The Frox Digital Audio System

D. Gareth Loy, Daniel J. Freed
Frox, Inc.
1450 Kifer Rd., Sunnyvale, CA
dfreed@frox.com, dg1@frox.com

Abstract

The Frox Digital Audio System (FDAS) provides ultra-high-fidelity presentation of digital audio. A Video Preprocessor (VPP) receives TV audio and the outputs from analog audio devices and converts them to digital signals, and passes them on to the Media Processor (MP). Digital audio devices such as CDs, Laserdiscs, etc., can be attached to the MP directly via fiber-optic or coax cables. If requested, the FDAS applies pure digital signal processing (DSP) effects such as Dolby ProLogic™, Tone controls, graphic equalizer, hall simulation (the Sonic Playground™) and THX™. DSP is performed by an array of up to 16 Motorola 56000’s executing synchronously at the instruction and sample levels, connected by a 24-bit wide digital crossbar switch move-engine.

The processed digital audio from the Media Processor can then be routed to the analog outputs of the VPP (to be connected to power amplifiers and speakers), or the Media Processor’s fiber-optic outputs can be connected to digital-input amps and analog speakers, or digital-input powered speakers.

In the case where the digital output of, for instance, the Frox CD changer is attached directly to the Media Processor, and the output of the Media Processor is attached to Frox loudspeakers, the signal remains in the digital domain from the audio source to the speaker.

1 Architectural Overview

Eight stereo digital signals can be received via coax or fiber-optic connections. Additionally, four channels of sixteen-bit 64-times oversampled analog-to-digital conversion are used only where analog audio sources are connected to the FDAS, otherwise digital signals are used directly. Sixteen stereo digital fiber optic output devices can be connected. Additionally, four channels of 18-bit Digital-to-Analog conversions are available.

Once an analog signal is digitized, there are no further internal analog routing or processing steps whatever: it is entirely digital throughout. Where optical-input powered speakers are used, the signal remains digital out to the speaker.
Frox uses Motorola 56001 Harvard-architecture signal processing computers to perform digital effects. This processor combines fast, pipelined operations, parallel data buses, with a general instruction set optimized for digital signal processing. The 56000's provide Frox with a completely general-purpose signal processing platform. This means that Frox can support any conceivable form of audio processing application merely by changing the software.

Using programmable DSPs has allowed Frox to develop and fine-tune complex audio algorithms while mostly avoiding the time-consuming and expensive chip-fabrication process.

For simple tasks like tone controls, one DSP is enough. For complex processing like ProLogic™ or THX™, multiple DSPs are required. It is a tough design problem to know how to interconnect multiple DSPs so as to allow them to work efficiently together and yet maintain the high degree of flexibility required in order to support many types of complex processing.

Frox uses a "Stream-oriented" architecture which allows arbitrary mapping of signals among DSPs while requiring only 3% of the DSP's processing power for inter-DSP synchronization. Signals can be routed arbitrarily from input devices, directly into or out of the DSPs, or written to output devices. Two custom ASICs engineered at Frox (the FSEQ and FDA1 chips) provide all interconnectivity.

The FDAS architecture also allows for on-screen controls on the graphic user interface, such as volume control, to change parameters in the DSPs, and for on-screen displays, such as VU meters, to read data from the DSPs.

Digital audio data paths in the FDAS are never less than 24 bits wide, so the dynamic range is greater than 144dB. There is no discernable degradation of signal quality because of limited-resolution data paths. Arithmetic operations inside the DSPs are performed with 56 bits of precision, which yields greater than 336dB of dynamic range for intermediate calculations.

Each DSP can perform 226 computations (e.g., multiply, add, subtract, test) per sample period. There are 44100 audio sample periods per second for each audio source. Hence there are 44100 x 226 = 10,000,800 operations per second per DSP. Up to 16 DSPs can be installed in a Media Processor, resulting in a very respectable figure of 160,000,000 operations per second for the system overall.

Another advantage of the combination of general-purpose DSP and arbitrary routing of audio signals is that the customer need buy only as much DSP
processing as is required for his or her needs. The system comes with four
DSPs, and is expandable up to 16. As a customer's needs expand, Frex can sell
that customer additional DSP power.

2 Software

Our main goal in designing the FDAS software was to create a flexible and
extensible environment for developing audio DSP applications while still using
the processors as efficiently as possible, especially minimizing system overhead
in the DSPs. The FDAS signal processing software is all written in 56000
assembler for maximum efficiency. All other audio software, like the majority of
software on the Frex, is written in C++.

Our data structure for representing arbitrary interconnections of source devices,
sink devices, and processing elements is called an AudioConfiguration, which is a
directed graph of source devices, sink devices, and processing elements. An
individual node is called a Component, which contain input/output points for
signal flow and control/display points for parameter data flow (usually
asynchronous and event-driven).

There are three subclasses of Component: AudioDevice, SimpleComponent, and
StructuredComponent. An AudioDevice represents an audio source or sink
device and contains information such as the physical addresses of the device's
connectors. A SimpleComponent represents a processing element that requires
no more than one DSP's worth of processing power; it contains a module of
assembled 56000 code in relocatable form. A StructuredComponent represents a
processing element that requires more than one DSP's worth of processing
power; it contains a directed graph of SimpleComponents.

Whenever the user requests a change in routing or processing, the current
AudioConfiguration is halted ('deactivated'), edited, and then restarted
'activated'). The primary work of activating an AudioConfiguration is to get all
of the SimpleComponents (including those within StructuredComponents) loaded
and running on DSPs. The FDAS software permits arbitrary combinations of
numbers and types of SimpleComponents to run simultaneously, in keeping with
the design goals of flexibility and extensibility. As a result, the software must
assign SimpleComponents to DSPs at run time, rather than permanently
reserving a given DSP to perform a given type of processing. We refer to the
process of assigning SimpleComponents to DSPs as 'packing'.

In our experience, the most limited DSP-related resource is processor bandwidth.
When possible, SimpleComponents which don't use an entire DSP are co-loaded
onto the same DSP to share the processor's bandwidth. The co-loading is handled automatically by the system software.

Given the list of SimpleComponents and their cycle counts, the FDAS software determines a packing scheme which is optimal with regard to utilization of available DSP cycles. The 56000 code associated with the SimpleComponents is then link-edited and downloaded to the DSPs. Linking and loading is performed in such a way that the SimpleComponents co-loaded on a DSP are serially executed in response to an IRQA interrupt. The last SimpleComponent on the DSP is followed by a return from interrupt (RTI) instruction, and any remaining cycles are spent in a busywait loop.

A small amount of system code, referred to as the "DSP kernel", is loaded onto each DSP along with the SimpleComponents. The DSP kernel uses less than 3% of the processor bandwidth within the sample loop (to handle the IRQA interrupt). It uses no registers within the sample loop, so all registers are available to the programmer. No processor attention is required to handle data flow to and from the DSPs, thanks to the FSEQ "move engine". Context switching between SimpleComponents is left to the programmer rather than being performed by the DSP kernel. This ensures that register saving and restoring is only performed on the registers that require it.

Since register saving and restoring on the 56000 can require many cycles, a special mechanism is provided for avoiding context switches when possible. The FDAS software supports two types of SimpleComponents: "coarse" and "fine". A coarse SimpleComponent always gets an entire DSP to itself. As a result, the contents of all registers are guaranteed to persist across sample boundaries. This allows the programmer to write a SimpleComponent which uses the entire bandwidth of a DSP without wasting any cycles on saving and restoring registers.

At present, a variety of audio processing features have been implemented on the Frox: Simulation of acoustical spaces, Dolby ProLogic (TM) and Lucasfilm Home THX (TM) surround processing, 8-band stereo graphic equalizer, tone controls, on-screen VU meters, volume, balance, mute, dither, and sampling rate conversion.

The design of the FDAS software permits the development of unlimited new audio features, all of which can be made available to Frox customers by releasing new software.

The design of the FDAS software permits the development of unlimited new audio features, all of which can be made available to Frox customers by releasing new software.