ADAM - A 64 CHANNEL GENERAL PURPOSE REALTIME AUDIO SIGNAL PROCESSOR

Dr. Robert Trausmuth, Michael Kollegger

IT Department
University of Applied Science Wiener Neustadt, Austria
geo@fhwn.ac.at

ABSTRACT

In this paper we introduce a 64 channel audio processing unit made in our department. The audio processor uses a 16 bit control unit (Infineon XC 167) with ethernet interface running a realtime operating system and two Analog Devices ADSP-TS101S high performance tigerSHARC DSPs for audio stream processing. The first project implemented on this equipment is a 64 channel in, 32 channel out audio mixer with a sampling frequency of 48 kHz (leaving another 32 channels for effect feedback loops) or 96 kHz, alternatively. The audio processor is fully remote controllable via TCP/IP.

1. INTRODUCTION

Teaching system programming is done best with a real world application. A sound system produces huge amounts of realtime data and the results can be heard immediately. So we started the 64 channel audio mixer 2 years ago from scratch. Every piece of hardware (except for the processor boards) was done in our labs. Protocols and system architectures had to be designed to meet the requirement of creating a fully remotely controllable audio mixer. The control is done via standard TCP/IP network, and the control console can either be a standard-mixer like console with faders and switches or a simple software application. Using a wireless LAN access point even gives the possibility of using handheld devices for stage personnel, allowing to adjust monitor channel settings directly at those devices with no need to contact the monitor sound engineer via walkie-talkie.

The mixer itself was built on a very general concept, leaving us the possibility of driving up to 256 output channels. For financial reasons we decided to implement only 32 output channels and use another 32 as feedback loops for effect calculations running on the second DSP.

Using ADAM means finally getting rid of multicore cables, since no audio data needs to leave the stage, although in the first implementation there is still the need of one audio line for the talkback monitor and one for the headset.

2. SYSTEM DESCRIPTION

The central player is a XILINX Spartan IIe 300 FPGA which handles all the data traffic and synchronizes the system components. It works with two different clock speeds to meet the needs of the CODECs and the DSP data ports. Communication with the XC 167 is done via dual port RAM to meet the different update speeds of slow control data and to avoid a third clock domain in the FPGA. Audio data is sent to and received from the CODECs in 32 bit words (24 bits significance) per channel at 48 kHz sampling rate.

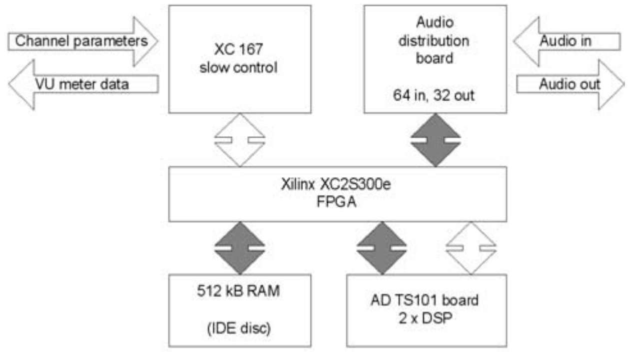

Figure 1: *Audio processor block diagram.*

The total amount of audio data handled by the FPGA is roughly 10 MB per second.

2.1. Data Flow Conceps

ADAM is contained in a 19 inch standard stage box. Communication with the outer world is done by Ethernet and TCP/IP protocols. Channel parameters like filter settings and fader controls are sent via TCP connection. VU meter data is provided by the stagebox via UDP broadcast telegrams. After implementing a realtime ethernet layer there will be 2 - 8 channels of realtime audio data for transmitting headset and talkback signals to the remote control. Another 8 channels will be available for transmitting output signals via standard WLAN equipment to distributed active speaker stations.

The first prototype also contains a touch panel directly on the stage box front side to check the function of the XC 167 module and to execute a variety of testing routines directly on the stage box.

2.2. Slow Control

The XC 167 handles all the slow control calculations. It is responsible for keeping up to 16 virtual audio mixer setups. Each setup can control 3 stereo main mix outputs with different delay times, 4 mono monitor channels and 8 stereo subgroups, which gives a total of 26 output channels. Of course not all of the channels have to be mapped to a real output channel.

Every time a parameter changes, the mixer matrix is recalculated and the new values are stored in the dual port RAM. The complete set of parameters is transferred to the DSP every 50 ms. At

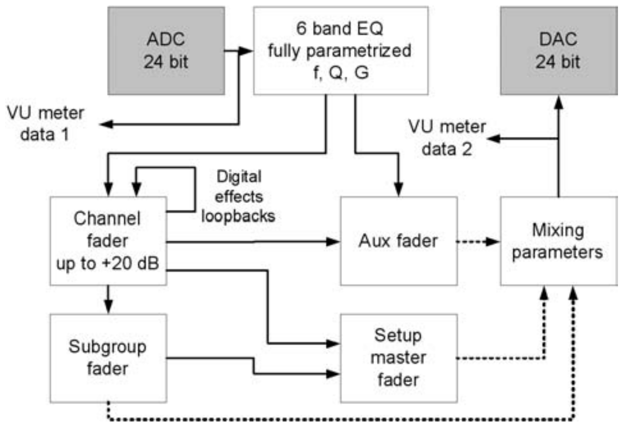

Figure 2: *The symbolic audio data path.*

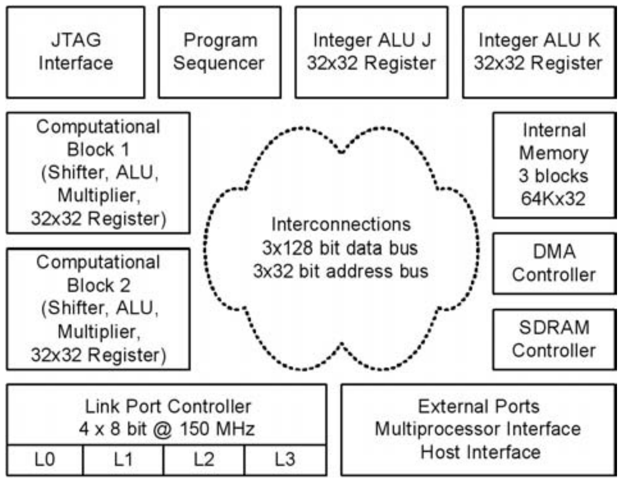

Figure 3: *ADSP-TS101S block diagram.*

the same time the DSP reports the last peak values to the slow control entity, which in turn broadcasts the signal values to all the remote controls. As shown in Fig. 2, audio data is first passed through the six band equalizer. The final mixing parameters are calculated depending on the selected signal routing. After the main channel fader the signal can be routed via one or more subgroups or sent directly to the master output. Each subgroup can be mapped to direct outputs (DACs) or to the master output. Four distinct AUX channels per setup can be routed pre or post the channel fader. The VU meter calculation is done directly after the ADC to check the input signal strength or directly after the mixer matrix, where the actual signal strength is available (per channel or for the subgroup/master sums).

Since only one DSP is working with the mixing parameters, the second DSP can be used for side-calculations like audio effects. A special parameter set gives the opportunity to set up 32 different audio effect chains which can be calculated in real time. This section of the audio mixer is currently under development.

The XILINX FPGA uses a 16 bit wide dual port RAM for communication with the XC 167. The DSP board is connected via one special synchronous 8 bit data port of the TS 101.

2.3. Audio Stream

Analog audio signals are sampled by high end 24 bit AD 1854 and AD 1871 CODECs. The regular sampling rate is 48 kHz, but this can be changed to 96 kHz if both DSPs are used for audio mixing calculation. The XILINX FPGA generates the sampling clock signal and synchronizes all other components. Data exchange with the DSP board is done via another of the 8 bit synchronous data ports of the TS 101 at 62 MHz clock speed. On the DSP side the data is handled by DMA transfers on the chip internal 128 bit wide data bus. This leaves nearly all the CPU time for the realtime calculations.

Sampling and transferring the data happens during one clock cycle. The second clock cycle is used for calculations. The third cycle has been inserted as idle cycle and will be used for audio effect calculations later on. During the first half of the fourth clock cycle, data is transferred back to the FPGA and another set of data is fetched into the DSP for calculation. The second half of this clock cycle is used to write the output data back to the CODECs. Two different memory segments are used to implement a working pipeline with a total of four steps. This gives a total of 83 μs response time per sample.

The audio mixer itself implements a 6 band fully parametrized equalizer for each input channel, followed by the 96 in / 64 out mixing matrix. Audio data is converted to floating point format prior to entering the input filter and reconverted to 24 bit integer after all calculations are finished. All calculations in the DSPs are done with floating point numbers.

Data for audio effect calculation is transferred via DMA to the second DSP. To grant more time for the effect calculations, the designated effect loop channels will be dealt with at the beginning of the calculation stage and transferred to the second DSP while the calculation on the "real" audio channels is done.

2.3.1. 6 Band Equalizer

Each input channel has its own 6 band equalizer with independent parameters (shelving and peak filters, [1]). The settings for each filter (cut-off frequency f_c, gain G and factor $Q = f_b/f_c$) are converted to corresponding parameters for calculation by slow control and sent to the DSP. The DSP code has been optimized for the TS 101 processor architecture, fully utilizing the internal 128 bit data busses and the two independent computational units. Calculations are done in IEEE floating point format with single instruction multiple data operations. This special mode is used for calculating 4 channels at the same time. The optimization result is documented in [3]. Starting with a textbook application using roughly 62 μs, the optimization of the code leads to a final execution time of 3,22 μs for all 64 channels. Since the equalizer is realized as six stage second order IIR filter, careful simulations have been done on the parameter ranges for the filter coefficients. In his diploma thesis Franz Siegmeth has also proven the validity and possible implementation of these filter banks.

2.3.2. Mixer Matrix

The mixer matrix itself is implemented using an adapted FIR filter algorithm. The coefficients are calculated by slow control and actualized every 50 ms. The slow control microprocessor uses a three stage multiplication scheme to get the right values: first there are the per channel faders. Channels can be combined into subgroups (8 groups stereo). Subgroups can be configured as real output channels or again mixed to the main output. In parallel, every channel can be a part of the main output without belonging to a subgroup.

The matrix calculation starts with the 32 effect channels, which are subject to DMA transfer to the second DSP right after the calculation has finished. The calculation of the 32 output channels follows immediately.

For the main signal output channels (max. 3 pairs) there is a special feature in the output chain. Each main signal pair has its own volume control and delay line. The main signal can be delayed for up to 500 ms to compensate for sound wave propagation in a distributed speaker environment.

2.3.3. VU Meter

After the matrix calculation there is another series of calculation which gives the VU meter data sum for each channel. This sum is sent to all mixer terminals every 50 ms via UDP broadcast. The parameters for this calculation can be updated every 50 ms. This feature can be used to display the de facto contribtion of each channel to the main signal, in which case the calculation parameters are the ones used in the matrix mixer section for the main output. When adjusting the input gain, the factors are changed to the channel fader coefficients. If the fader is set to 0 dB the input gain can be adjusted using the gain control.

For providing the standard VU meter display the 32 output channels are monitored, too, but without any calculation parameter. Therefore the real signal strength can also be drawn from the VU meter broadcase package.

2.3.4. Audio Recording

Current development deals with the implementation of an IDE interface on the FPGA. Using ATA-2 (EIDE, 16 MBps) or even ATA-3 (Ultra ATA, 33 MBps) standard, the amount of realtime data should be no problem even if the recording is done uncompressed. Once this interface is included in the audio data stream, a control instance can use it to record the 64 channels of live data on hard disc. The FPGA can be used as source control and play back the recorded data to the DSP, making no difference to the live performance. Using this approach gives the possibility to record a live concert and use the raw sound data later in a studio the remix the recorded tracks for studio production.

2.4. Hardware Setup

The audio frontend is organized in 8 input / 4 output channels per unit. The audio signal enters a pre-ADC section where phantom power can be applied if necessary. Mic and Line signals can be sampled because prior to the ADC there is an analog gain controller (digitally controlled via a feedback DAC). The ADCs are chained together, so the ADC clock of 12,288 MHz gets 8 x 32 bits out of the chain per sampling interval. Data is left justified, so there is time enough after sampling the last channel to transfer the data to the DSP. Data transfer starts in the middle of the sampling clock cycle. This point is used for synchronization, since the DSP initializes its DMA channel after finishing the backtransfer of the output data and remains idle until new data is available. The

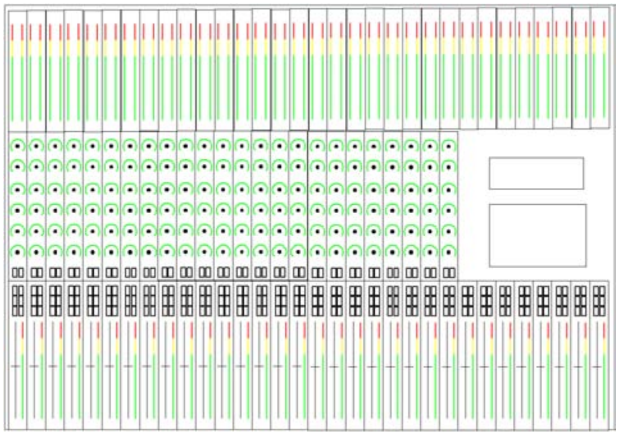

Figure 4: *The live performance mixing console.*

possible data transfer rate to the DSP in this setup is 62 MBit x 8 per second, the ADC data rate is 12,288 MBit x 6 (one byte out of 4 is always 0), which gives enough safety margin for the DMA transfer.

DACs cannot be chained together, so each of them gets only two channel's data. All DACs are set with the same clock signal, so each output channel has the new data available at the same time (which is trimmed to the end of the sampling interval). Following the DAC, there is a symmetric signal distributor, so the audio signal can be used either in a symmetric or asymmetric way, in which case the second line is shortcut to ground. The chip will rise the asymmetric signal level accordingly.

2.5. User Interface

We have two different approaches to the user interface. The first goal was to keep the traditional mixing console feeling for live performances. Thus we built a microcontroller based control system which looks familiar to the sound engineer [6]. On the other hand, we developed a computer application which can be used in an audio studio and provides full access to all the ADAM features. The application can be used during live performances to restore certain scene settings like fader setups or even filter and level setups. All those settings can be restored selectively so that the interference with live settings can be minimized. Since each fader module is an independent system controlled by the mixing console master (another XC 167), channels can be assigned to each fader as needed. The mixing console itself can store up to four different fader layouts. Using the software application, this number can be extended as needed.

Both application and hardware user interface communicate with the stage box over TCP/IP connections. A special protocol has been developed to meet the needs for exchanging setup and slow control data. The protocol has been prepared to include further extensions like transferring live audio data using a realtime ethernet protocol extension. The mixing console has a VU meter bridge to show all 64 input channels with the corresponding fader settings for the selected mixing console setup. The fader and control displays are updated by the stagebox so that more than one mixing console can be used for the same output setup. Faders and controls can be moved once their value does not longer correspond to the actual stage box setting, and they have to "hook into" the actual value again before their movement will be sent to the stage box. This way we can avoid jumps in volume or filter settings even if more than one mixing console (or restored settings from the software application) have been applied to the stage box.

3. RESULTS

All parts of the system have been evaluated with realtime measurements. One sampling slot has about 21 μs. The FPGA uses the full slot time for sampling 8 x 8 channels of ADCs. During the last 650 ns of the time slot, there is time enough to transfer the last 8 channels to the DSP, so the calculation of the mixing stage starts nearly synchronously with the next sampling slot.

The mixing stage calculation uses 3,2 μs for the input amplifier (including the int to float conversion), leaving 16 μs for the matrix mixer. The calculation of the VU meter data uses another 1 μs, leaving nearly 1 μs as safety margin to cope with DMA transfer delays.

The second DSP is not used for the basic audio mixer setup and will be included in ongoing work to do some effect calculations on 32 extra channels.

4. CONCLUSIONS

ADAM is a powerful audio processing setup which can be adapted to other tasks as well. There are many ideas of further improvements, like 64 channel hard disc recording or realtime audio data transfer using the standard ethernet link to the remote control. The setup can also be altered to support less input channels but up to 256 output channels, which would be a great setup for wave front synthesis.

Ongoing developments are now dealing with realtime data compression which can be used for the hard disc recorder. Another project starting soon will be the development of an 8 channel audio WLAN box which can support active speaker setups with one of 8 channels coming from the ADAM mixer.

5. REFERENCES

[1] Udo Zölzer, *DAFX Digital Audio Effects*, Wiley & Sons, 2003, ISBN 0-471-49078-4

[2] Udo Zölzer, *Digital Audio Signal Processing*, Wiley & Sons, 1999, ISBN 0-471-97226-6

[3] Franz Siegmeth, *Realisierung eines digitalen Mischpultes mit DSP*, diploma thesis, 2003

[4] James Noble, Charles Weir, *Small Memory Software, Patterns for systems with limited memory*, Addison Wesley, 2001, ISBN 0-201-59607-5

[5] John Watkinson, *The Art of Digital Audio*, Focal Press, 2001, ISBN 0-240-51587-0

[6] Michael Kollegger, *Implementierung von Echtzeitapplikationen zur Kommunikation zwischen Geräten für Audioanwendungen*, diploma thesis, 2003