Audio Engineering Society

Convention Paper 5377

Presented at the 110th Convention
2001 May 12–15 Amsterdam, The Netherlands

DSD-Wide
A Practical Implementation for Professional Audio

P. Eastty, N. Bentall, G. Cook, C. Gerard,
C. Sleight, M. Smith and P. Thorpe
SONY Pro-Audio R&D
Oasis Park, Eynsham, Oxford OX8 1TP, UK

ABSTRACT

This paper presents practical recipes for the processing of DSD-Wide [64FS 8-bit] signals which are fully compatible with the DSD [64FS 1-bit] signals used by the SACD consumer audio format. The designs are presented in a schematic form compatible with implementation by interested engineers in either FPGA or (with some modification) by traditional DSP methods. This is intended to open up the processing of such Super High Fidelity signals to a wider audience.

INTRODUCTION

The work described in this paper extends our earlier work on the processing of single bit signals at 64FS as detailed in (1).

This paper presents an intensely practical view of the processing of 8-bit signals at 2.8224MHz (64FS).

It begins with a short historical summary and discusses the reasons for migrating to 64FS 8-bit signals [DSD-Wide].

A detailed description of the hardware produced to process such signals is given addressing in particular choice of processing circuits, module size, inter-processor and inter-card connection, system I/O, clocks, power supplies and thermal considerations. The hardware format chosen, a single (but expandable) PCI card in a standard PC, leads to a moderate cost system capable of a fair degree of expansion towards larger systems.

The largest section of the paper is devoted to the processing algorithms used including a single bit router, 8-bit router, conversion from single bit to 8-bit and 8-bit to single bit, equalization and filtering, dynamics processing, gain control and mixing, and metering.

Specifically examined are factors contributing to processing efficiency that lead to single chip designs that individually achieve:

128*128 1-bit router, 512*512 8-bit router, 64-way 1-bit to 8-bit conversion, 64-way 8-bit to 1-bit conversion. Frequency response plots of both the 1-bit to 8-bit and 8-bit to 1-bit conversions are included;

64-input, 64-output mixer containing a maximum of 512 freely positioned fully interpolated cross-points;

128 second-order filtering/equalizing sections, including full linear interpolation of coefficient values;

32-way dynamics section, each with independently controllable breakpoints and time-constants;

32-way metering, each with five independent frequency bands.

A final short section deals with a single card 8-channel mastering mixer created by software from the hardware and DSP elements described above capable of simultaneously processing both surround and stereo recordings. This mixer is fully integrated with a hard-disk recording/editing system (not forming part of this paper), sharing the same PC and user interface.

HISTORY

The format used on SACD calls for 1-bit samples at 2.8224MHz. In all practical implementations this format is combined with fifth order or greater noise shaping to yield a 1-bit Delta-Sigma signal referred to as Direct Stream Digital or DSD.

For some years now we have been investigating professional audio processing and storage formats suitable for the preparation of recordings for release on SACD.

There has always been a tendency for the development of an appropriate professional format to lag behind the release of a new consumer format. Typical of this was the introduction of CD (a 16-bit format), where two-channel 16-(or fewer) bit recorders appeared first, to be followed by multi-track 16-bit recorders, and only much later by compatible digital mixers and multi-track recorders of more than 16-bit resolution. A similar sequence of introductions has occurred for SACD: 1-bit two-track recorders followed by multi-track recorders then mixers and editors. We are now beginning to see the first professional processing equipment of greater than 1-bit (consumer format) resolution.

Choice of Professional Format

Since today large quantities of equipment are in existence that use the consumer DSD format, and we shall always need to convert from professional to consumer for release, the ease of conversion between professional and consumer formats is an important factor in the selection of the professional format.

Given that the consumer format is 1-bit at 64FS (64 times 44.1kHz) the choice of a professional format might go in one of two directions (or possibly in both directions at once!): we might increase the number of bits, the sampling frequency or both.

In effect there are two areas of professional equipment which have slightly differing requirements, processing and storage.

Processing

The format requirements for processing are usually dominated by those for the interconnection of the various processing units. Typically the internal arithmetic carried out inside any given processing unit is the private domain of the equipment designer. However, there are two important provisos to this.

Firstly, the internal processing must be of sufficient quality to justify its connection to the rest of the system. There really is no point designing a piece of equipment with telephone bandwidth and expecting users to make CDs with it. Similarly there is little point in making a DSD recording and then reducing the sampling rate by a factor of eight to do the processing.

Secondly, there is no point in using an internal processing method which is so incompatible with the interconnection standard that the costs and signal degradation incurred at the input and output of the processing are significantly greater than the costs of the processing itself.

Both of these considerations rule out the use of 44.1kHz (or even 176.4kHz or higher) PCM systems as part of a DSD production chain.

Storage

The format requirements for recording must be compatible with those for processing but include the requirement for variable resolution. Not every recording needs (or requires) to be done at the absolute maximum fidelity. A recording format should allow the user to manage the trade-off between quality and media cost. Analogue recording allowed the varying of tape speed (usually at the flick of a switch) or tape width (by changing head blocks); these corresponded (loosely) to changing bandwidth and dynamic range. It would be good if a professional DSD recording system could afford us at least some of these facilities.

Thus while we can expect a single format for professional processing we will sometimes require multiple, compatible, formats for professional storage.

From Processing to Storage and Back Again

The transition between the signal representation used for processing and the formats used for storage is vitally important. In a typical production chain a signal may go from processing to storage to processing to storage, etc., dozens of times. Any inaccuracies in the signal conversion (in either direction) will be magnified each time the conversion takes place. This is why we should put such an emphasis on examining the inaccuracies of conversion when we come to selecting a professional format.

Let's consider the two types of conversion that might be needed: change of bit-width and change of sampling frequency. Since change of sampling frequency involves change of bit-width as part of the required processing, I'll deal with bit-width first.

Change of Bit-Width

Obviously the width of a signal can be increased or decreased. And although a professional format would have as its minimum width that of the distribution format (i.e., 1 bit), in the real world we shall need the ability to convert from professional to consumer AND from consumer to professional.

The increase in the number of bits used to represent a signal is usually done by simply adding some zeros to the least significant end of the word. This is similar to writing 3kg as 3000g. It is the same quantity; we've just used a representation which will allow us to add extra precision when we do processing. Whether we wish to halve 3kg of tomatoes or reduce an audio signal by 6dB, we usually need more digits in the representation.

Reducing the number of bits we use requires a little more artifice. If we halve our 3000g of tomatoes we get 1500g. Now if we wish to convert this back to a number of kilograms we have a problem, we can either say 1kg or 2kg but neither of these is accurate. We have an error!

If the greengrocer charges us for 1kg we get 500g of tomatoes free; if he charges us for 2kg we pay for 500g we have not received. Neither is correct. If we tried to buy 1500g of tomatoes every day and the greengrocer always charged us for 2kg we would not be happy. If he always charged us for 1kg he would not be happy. Either of these "always" strategies is "truncation": with tomatoes it makes somebody lose money, with audio it makes a nasty noise. However, suppose that we toss a coin each day: heads we pay for 1kg, tails we pay for 2kg. Over a number of days we will have paid the correct amount. This is what happens when we use a properly dithered conversion from more bits to fewer: with tomatoes or audio the result is a correct conversion.

However, in audio the addition of the coin toss is equivalent to a slight increase in noise. The conversion of the audio is "perfect" in the sense that we have added no distortion or other horrible artifacts, but we have added a small amount of noise. The spectrum of this noise can be controlled by "noise shaping" and this is frequently used to sweep the necessary noise (the penalty for a "perfect" conversion from more bits to fewer) into an area of the spectrum where the ear is less sensitive. In typical 44.1kHz digital audio there is only a small part of the available spectrum upto 22.05kHz where this noise can be swept. This means the gains which noise shaping can provide at 44.1kHz are limited. Alternatively, when using the 2.8224MHz sample rate of DSD, there are vast areas of the spectrum up to 1.4112MHz which can be used. This accounts for the massive signal to noise ratio improvements which can be achieved by noise shaping at DSD-like sampling rates.

The term "noise shaping" is generally reserved for the noise spectrum manipulation applied when converting from many bits to not-so-many bits. When this same technique is applied to conversion from many-bits to 1-bit it is referred to as Delta Sigma Modulation. For our present purposes the name is very different but the techniques are substantially the same.

To summarise:

> The conversion from few bits to many bits is simple and can be done completely accurately.
>
> The conversion from many bits to few bits is slightly more complex but can be done completely accurately without introducing any distortion. There is a slight noise penalty which can be minimised by noise shaping, particularly at the high sample rates used for DSD.

Almost any processing we do, mixing, filtering, editing, etc., will involve the creation of longer words from shorter ones and the subsequent requirement to reduce the word length back to the standard value. We're going to do a lot of this conversion. So we have to get it right whatever else we decide about professional standards. Of course this applies to 44.1kHz PCM too but that's another story ...

Change of Sampling Frequency

This is also known as sample rate conversion. We normally meet this at lower frequencies between say 44.1kHz and 48kHz, but in the DSD case we would be converting 64FS to or from 128FS or higher.

It is worth noting here that we are not considering using a LOWER sampling rate for professional equipment than for consumer. That would be like using a lower analogue tape speed for professional recordings than for consumer distribution.

Sample rate conversion is a complex business and by its very nature will involve the creation of longer words than are present at the input. This will lead to the subsequent requirement for the reduction of the word length of these internal signals to the standard value for output. Again we'll be using the bit-width reduction processing from above, including the noise shaping.

Sample rate conversion involves a large quantity of filtering.

Bandwidth Limitation

This is only required when we convert from a higher sampling rate to a lower one. Under these circumstances the available bandwidth at the input will be greater than the available bandwidth at the output (the input Nyquist limit is higher than the output Nyquist limit). Because of this we'll need to filter the input signal to remove components above the output Nyquist limit. Otherwise these components will produce alias frequencies in the output which are unrelated to anything in the input signal. A nasty noise.

Interpolation

The heart of a sample rate converter is an interpolation filter, designed to produce output samples at the output sampling rate that correspond to the times between the input sampling sequence when the output samples should occur. Without going into detail, this is yet another filter.

The Problems of Sample Rate Conversion

None of these filters can be made with a perfectly flat in-band frequency response. None of these filters can be made with perfect rejection of alias components. Every one of these filters requires the noise shaped reduction of its output bit-width back to the standard.

Most implementations of sample rate converters have some deeply buried truncation effects, certainly not all, but most.

To summarise:

> Sample rate conversion is MUCH more complex and costly than bit-width conversion.
>
> Sample rate conversion INCLUDES bit-width conversion.

All this is not to say that sample rate conversion should never be done. There are times when it is unavoidable, as a part of vari-speed (not vari-sample-rate) and when converting a recording from one standard to another. However no sane professional standard should require that it be done on a regular basis, and certainly not at the input or output of every piece of professional equipment.

Making a Choice

For processing, all these factors lead to the adoption of a professional DSD-compatible format based upon a variable number of bits at the fixed DSD sampling rate of 2.8224MHz (64FS). Internally to a piece of professional processing equipment the designer may choose any bit-width. Typically, inside professional equipment one might use 32-bits at 64FS for mixing and more than 48-bits at 64FS for equalizers.

For interconnections between processing units we use 8-bits at 64FS known as DSD-Wide. This provides ample performance for a professional interconnect at a realistic data rate of 22.5792Mbps.

For storage, the recorder designer has a choice of resolutions from professional DSD-Wide (8-bits at 64FS) down to consumer DSD (1-bit at 64FS) with easy conversion available at every step.

Analogue to Digital and Digital to Analogue Converters

Converter design is hard. Ask anyone who has tried to design a professional converter. Just when you've got it sounding right the requirements of space, power consumption, temperature, cost, reliability, line driving, jitter rejection, etc. arrive to make the whole thing even more difficult.

The choice of a professional format to feed SACD should not, if at all possible, impinge upon the freedom of the converter designer to solve all these other problems.

The short answer is that there is no "best format" with which to interface with a converter. However, some flexibility in interface design will help ease the problem.

In the world of analogue mixers, line level, defined impedance interconnects are standard. However, there is a long history of special microphone inputs from which we expect different signal levels, switchable impedance, phantom power, etc. Following this precedent it makes sense to allow ADC and DAC designers a choice of interconnection formats.

A converter interface can be provided that allows for any sample rate that is an integer multiple of 2.8224MHz (64FS, 128FS, 192FS, etc.) and any fixed weight coding (binary, offset binary, 1-of-n, differential, etc.). Some converter designers have been led to believe that all professional converters intended to originate SACD recordings must use 64FS at 1-bit. This is untrue. However, an "Integer multiple of 64FS, any number of bits" requirement for professional ADCs and DACs makes a sensible compromise. Note that acceptance of rates other than 64FS will require some sample rate conversion, but these conversions are not repeated: they occur only once, adjacent to the converter.

Professional Audio for SACD: Some Conclusions

Professional equipment for the origination of SACD recordings will eventually use a format that is significantly better than the consumer format:

The theoretical and practical criteria which influence the choice of such a professional format are well understood;

It is better to change bit-width than sample rate;

Conversion between appropriate formats is well understood and reasonably economic.

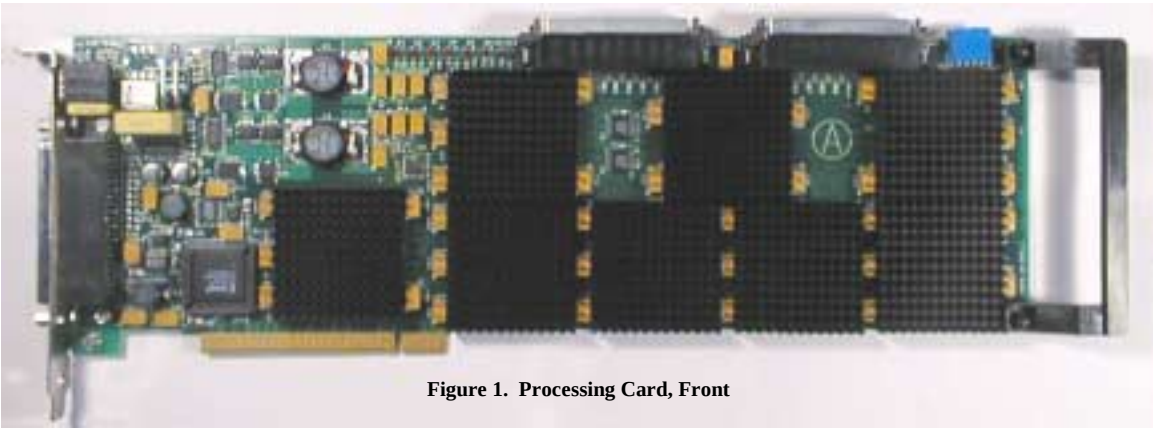

Figure 1. Processing Card, Front

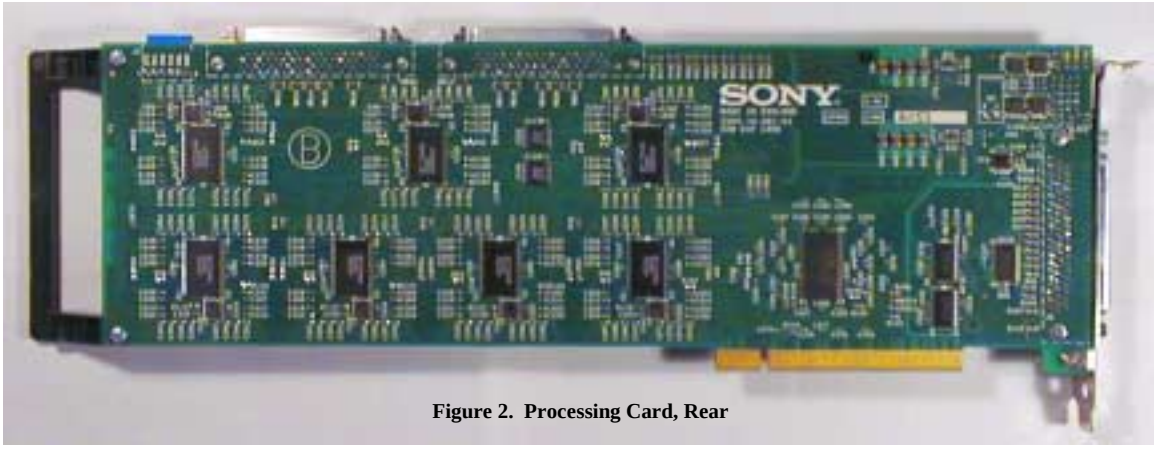

Figure 2. Processing Card, Rear

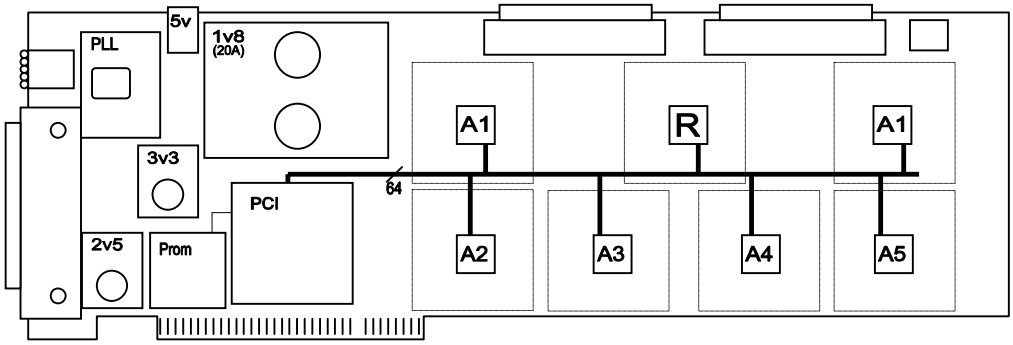

Figure 3. Processing Card, Major Components and Computer Busses

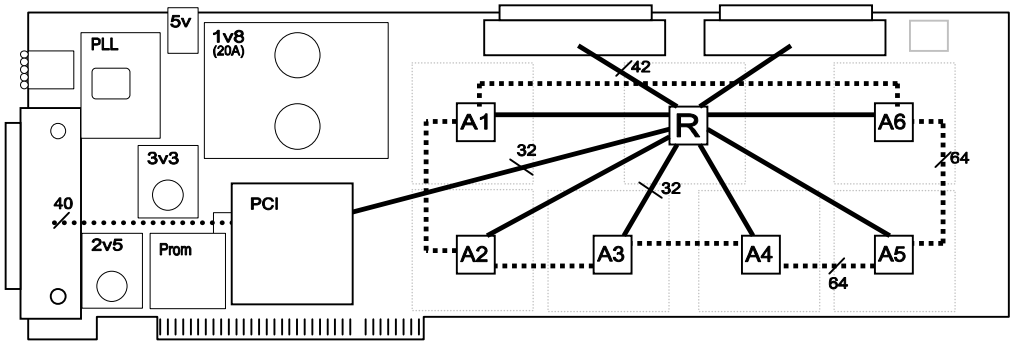

Figure 4. Processing Card, Audio Busses

HARDWARE DESIGN

The purpose of the hardware design is to provide high performance, expandable, flexible processing well adapted to the processing of DSD-Wide signals.

Card Format

The processing is based around a standard size PCI card used in a standard PC. The choice of the PCI card format has plusses and minuses. On the positive side compatible slots are widely available and the bus has sufficient bandwidth for the application. On the negative side the specified power limit of 25 watts per slot is much lower than the power density required by modern processing silicon.

Processor Choice

Following earlier work, Field Programmable Gate Arrays (FPGAs) have been retained as the major processing workhorse. These provide a flexible source of millions of gates in an affordable, available package. The number of such processors on a card is simply limited by the real-estate they consume. With present-day 680- way Ball Grid Array packages we fit seven processors onto a card together with an eighth FPGA for PCI and I/O interface.

Input/Output

One of the disadvantages of the PC style card format is its very low panel area. This makes it difficult to accommodate all the I/O connections for which one might wish. This is particularly true in the relatively new field of DSD and DSD-Wide where interconnection standards are changing with what feels like daily frequency.

There is also a serious legacy I/O problem. The vast majority of available equipment is connected via individual BNC terminated co-axial cable, one signal per cable. There is no way 40 or even 20 BNC sockets will fit on the exterior surface of a PC-compatible processor card. This problem is solved by placing a single multi-way connector on the card and providing a cat-o'-nine-tails for connection to legacy equipment. The multi-way connector itself provides 16 signal inputs, 16 outputs and eight switchable (in-or-out) connections, as well as a selection of clocks and synchronization signals. One third of the pins on the connector are individual ground connections assuring electronic signal cleanliness. The cat-o'-nine-tails cable provides connection to 20 of these audio signals and several clocks, each signal using an individual ground pin.

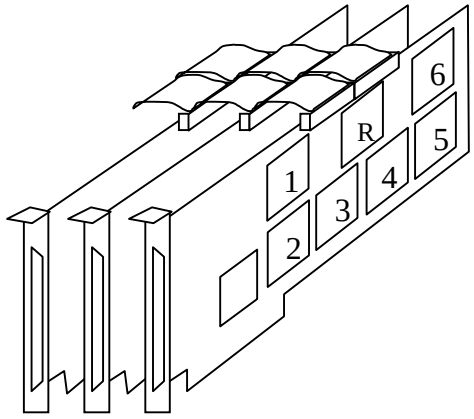

Figure 5. Inter-Card Interconnections

Inter-Card Connections

Experience has taught us that users always require more channels and more features, so the interconnection of cards to form systems is always on the cards.

In this case two internal (card to card) connectors are provided on the card edge. Each connector uses a 68-pin SCSI-style connector with a proprietary bus. The bus is composed of 21 differentially transmitted signals. Each signal may be selected under software control to be sourced or received by any card on the bus. In this manner many

different, potentially asynchronous signals of high data-rate may be transmitted simultaneously. Clock signals may also be transmitted between cards in this manner. However, the transmission of data over considerable inter-card distances at high rates normally requires the use of self-clocking codes.

Power

The power requirements of a card of this type are complex.

5V
The input power from the PCI bus.

3.3V
Used for the PCI interface chip. Required because of the high voltage signals mandated by the PCI specification.

2.5V
Used to power the I/O connections between processing chips.

1.8V
Powers the processing core of the FPGAs. This is the highest current supply on the card. Approximately 20A are available at 1.8V. This supply may also be software controlled from the PC.

The three working voltages are generated on the card from the 5V input by high efficiency switching power converters.

Thermal Factors

Thermal considerations on a card of this type constitute one of the most difficult areas. Unfortunately, although the electrical specification of a PCI-compatible card is rigorous, the thermal environment in the average PC is a complete lottery. Direction, amount and even the existence of mechanically assisted air-flow appears totally uncontrolled. Even convection cannot be counted on since mounting positions vary from case to case.

If there are no guarantees about the thermal surroundings one must do one's best to protect the equipment against the trials and tribulations of its (un)expected environment. This is done in two ways. Firstly, heatsinks are applied to all the FPGAs. Secondly, the manufacturers having kindly supplied independently accessible temperature monitoring diodes inside the FPGAs, circuitry was designed so that their junction temperatures, as well as the ambient air temperature, may be monitored by the controlling software application in the PC. This has proved useful in reassuring those users of all but the most thermally inept PCs that there is in fact enough cooling. It's also great at alerting users to stalled fans.

Pathological Program Protection

As wonderful as modern large FPGAs are they lack any effective form of protection circuitry. It is possible either intentionally or unintentionally to design a program for an FPGA which will cause it to consume masses of current. Stories abound of the effect of intentionally causing the PCI card limit of 25 watts to be dissipated in just one FPGA. The results are said to be dramatic and costly.

In order to avoid this "meltdown" situation, fully automatic circuitry is provided which, without aid of the PC, can remove the 1.8V power if any junction temperature exceeds a software-settable limit.

Bootstrap Programming

Normally one would like to make every register and memory on a card such as this accessible from the controlling computer. However, there is a PCI-related problem which prevents this. Specifically, before the PC can access anything, the PCI bus must go through its startup procedure. This means that the FPGA providing the PCI interface function must have already been programmed through its serial port. Not so much "I've locked the keys in the car," more "The only door lock is on the INSIDE of the car." To overcome this the PCI interface FPGA is programmed from an EEPROM that may itself, after its contents have been used at PCI startup, be reprogrammed for use at the next power-on cycle.

The I/O circuitry, clock routing, power control and voltage trimming, and the previously mentioned temperature monitoring circuitry are also resident on the PCI interface FPGA. For this reason major additions to I/O formats are changed by reloading this bootstrap EEPROM. This gives some protection against "rogue" programs gaining access to safety-critical systems while still allowing clock source and I/O selection by user programs.

Memory

The FPGAs used each contain 96 internal blocks of individually configurable dual-port RAM. Each block contains 4k bits which may be used in any configuration from 4k words of 1 bit to 256 words of 16 bits. Of course blocks may be used together to constitute larger memories when required.

A frequent use of these dual-port RAMs is as buffer memory between the control computer (the PC) and the audio processing. Simultaneous asynchronous access is possible from the PCI bus on one port and from the audio processing on the other.

Individual external RAM chips (on the reverse side of the card) are connected to each of the processing FPGAs. These provide an extra 512K bytes of high speed memory per FPGA.

The PCI interface FPGA is provided with 8 Mbytes of external RAM, sufficient to buffer 2.972 seconds of DSD-Wide or 23.777 seconds of consumer format DSD.

Clocks

One of the authors is well known for muttering "power and clocks, power and clocks, power and clocks ..." whenever a hardware bug is being sought. And it is certainly true that care in these two areas usually repays the effort expended.

There are several clocks active on the card.

44.1 kHz. Referred to as "FS".
Used for connecting to equipment based on the CD standard.

48 kHz.
Available for connection to standard professional PCM equipment.

2.8224 MHz. Referred to as "64FS".
The DSD and DSD-Wide sample clock. Primarily used for communication with other DSD-based equipment and internal synchronization of higher rate computational clocks.

33 MHz.
Derived directly, without a Phase Locked Loop (PLL) from the computer PCI bus. Used for all communications with the host computer, program loading of the FPGAs and data communications of coefficients, etc. to/from the FPGA dual port RAMs.

90.3168 MHz. Referred to as "2KFS".
The basic computation clock used allover the card. Almost everything runs at this frequency. This rate is 32 times the basic DSD-Wide sample rate. Nearly all internal processor computation and inter-processor communication takes place at this rate using a 32-way time division multiplex of 32 independent DSD-Wide audio signals.

180.6336 MHz. Referred to as "4KFS".
Used only internally to the FPGAs. This clock is generated by internal Delay Locked Loops based upon the standard 2KFS clock. At present this is mainly used inside experimental digital filter designs. Some standard designs may migrate to this rate in the future.

The FS related frequencies are all generated from a single Voltage Controlled Crystal Oscillator (VCXO) at 90.3168 MHz. inside a PLL. The reference for this PLL may be selected under software control from any available clock source, i.e., from the external BNC inputs or from the inter-card bus. The clock circuitry automatically adjusts to clock inputs at either FS (44.1 kHz) or 64FS (2.8224 MHz).

If no external clock source is available the VCXO can be switched under software control to the center of its nominal lock range. Thus it can provide a fully "in spec." clock frequency to other connected equipment.

DSP FOR DSD-WIDE

This section is devoted to the processing algorithms used including a single bit router, 8-bit router, conversion from single bit to 8-bit and 8-bit to single bit, equalization and filtering, dynamics processing, gain control and mixing, and metering.

DSD Router

In order to connect the DSD (64FS 1 bit) signals flexibly between the internal processing and the various possible external sources and destinations, a 128 input * 128 output router is required.

This router is implemented inside the router FPGA. The 128 independent router input signals arrive as four 32-way Time Division Multiplexes (TDM) on four wires. The 128 independent output signals leave on four similar wires, again in a 32-way TDM manner. The sample rate is 64FS, the data rate per wire is 2KFS.

Each data path for this router passes through one of four blocks of the internal FPGA dual port RAM. Each of the four blocks acts as the source for one of the output streams. The write addresses for each router dual port RAM are derived from a counter such that each of the 128 incoming signals is placed in a specific location in each memory. The read addresses for each router dual port RAM are derived themselves from data read from a router programming dual port RAM whose contents are written from the controlling computer.

The resultant 128 input * 128 output DSD router delivers a non-blocking throughput of over 360 Mbits per second. It consumes 9 percent of the RAM blocks available in the router FPGA.

DSD-Wide Router

The processing contained in the six processing array FPGAs is linked together by a DSD-Wide router in the router FPGA. Each array FPGA is linked to the router FPGA by four 8-bit wide busses, two for input and two for output. The router is also linked to the inter-card busses and to converters to and from the DSD router.

All this leads to the requirement for a 512 input * 512 output router for 8-bit signals at 2.8224 MHz. The input signals are connected via 16 input ports each carrying a 32-way time division multiplex on an 8-bit wide bus. The output signals are similarly connected.

The router uses the Time-Space-Time (TST) strategy commonly found in telecommunications switches. The route from input to output passes through three distinct layers:

First Temporal Router.
Each of the 16 input ports is connected to a double buffered, dual port memory. The input address is derived from a counter, the output is controlled by a program RAM loaded from the control computer. This arrangement allows complete, non-blocking reordering of the TDM output under program control.

Spatial Router.
The 16 outputs from the first stage router are fed to a collection of sixteen 16-input multiplexers. The outputs of these multiplexers form another 16 TDM busses. Each of these multiplexers is individually controllable at every TDM cycle by program memories loaded from the control computer. At each of the 32-clock cycles the array of multiplexers forms an independently programmable non-blocking 16 input, 16 output router.

Second Temporal Router.
The outputs of the spatial router feed another temporal router identical to the first one.

In this sequence of routers all three are individually non-blocking. However when the requirements of one-to-many connections are added the total router is unfortunately not non-blocking. To overcome this all three routers are "doubled". The input temporal router is doubled to have 32 output busses instead of 16, the spatial router is doubled to have 32 input busses and 32 output busses and the output temporal router is doubled to have 32 input busses.

The assignment of paths through this central router is achieved at run-time in response to the varying demands placed on the system by the user.

DSD to DSD-Wide Conversion

Any single-bit DSD signals that need to be processed are first converted into the internal 8 bit DSD-Wide representation. This could, as outlined above, be accomplished by simple substitution of longer words for shorter ones, say 01111111 for a positive sample and 10000001 for a negative sample. However, we take advantage of this conversion to insert some gentle out-of-band low pass filtering at this point. The filter used is a short, symmetrical, 34-tap FIR.

```
-1 –2 –3 –4 –2 0 2 4 5 6 7 9 11 13 15 14 14
14 14 15 13 11 9 7 6 5 4 2 0 –2 –4 –3 –2 -1
```

The total impulse response length of this filter is only slightly over that of a 96kHz sample period. The frequency response of this filter differs from flatness by less than 0.002dB over the frequency range to 20kHz.

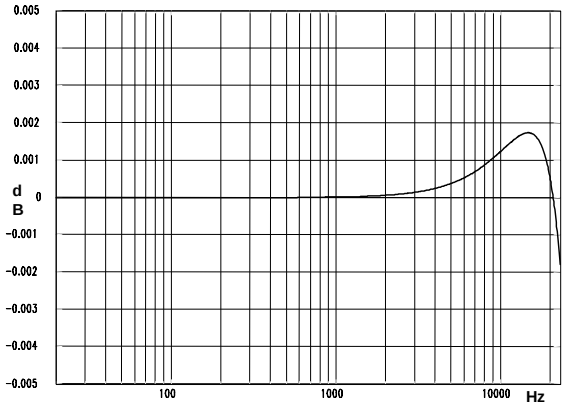

Figure 6. Frequency Response of 1-bit to 8-bit Conversion

It should be noted that there is no low pass filtering in any of the following processing. None of the mixing, gain control, dynamics processing, etc. contain any implied low pass filtering. The only filtering in an equalisation or filtering stage is that required by the user. Again there is no mandatory low pass filtering.

The length and tap weights of this filter are specially chosen to yield a sum which can always be represented within the 8-bit DSD-Wide range without truncation, dither or further processing.

As can be seen in Figure 7 this filter can be constructed as a simple, sparse, 1-bit FIR with very small coefficients followed by a double integration. This leads to a very economical implementation.

64 instances of DSD to DSD-Wide conversion are available inside the router FPGA, they are permanently connected between the DSD and DSD-Wide routers.

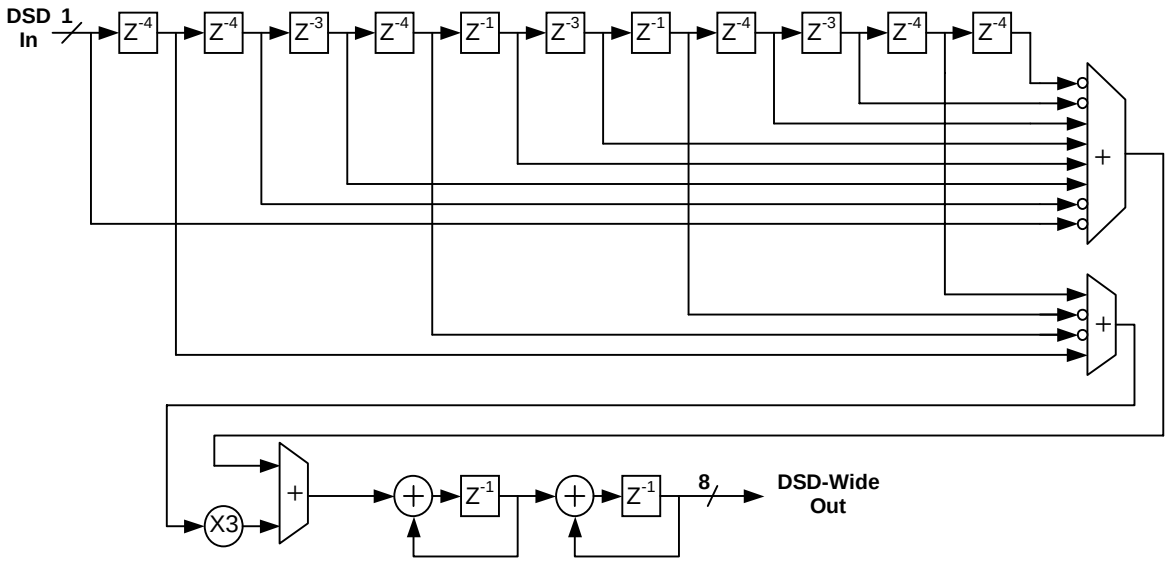

Figure 7. DSD to DSD-Wide Conversion

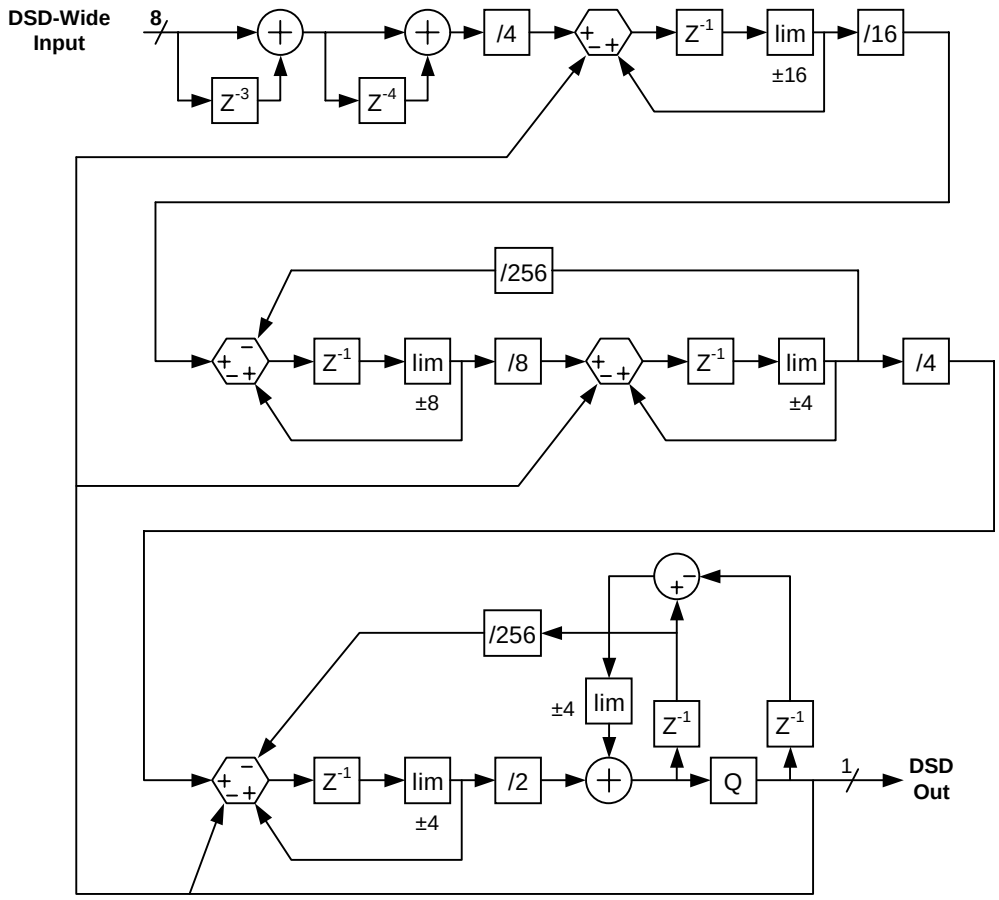

Figure 8. DSD-Wide to DSD Conversion

DSD-Wide to DSD Conversion

The conversion from 8-bit to 1-bit representations takes place using a fifth order Delta Sigma Modulator shown in Figure 8.

The frequency response of this conversion differs from flatness by less than 0.007dB over the frequency range to 20kHz.

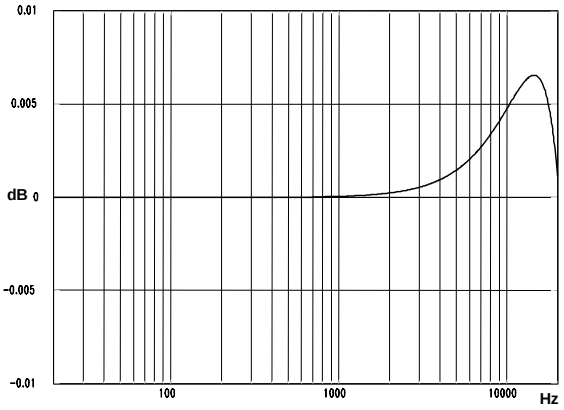

Figure 9. Frequency Response of 8-bit to 1-bit Conversion

Note that this conversion only occurs at the output of the processing. All internal connections between processing elements use the 8-bit interface which involves no filtering.

64 instances of DSD-Wide to DSD conversion are available inside the router FPGA. They are permanently connected between the DSD-Wide and DSD routers.

Gain Control and Mixing

All gain control and mixing functions take place in a single FPGA. This provides a maximum of 512 interpolated, gain controlled cross-points feeding a maximum of 64 outputs from 64 inputs.

Each actual cross-point involves the multiplication of a 24-bit interpolated coefficient by an 8-bit DSD-Wide sample yielding a 32-bit result. This action takes place in sixteen individual processing slices.

These slices are connected to two common input busses each carrying a 32-way time division multiplex of input signals, thus making all 64 of the FPGAs input signals available to each slice. These signals are fed to a double-buffered 64-location internal RAM (one per slice) which serves as a 64-input 32-output router. A central interpolating engine (not shown) supplies a stream of 512 fully interpolated 24-bit coefficients.

The 32-bit outputs of the multiplication process are summed in an accumulating 512 input 64 output router. A limiter is provided after the accumulator to restrict the final sum of products to a 32-bit number. Activation of this limiter is reported back to the controlling computer and thence to the user.

Note that in this whole process of generating the sum of products of various signals and coefficients no truncation or loss of precision ever occurs. The resulting 32-bit number is numerically exact.

In any digital audio processing system there comes a time when the larger bit-width produced by processing must be restored to the nominal value. It is most important that this is done correctly. In the case under consideration the 32-bit sum of products is restored to the 8-bit DSD-Wide format in a correctly dithered fifth order noise shaper shown in Figure 10.

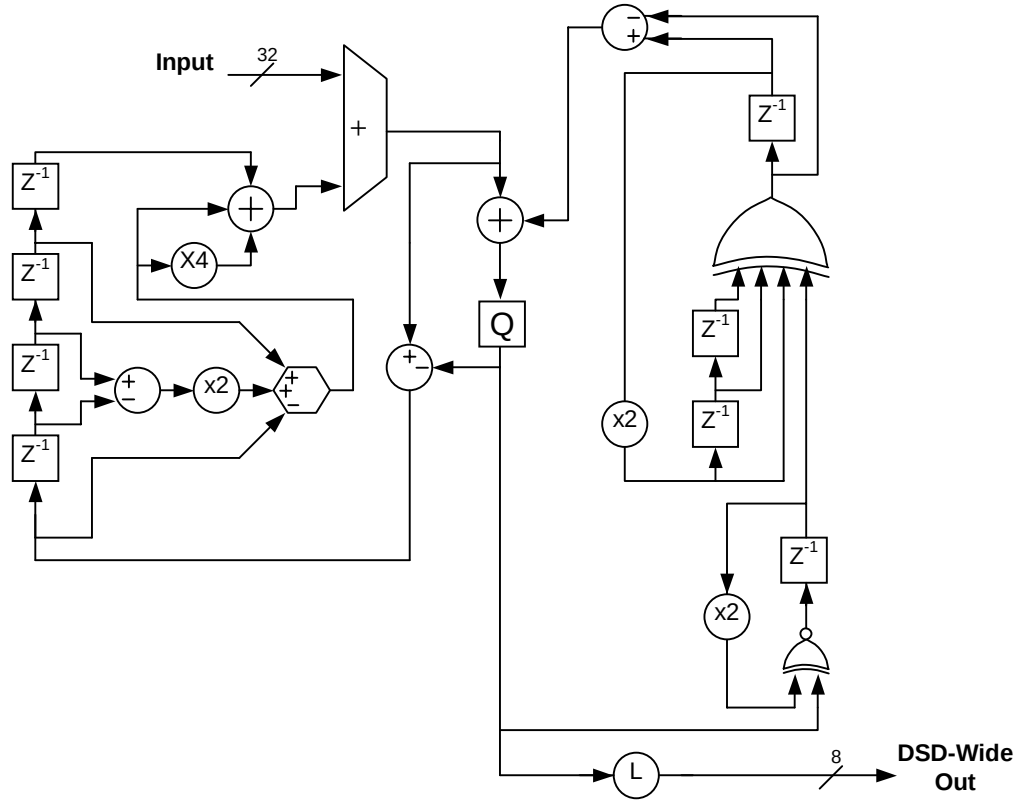

Figure 10. The Noise Shaper Used in the Gain Control and Mixing

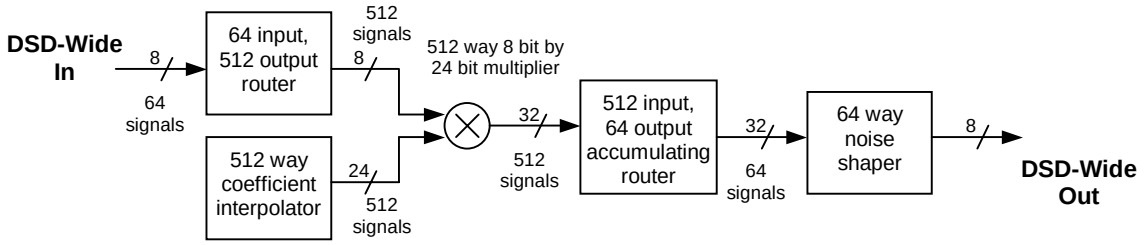

Figure 11. Overall Signal Flow of Gain Control and Mixing

Filtering and Equalization

The filtering and equalization requirements are provided by 256 second order sections implemented on two FPGAs.

Five 48-bit coefficients are used to implement a standard second order filter applied to the 8-bit DSD-Wide input and output signals. The 640 required fully interpolated coefficients are provided by an on-chip coefficient engine. The 56-bit outputs of the multipliers are summed without loss of precision and the resulting 60-bit sum is applied to an embedded fully dithered noise shaper. There is no un-dithered or un-noise shaped bit-width reduction. Two independent limiters are provided, one in the feedback loop, the other in the output. This assures that output limiting occurs before internal limiting. It should be noted that the noise-shaper imposes no delay on the signal, thus the five filter coefficients may be calculated exactly as though no noise-shaping was being applied.

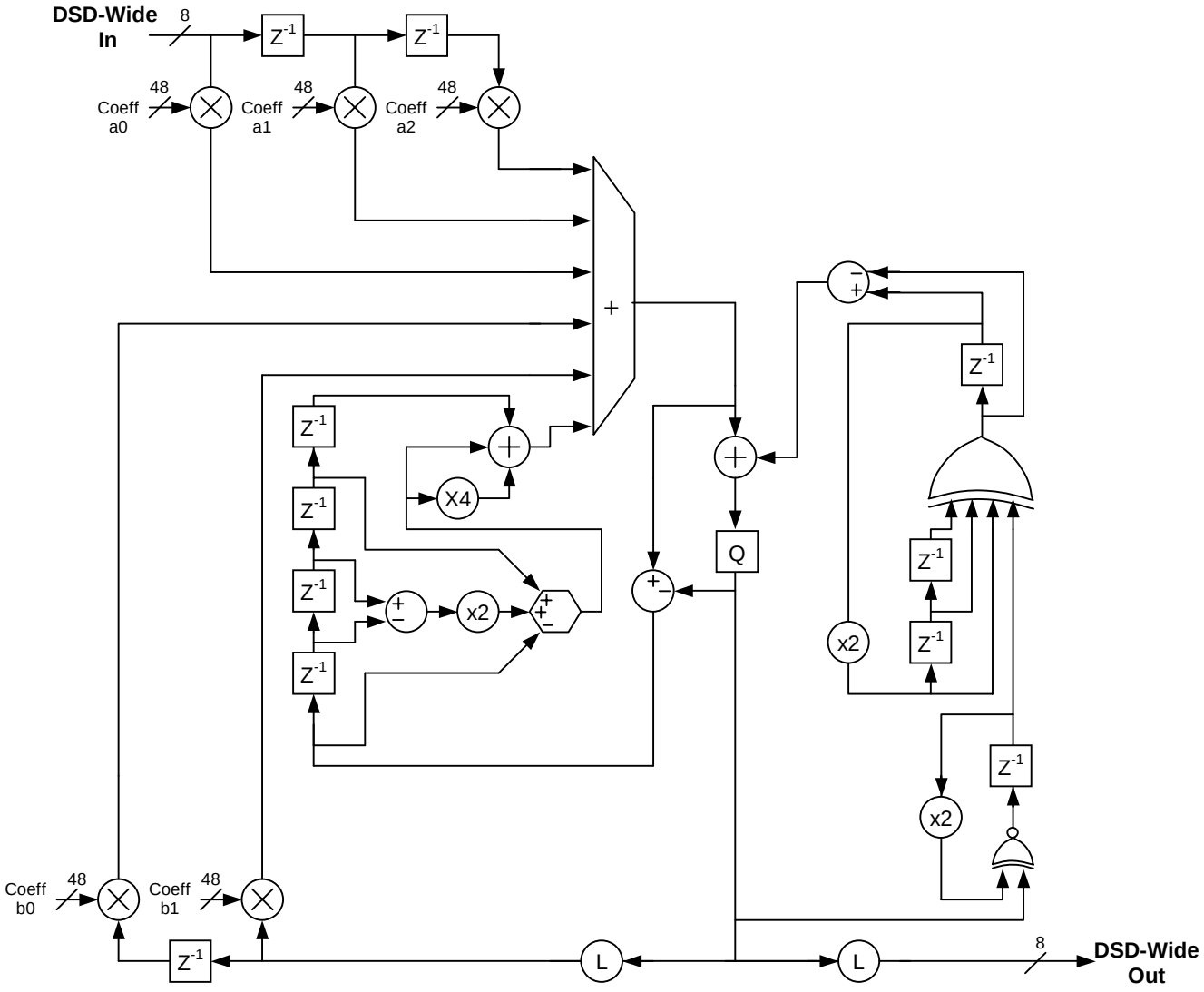

Figure 12. Typical Combined Second Order Section and Noise Shaper Used for Equalization

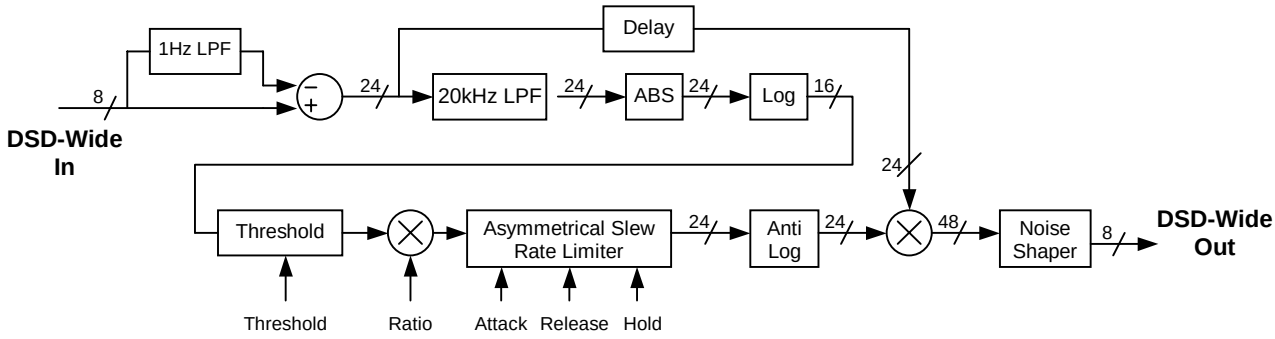

Figure 13. Dynamics Processing

Dynamics Processing

One of the array of FPGAs is used for the dynamics processing of 32 DSD-Wide signals. The basic signal flow is shown in Figure 13.

A low pass filter is used to gain a 24-bit representation of the DC component of the input signal. This is subtracted in order to provide a DC-free signal to the following circuitry. Both the side-chain and the signal path would be disturbed by the presence of DC. Note that this method of DC removal leaves the original 8-bit signal unmolested, in the sense that no truncation or other nasty processes are employed.

The side-chain processing begins with a low pass filter used to gain a multi-bit approximation to the input signal. This multi-bit signal is then treated in a manner similar to that seen in any other dynamics section, but at a higher sampling rate. The resulting gain control signal is multiplied by a suitably delayed copy of the input signal. Again note that no bit-width reduction is applied to the signal until a final fully dithered noise shaper is used at the output of the dynamics processing. This noise shaper is a 48-bit version of that shown in Figure 10.

Metering

The metering of DSD signals involves filtering the incoming signal into specified bands and the application of envelope followers and peak detection circuits to the resulting band limited signals as shown in Figure 14.

A SMALL MASTERING MIXER

The hardware and DSP outlined above has been used to create a small, 8-channel mastering mixer for the production of stereo and multi-channel SACDs. A screen shot of the mixer is included in Figure 15. The mixing, equalization, routing, dynamics and metering required by this mixer are contained in six of the seven processing FPGAs on the card as described. Now what shall we do with the one remaining?

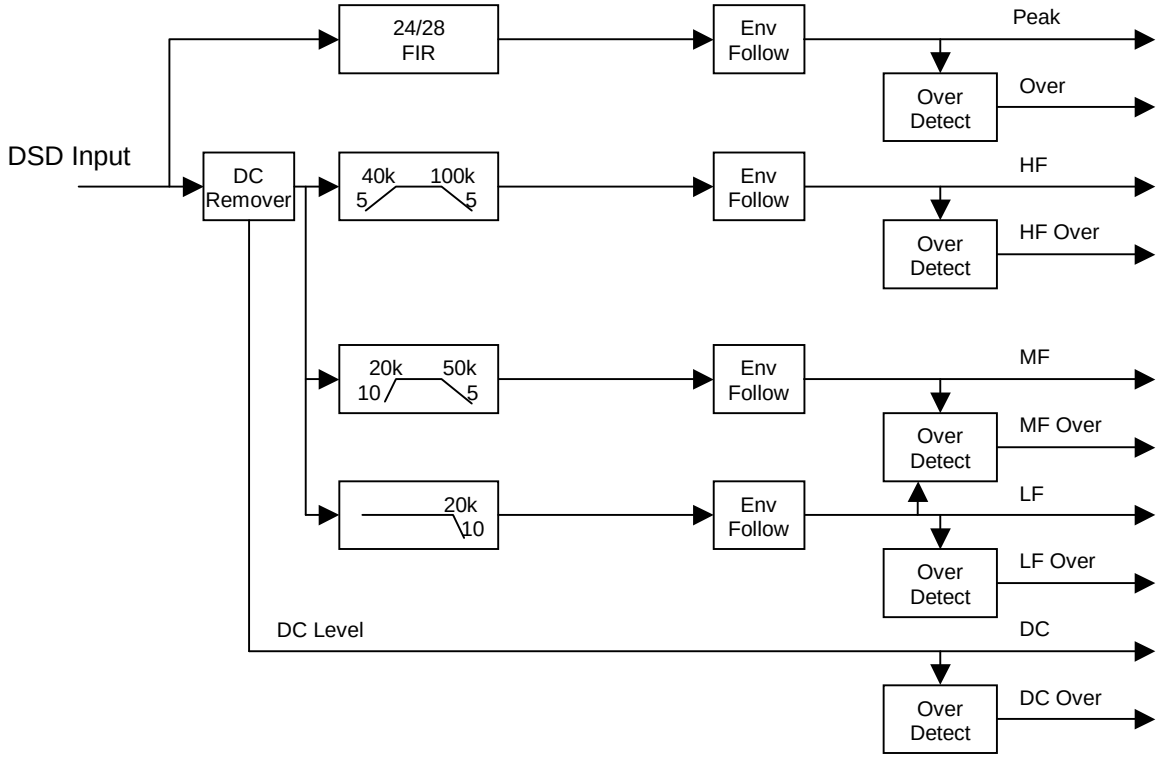

Figure 14. Metering

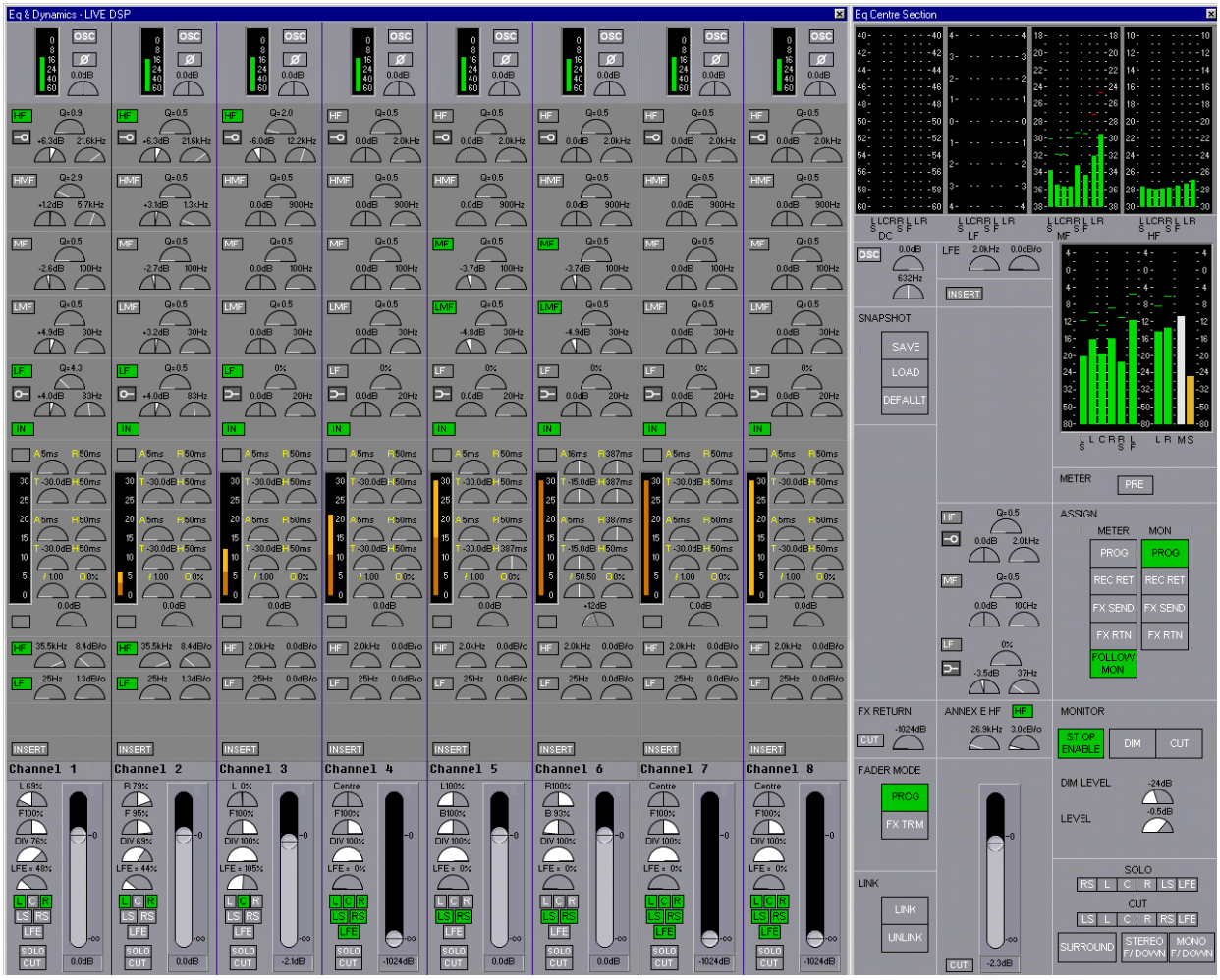

Figure 15. An 8-Channel Mixer Using the Hardware and DSP Described

CONCLUSION

This paper has presented the reasons behind our choice of signal representation, the design of flexible hardware to support this representation and practical recipes for the processing of DSD-Wide [64FS 8-bit] signals. These signals are fully compatible with the DSD [64FS 1-bit] signals used by the SACD consumer audio format.

We have shown that professional audio processing at 8 bits at 2.8224MHz can achieve all that is presently required to serve a release format of 64FS 1-bit. Such 8-bit processing is shown to be feasible within a small number of modern integrated circuits, typically fewer than one chip per channel.

ACKNOWLEDGMENTS

The authors owe a special debt of gratitude to Paul Frindle for his work on the voltage conversion circuitry, without which we would be truly powerless.

We would like to thank the engineering staff at Xilinx both in the UK and the US for their technical support; International Rectifier for the power transistors; Advanced Layout Solutions for squeezing everything onto the card and Total Interconnection Solutions for putting the chips on the cards the right way up.

We cannot list the dozens of SONY engineers and managers on three continents who have helped the project in one way or another but those in Tokyo, Walnut Creek, Boulder, New York, Amsterdam, Basingstoke and Oxford know who they are.

Our good friends in Eindhoven also deserve our thanks.

We would also like to thank James Angus for his intellect and good humour.

References

[1] P. C. Eastty, Sleight, C., Thorpe, P.D.:
“Research on Cascadable Filtering, Equalization, Gain Control, and Mixing of 1-Bit Signals for Professional Audio Applications,” 102nd AES Convention, Munich, 1997.