativo y redondeo. ($D=D+S1*S2+r$, $D=D-S1*S2+r$).
- Suma. ($D=S1+S2$).
- Resta. ($D=S1-S2$).
- Una iteración de división (no realiza divisiones completas).
- Desplazamientos a nivel de bits, a izquierda o derecha.
- Operaciones lógicas de 24 bits: AND, OR, NOT y OR exclusivo.

Dentro de este conjunto de operaciones, las que merecen mayor atención son las de multiplicación y acumulación combinadas. Este tipo de operaciones no suele existir dentro de las unidades aritmético-lógicas de microprocesadores de propósito general. Su existencia en el DSP56002 se debe a que la operación de convolución, presente en la mayoría de las aplicaciones de procesamiento de señal, se simplifica considerablemente. Por ejemplo, para la implementación de un filtro, se debe realizar el siguiente cálculo:

$$y(n) = \sum_{k=0}^N h(k) \cdot x(n-k)$$

En el DSP56002 este filtrado se realizaría con dos lecturas de datos y una instrucción de cálculo incluida en un bucle que se repitiera N veces. Además, si se aprovecha la capacidad del procesador de realizar simultáneamente un cálculo en la ALU y dos lecturas en memoria, el interior del bucle consistiría en una única línea. Algo parecido a esto:

```

do #N,FinBucle
mac x0,y0,a  x(⊗r0)+,x0  y(⊗r4)+,y0
FinBucle

```


D.2.5) Formato numérico.

El DSP56002 usa una representación fraccional en complemento a 2 de los datos para todas las operaciones de la ALU. Esto implica que los datos de entrada de una aplicación deben ser convertidos a número fraccional antes de ser usados en algún cálculo.

Para palabras de 24 bits, las correspondientes a los registros de entrada de la ALU y a las posiciones de memoria, los valores numéricos posibles se encuentran entre el -1 y el $1-2^{-23}$, como puede verse en la siguiente tabla:

DECIMAL	HEXADECIMAL
$1-2^{-23}$	\$7FFFFFF
$1-2^{-22}$	\$7FFFFE
...	...
2^{-22}	\$000002
2^{-23}	\$000001
0	\$000000
-2^{-23}	\$FFFFFF
-2^{-22}	\$FFFFFE
...	...
$1-2^{-23}$	\$800001
-1	\$800000

Tabla D.2.1.

Como se deduce de la tabla, el bit 23 de la palabra de 24 bits (bit más significativo o BMS) es el bit de signo (0 positivo y 1 negativo). El punto decimal se encuentra entre el bit 23 y el bit 22 de la palabra de 24 bits. Hay que notar que el conjunto de valores no es simétrico, es decir, existe un número negativo más (el -1). El hecho de que no exista el número 1 puede afectar algunas operaciones. Por ejemplo, al no existir el 1 no existe el elemento neutro de la multiplicación, es decir, no podemos multiplicar un número por ningún otro y que el resultado sea idéntico al número inicial. Sin embargo, en cualquier aplicación práctica no supone ningún problema emplear el valor más cercano ($1-2^{-23}$) en lugar del 1. Cuando se trata de datos de 56 bits en los acumuladores, los valores numéricos ya no están necesariamente incluidos entre 1 y -1. Cuando se copia un número de 24 bits a un acumulador, éste se copia en el registro intermedio A1 ó B1 se colocan ceros en los 24 bits menos significativos correspondientes a los registros A0 ó B0 y se produce una extensión del signo al registro de extensión de 8 bits A2 ó B2 (Fig. D.2.3).

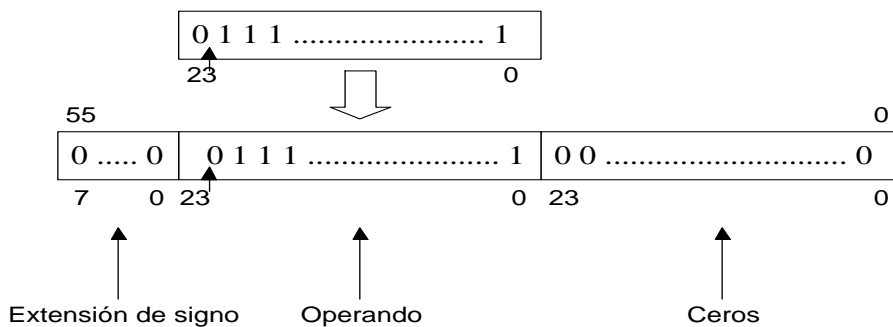


Figura D.2.3. Conversión de un número de 24 bits a 56 bits.

El registro de extensión de 8 bits asociado al acumulador permite el crecimiento de los datos por encima de la unidad, de modo que los valores que el acumulador puede contener varían entre -256 y 256 aproximadamente. Esta característica permite realizar en la ALU cálculos intermedios que pueden desbordar el límite de las palabras de 24 bits. Sin embargo, cuando se usan los registros de extensión del acumulador, los datos contenidos en el acumulador no pueden ser transferidos a memoria o a otros registros sin ser limitados. El mecanismo de limitación consiste en convertir el dígito de 56 bits en el valor más cercano representable con 24 bits. Cuando el dígito del acumulador es mayor que 1 o menor que -1, el mecanismo de limitación de la ALU transfiere al registro o la memoria los valores \$7FFFFFFF y \$80000000 respectivamente. El mecanismo de limitación es análogo al que ocurre en caso de desbordamiento de una operación de cálculo. En ambos casos queda reflejado en el registro de estado (SR).

Independientemente de los registros de extensión, al duplicar el número de bits de la palabra se obtiene en los acumuladores un aumento de la precisión en las operaciones intermedias. Con los 48 bits restantes, los valores numéricos posibles se encuentran entre el -1 y el $1-2^{-47}$, como puede verse en la siguiente tabla:

DECIMAL	HEXADECIMAL
$1-2^{-47}$	\$7FFFFFFF
$1-2^{-46}$	\$7FFFFFFE
...	...
2^{-46}	\$0000000002
2^{-47}	\$0000000001
0	\$0000000000
-2^{-47}	\$FFFFFFFF
-2^{-46}	\$FFFFFFF
...	...
$1-2^{-47}$	\$8000000001
-1	\$8000000000

Tabla D.2.2.

Ahora, el bit 47 de la palabra de 48 bits es el bit de signo (0 positivo y 1 negativo). El punto decimal se encuentra entre el bit 47 y el bit 48 de la palabra.

D.2.6) Unidad Generadora de Direcciones (AGU).

La unidad generadora de direcciones es la encargada de realizar los cálculos necesarios para obtener la dirección efectiva de los operandos en memoria. No utiliza aritmética fraccionaria sino aritmética entera. La AGU opera en paralelo junto a otros recursos del sistema para ahorrar tiempo de ejecución.

La AGU está dividida en dos mitades idénticas, cada una de las cuales posee una unidad aritmético-lógica destinada a realizar los cálculos pertinentes y cuatro conjuntos de tres registros (Fig. D.2.4).

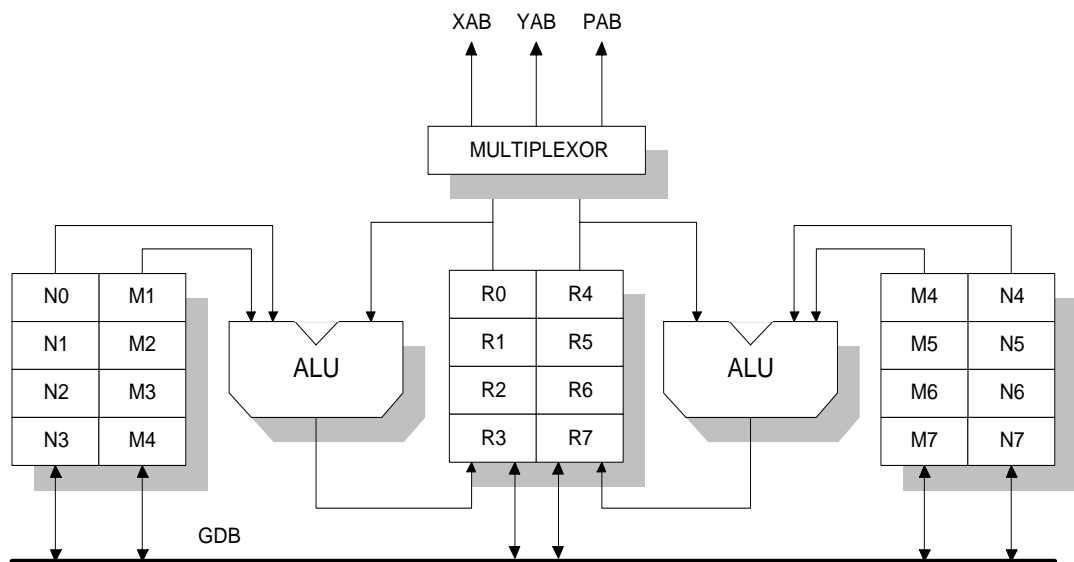


Figura D.2.4. Unidad Generadora de Direcciones.

Estos registros son los registros de direcciones (R0 - R3 y R4 -R7), los registros de offset (N0 - N3 y N4 - N7) y los registros modificadores (M0 - M3 y M4 -M7). Los registros, de 16 bits, trabajan siempre por tríos.

Los ocho registros de dirección (R0 - R7) pueden contener direcciones o datos. Cuando se usan para direccionamientos, éstos pueden contener la dirección o pueden llevar un offset que será el contenido en el registro Nn asociado. Además, el contenido

de Rn puede ser pre-actualizado o post-actualizado de acuerdo con el modo de direccionamiento seleccionado. Si existe actualización, los registros modificadores Mn son los que especifican la aritmética de actualización.

Existen tres aritméticas de actualización:

- Lineal: los incrementos y decrementos se realizan linealmente a lo largo de las 65536 localizaciones de memoria direccionables.
- Módulo: se crean *buffers* circulares de cualquier tamaño entre 1 y 65536. Dentro del *buffer*, los incrementos y decrementos se realizan de manera lineal.
- Acarreo inverso (*Bit-reverse*): Es la aritmética de actualización especialmente pensada para la realización de la FFT. Con esta opción es posible crear *buffers* circulares en los que los datos se almacenan en el orden adecuado para que al realizar la FFT los datos salgan ordenados de manera secuencial.

En cualquiera de las aritméticas anteriores existen los siguientes modos de direccionamiento:

- Sin actualización: (Rn).
- Post-incremento por 1: (Rn)+.
- Post-decremento por 1: (Rn)-.
- Post-incremento por el offset Nn: (Rn)+Nn.
- Post-decremento por el offset Nn: (Rn)-Nn.
- Indexación por el offset Nn: (Rn+Nn).
- Pre-decremento por 1: -(Rn).

D.2.7) Unidad de Control de Programa (PCU).

La unidad de control de programa tiene como funciones la adquisición de las instrucciones del programa, la decodificación de las instrucciones, el control de los bucles y el procesamiento de interrupciones.

La PCU se divide en tres bloques funcionales:

- Controlador de decodificación del programa.
- Generador de instrucciones de programa.

- Controlador de interrupciones.

Desde el punto de vista del programador, la PCU puede reducirse a un conjunto de registros con el que se controla la ejecución del programa. Estos registros son:

- Contador de programa (PC): es un registro de 16 bits que contiene la posición de la próxima posición de memoria.
- Registro de estado (SR): es un registro de 16 bits que, a su vez, se subdivide en dos registros de 8 bits:
 - CCR: corresponde a los bits menos significativos y define el estado actual del usuario. Sus bits se pueden ver afectados por operaciones en la ALU y por movimientos de datos en los que se produce limitación.
 - MR: define el estado actual del sistema y contiene, entre otras cosas, la máscara de interrupción.
- Registro de modo de operación (OMR): Es un registro de 24 bits, aunque sólo se usan 6. Determina el mapa de memoria que se usa dentro de los posibles y el modo de arranque.
- Pila del sistema (SS): Es un memoria interna de 15x32 bits y se emplea para almacenar el PC y el SR en llamadas a rutinas, interrupciones y bucles.
- Registro de puntero de pila (SP): Es un registro de 6 bits que contiene la última dirección utilizada de la pila. Además contiene unos bits que avisan cuando ocurre un desbordamiento de la pila.
- Registro de dirección de bucle (LA): Es de 16 bits e indica la posición de la última instrucción de un bucle
- Registro contador de bucle (LC): Es de 16 bits e indica el número de veces que debe ejecutarse un bucle.

D.3) El Sistema de Desarrollo de Aplicaciones (ADS)

El sistema de desarrollo de aplicaciones se utiliza para diseñar sistemas de procesamiento de señal en tiempo real. El ADS proporciona una herramienta de diseño, depuración y evaluación de sistemas basados en DSP.

Está formado por cuatro componentes (Fig. D.3.1): el módulo de desarrollo de aplicaciones (ADM) que contiene al DSP y a la circuitería de control, la placa de interfaz con el bus del ordenador para controlar hasta 8 ADMs, el convertor de

comandos que interacciona con el puerto OnCE del DSP y un programa software que comunica al usuario con el ADM y con el convertor de comandos.

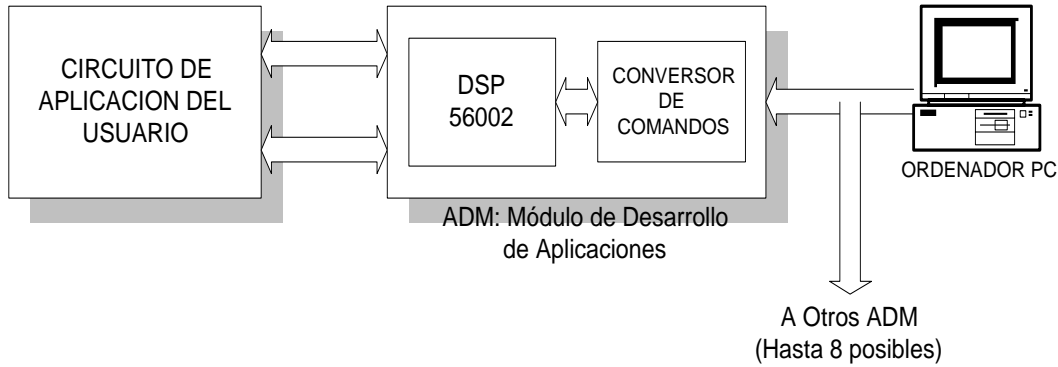


Figura D.3.1. Esquema del ADS.

El *hardware* y el *software* del ADS requieren para su funcionamiento un ordenador personal (PC). El programa de interfaz de usuario proporciona las rutinas necesarias para la comunicación con el DSP56002, incluido en el ADM. Este programa posee un grupo de comandos que permiten al usuario realizar una gran variedad de tareas.

El ADM es la plataforma básica para evaluar el funcionamiento del DSP. Contiene el chip DSP56002 con un conector al puerto OnCE y permite ser configurado como esclavo del ordenador o como una unidad de funcionamiento independiente. Actuando como esclavo, el usuario controla el procesador y es capaz de interrogar su estado. Esto permite la depuración sencilla del *hardware* y del *software*. En la configuración como unidad independiente, un programa de usuario residente en memoria ROM controla el ADM y puede ser utilizado como prototipo de un sistema final. Por su parte, el usuario interacciona con el sistema a través de una línea de comandos en la pantalla del ordenador.

Al ADS se conectarán los convertores A/D y D/A que proporcionaran las señales de entrada y que tomarán las que correspondan como salida.