Audio Processing Systems

Udo Zölzer

Dept. of Telecommunications · Technical University of Hamburg
Eissendorfer Str. 40 · D-21073 Hamburg · Germany
http://www.et2.tu-harburg.de/zoelzer/zoelzer.html

Abstract

This tutorial[1] gives an introduction into several hardware aspects for designing audio processing systems based on digital signal processors (DSP). Digital signal processors of different manufacturers and their use in practical circuits will be discussed.

1 Digital Signal Processors

Digital signal processors (DSP) are used for discrete-time signal processing. Their architecture and instruction set is specially designed for real-time processing of signal processing algorithms. Digital signal processors of different manufacturers and their use in practical circuits will be discussed. The restriction to the architecture and practical circuits shall provide the user with the criteria necessary for selecting a DSP for a particular application. From the architectural features of different DSPs, the advantages of a certain processor with respect to fast execution of algorithms (digital filter, adaptive filter, FFT etc.) automatically result. The programming methods and application programs are not dealt with here, because the DSP user guides from different manufacturers provide adequate information in the form of sample programs for a variety of signal processing algorithms.

After comparing DSPs with other microcomputers, the following topics will be discussed in the forthcoming sections: fixed-point DSPs, floating-point DSPs, development tools, single-processor systems (peripherals, control principles) and multi-processor systems (coupling principles, control principles).

The internal design of microcomputers is mainly based on two architectures; the *von Neumann* architecture which uses shared instruction/data bus; and the *Harvard* architecture which has separate buses for instructions and data. Processors based on these architectures are CISCs, RISCs and DSPs. Their characteristics are given in Table 1.

[1]The tutorial is based on the book "Digital Audio Signal Processing", Chapter 4, J. Wiley & Sons, Chichester 1997 [1].

Table 1: CISC, RISC and DSP.

type	characteristics
CISC	Complex Instruction Set Computer - von Neumann architecture - assembler programming - large number of instructions - computer families - compilers - application: universal microcomputers
RISC	Reduced Instruction Set Computer - von Neumann/Harvard architecture - number of instructions < 50 - number of address modes < 4 - hard wired instruction (no microprogramming) - processing most of the instructions in one cycle - optimizing compilers - application: workstations
DSP	Digital Signal Processor - Harvard architecture - several internal data buses - assembler programming - parallel processing of several instructions in one cycle - optimizing compilers - real-time operating systems - application: real-time signal processing

Besides the internal properties listed in the table, DSPs have special on-chip peripherals which are suited to signal processing applications. The fast response to external interrupts enables their use in real-time operating systems.

1.1 Fixed-point DSPs

The discrete-time and discrete-amplitude output of an AD converter is usually represented in 2s complement format. The processing of these number sequences is carried out with fixed-point or floating-point arithmetic. The output of a processed signal is again in 2s complement format and is fed to a DA converter. The signed fractional representation (2s complement) is the common way for algorithms in fixed-point number representation. For address generation and modulo operations unsigned integers are used. Figure 1 shows a schematic diagram of a typ-

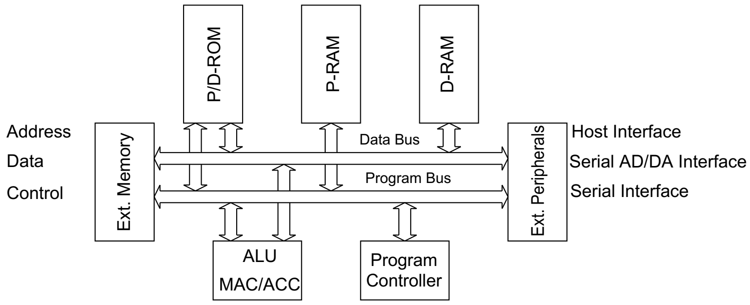

Figure 1: Schematic diagram of a fixed-point DSP.

ical fixed-point DSP. The main building blocks are program controller, arithmetic logic unit (ALU) with a multiplier-accumulator (MAC), program and data memory and interfaces to external memory and peripherals. All blocks are connected with each other by an internal bus system. The internal bus system has separate instruction and data buses. The data bus itself can consist of more than one parallel bus enabling it, for instance, to transmit both operands of a multiplication instruction to the MAC in parallel. The internal memory consists of instruction and data RAM and additional ROM memory. This internal memory permits a fast execution of internal instructions and data transfer. For increasing memory space, address/control and data buses are connected to external memories like EPROM, ROM and RAM. The connection of the external bus system to the internal bus architecture has great influence on efficient execution of external instructions as well as on processing external data. In order to connect serially operating AD/DA converters, special serial interfaces with high transmission rates are offered by several DSPs. Moreover, some processors support direct connection to an RS232 interface. The control from a microprocessor can be achieved via a host interface with a word-length of 8 bits.

An overview of fixed-point DSPs with respect to word-length and cycle time is shown in Table 2. Basically, the precision of the arithmetic can be doubled if quantization affects the stability and numeric precision of the applied algorithm. The cycle time in connection with processing time (in processor cycles) of a combined multiplication and accumulation command gives insight into the computing power of a particular processor type. The cycle time directly results from the maximum clock frequency. The instruction processing time depends mainly on the internal instruction and data structure as well as on the external memory connections of the processor. Table 3 contains the internal memory partitioning of several DSPs. A large on-chip memory for data and instructions is a precondition for efficient programming of algorithms. Data and instruction transfer from external memories can hence be avoided. The availability of special tables (cosine, sine) in ROM supports algorithms like FFT.

Table 2: Fixed-point DSPs.

type	word-length	cycle time
ADSP-2100	16	60/77/80/100 ns
Motorola DSP56156	16	33/50 ns
Motorola DSP5600x	24	60/74 ns
TI 320C1x	16	114/120/200/280 ns
TI 320C2x/5x	16	C2x 78/98/125 ns C5x 35/50 ns

Table 3: Internal memory structure (P = program, D = data).

type	on-chip D-RAM	on-chip P-RAM	on-chip ROM
ADSP-2100	-	16	-
ADSP-210x	1k	2k	-
DSP56156	2k	2k	64 (P)
DSP5600x	2x256	512	2x256 (D)
TI 320C1x	256	-	4k (P)
TI 320C25	288	256 (P/D)	4k (P/D)
TI 320C26	-	1.5k (D/P)	256 (P)
TI 320C50/51	1k	9k/1k (D/P)	2k (P)/8k (P)

The fast access to external instruction and data memories is of special significance in complex algorithms and in processing huge data loads. Further attention has to be paid to the linking of serial data connections with AD/DA converters and the control by a host computer over a special host interface (Table 4). Complex interface circuits could therefore be avoided. For stand-alone solutions, program loading from a simple external EPROM can also be done.

Table 4: External peripherals (xS = x serial interface, xP = x parallel interface).

type	external memory	on-chip peripherals
ADSP-2100	32k (P), 16k (D)	-
ADSP-2101/2	16k (P), 16k (D)	2S
ADSP-2111	16k (P), 15k (D)	2S, 1P
DSP56156	64k (P), 64k (D)	2S, 1P
DSP5600x	64k (P), 128k (D)	2S, 1P
TI 320C1x	4k	
TI 320C25/26	64k (P), 64k (D)	1S
TI 320C50/51	64k (P), 64k (D)	2S

For signal processing algorithms, the following software commands are necessary:

1. MAC (multiply and accumulate)
   → combined multiplication and addition command
2. simultaneous transfer of both operands for multiplication to the MAC (parallel move)
3. bit-reversed addressing (for FFT)
4. modulo addressing (for windowing and filtering)

Different signal processors have different processing times for FFT implementations. The latest signal processors with improved architecture have shorter processing times. The instruction cycles for the combined multiplication and accumulation command (application: windowing, filtering) are approximately equal for different processors, but processing cycles for external operands have to be considered.

1.2 Floating-point DSPs

Figure 2 shows the block diagram of a typical floating-point DSP. The main characteristics of the different architectures are the dual-port principle (Motorola, Texas Instruments) and the external *Harvard* architecture (Analog Devices).

Floating-point DSPs internally have multiple bus systems in order to accelerate data transfer to the processing unit. An On-chip DMA controller and cache-memory support higher data transfer rates. An overview of floating-point DSPs is shown in Table 5. Besides the standardized floating-point representation IEEE-754, there are also manufacturer-dependent number representations. The internal memory structure is given in Table 6.

Table 5: Floating-point DSPs.

type	word-length	cycle time
ADSP-21060	32 (IEEE-754)	25 ns
AT&T DSP32C	32	80/100 ns
Motorola DSP96002	32 (IEEE-754)	50/60/74 ns
TI 320C3x	32	50/60/74 ns
TI 320C40	32	40/50 ns

An overview of external address space and on-chip peripherals is given in Table 7. The external dual-port architecture of some floating-point DSPs supports the design of multiprocessor systems.

1.3 Development Tools

The rapid development of hard- and software for a certain application is supported by development tools which are listed below:

Table 6: Internal memory structure (C = cache memory).

type	on-chip D/P-RAM	on-chip ROM
ADSP-21060	4 Mbit (P/D)	
AT&T DSP32C	2x512 (P/D)	4k (P/D) or 512 P/D-RAM
M DSP96002	2x512 (D), 1k (P)	64 Boot, 2x512 (D)
TI 320C3x	2x1k (P/D) 64 (C)	4k (P/D)
TI 320C40	2x1k (P/D) 128 (C)	4k (P/D)

Table 7: External peripherals (xS = x serial interface, xP = x parallel interface).

type	ext. buses	on-chip periph.
ADSP-21060	4Gx32/48 (P/D)	2S/6P
AT&T DSP32C	4Mx32 (P/D)	1S, 1P
DSP96002	2x 32 bit (A), 2x 32 bit (D) dual-port architecture	-
TI 320C30	16Mx32 (P/D), 8kx32 (P/D) dual-port architecture	2S
TI 320C40	2x 32 bit (A), 2x 32 bit (D) dual-port architecture	6P

- Manufacturers' Literature (Data Books): Data books of manufacturers, application examples and detailed program libraries.
- Assembler/Compiler/Linker: Tools for developing software.
- High-level Language Compilers: The use of higher language compilers allows a fast software development without special knowledge of the architecture and the instruction set of the processor. The generated assembler code can be optimized with respect to processing speed. The advantage of using higher language compilers is the compatibility of the code for different signal processors and the associated fast access to algorithms for future DSPs.
- Real-time Operating Systems: Special operating systems for DSPs with a core consisting of memory management and hardware-interrupt handling, a programming interface for linking application programs and a real-time multitasking core.
- Software Simulator: A software simulator simulates the modules of a DSP and the execution of programs. All registers, memories and interfaces are accessible. Therefore, the programs can be tested under the boundary conditions of the DSP.
- Hardware Emulator (In-circuit Emulation): In-circuit emulation serves for testing the DSP in

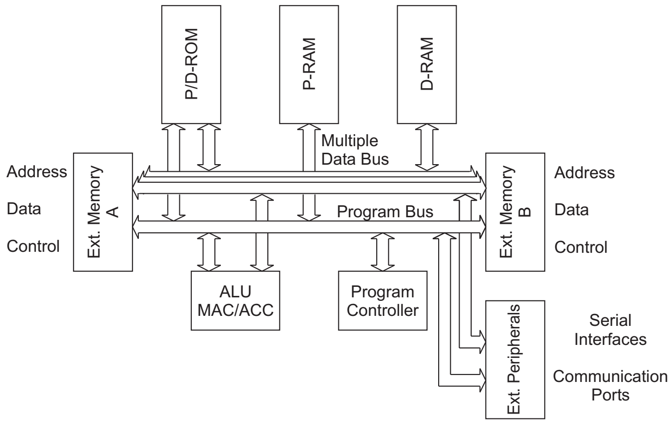

Figure 2: Block diagram of a floating-point digital signal processor.

the target hardware. With the help of a special target cable, the hardware emulator is connected into the socket for the DSP.

- On-chip Emulation: The advantage of a DSP with integrated on-chip emulation is the possibility of self-testing the hardware and software in the target application with the DSP.
- EPROM Simulator: Besides the on-chip emulation, the use of EPROM simulators (or program loading by a control processor) supports the hardware and software development in the target hardware.

2 Digital Audio Interfaces

For transferring digital audio signals, two transmission standards have been established by the AES (Audio Engineering Society) and the EBU (European Broadcasting Union) respectively. These standards are for two-channel transmission [AES92] and for multichannel transmission of up to 56 audio signals [AES91].

2.1 Two-channel AES/EBU Interface

For the two-channel AES/EBU interface, professional and consumer modes are defined. The outer frame is identical for both modes and is shown in Fig. 3. For a

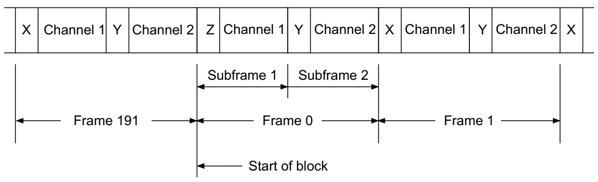

Figure 3: Two-channel format.

sampling period a frame is defined so that it consists of two subframes, for channel 1 with preamble X, and for channel 2 with preamble Y. A total of 192 frames form a block, the block start is characterized by a special preamble Z. The bit allocation of a subframe consists of 32 bits as in Fig. 4. The preamble consists of 4 bits (bit 0...3) and the audio data of up to 24 bits (bit 4...27). The last four bits of the subframe characterize Validity (validity of data word or error), User Status (usable bit), Channel Status (from 192 bits/block=24 bytes coded status information for the channel) and Parity (even parity).

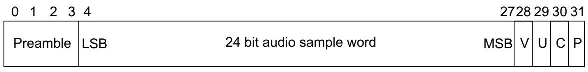

Figure 4: Two-channel format (subframe).

The transmission of the serial data bits is carried out with a biphase code. This is done with the help of an XOR relationship between clock (of double bit rate) and the serial data bits (Fig. 5). At the receiver, clock retrieval is achieved by detecting the preamble (X=11100010, Y=11100100, Z=11101000) as it violates the coding rule.

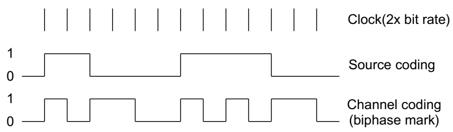

Figure 5: Channel coding.

For consumer applications, two-wired leads with RCA connectors are used. The inputs and outputs are asymmetrical. Also, optical connectors exist. For professional use, shielded two-wired leads with XLR connectors and symmetrical inputs and outputs (professional format) are used.

2.2 MADI Interface

For connecting an audio processing system at different locations, a MADI interface (Multichannel Audio Digital Interface) is used. A system link by MADI is presented in Fig. 6. Analog/digital I/O sys-

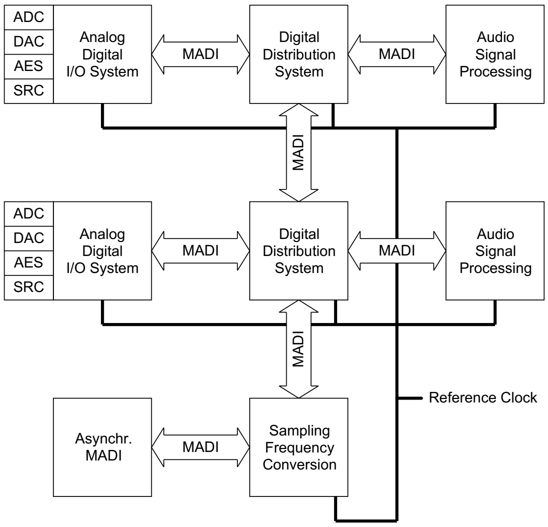

Figure 6: A system link by MADI.

tems consisting of AD/DA converters, AES/EBU interfaces (AES) and sampling rate converters (SRC) are connected to digital distribution systems with bidirectional MADI links. The actual audio signal processing is performed in special DSP systems which are connected to the digital distribution systems by MADI links. The MADI format is derived from the two-channel AES/EBU format and allows the transmission of 56 digital mono channels (see Fig. 7) within a sampling period. The MADI frame consists of 56 AES/EBU subframes. Each channel has a preamble containing the information shown in Fig. 7. The bit 0 is responsible for identifying the first MADI channel (MADI Channel 0). The maximum data rate

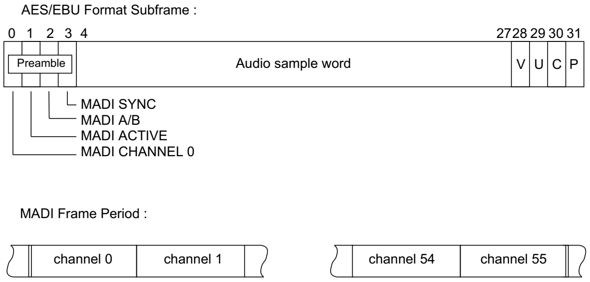

Figure 7: MADI frame format.

of 96.768 Mbit/s is required at sampling rate of 48 kHz+12.5%. Data transmission is done by FDDI techniques (Fiber Distributed Digital Interface). The transmision rate of 125 Mbit/s is implemented with special TAXI chips. The transmission for a coaxial cable is already specified.

3 Single-processor Systems

3.1 Peripherals

A common system configuration is shown in Fig. 8. It consists of a DSP, clock generation, instruction and data memory and a BOOT-EPROM. After RESET, the program is loaded into the internal RAM of the signal processor. The loading is done byte by byte so that only an EPROM with 8 bit data word-length is necessary. In terms of circuit complexity the con-

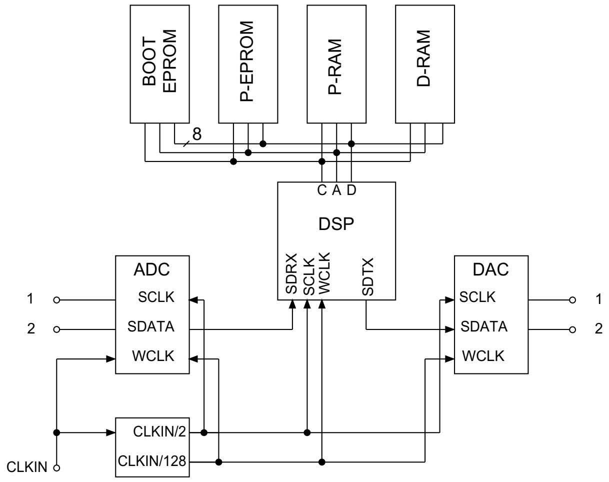

Figure 8: DSP system with two-channel AD/DA converters (C = control, A = address, D = data, SDATA = serial data, SCLK = bit clock, WCLK = word clock, SDRX = serial input, SDTX = serial output).

nection of AD/DA converters over serial interfaces is the simplest solution. Most fixed-point signal processors support serial connection where a lead for bit clock SCLK, sampling clock/word clock WCLK, and the serial input and output data SDRX/SDTX are used. The clock signals are obtained from a higher reference clock CLKIN (see Fig. 9). For non-serially operating AD/DA converters, parallel interfaces can also be connected to the DSP.

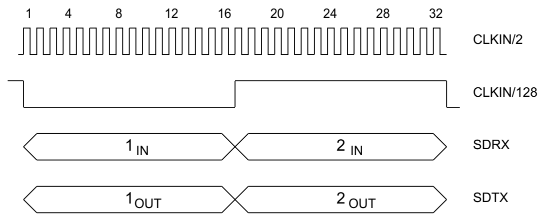

Figure 9: Serial transmission format.

3.2 Control

For controlling digital signal processors and data exchange with host processors, some DSPs provide a special host interface that can be read and written directly (see Fig. 10). The data word-length depends on the processor. The host interface is included in the external address space of the host or is connected to a local bus system, for instance a PC bus.

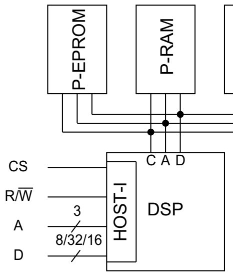

Figure 10: Control via a host interface of the DSP (CS = chip select, R/$\overline{\mathrm{W}}$ = read/write, A = address, D = data).

A DSP as a coprocessor for special signal processing problems can be used by connecting it with a dual-port RAM and additional interrupt logic to a host processor. This enables data transmission between the DSP system and host processor(see Fig. 11). This results in a complete separation from the host processor. The communication can either be interrupt-controlled or carried out by polling a memory address in a dual-port RAM.

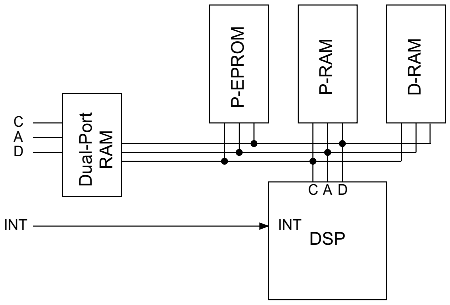

Figure 11: Control over a dual-port RAM and interrupt.

A very simple control can be done directly via an RS232-interface. This is can be carried out via an additional asynchronous serial interface (Serial Communication Interface) of the DSP (see Fig. 12).

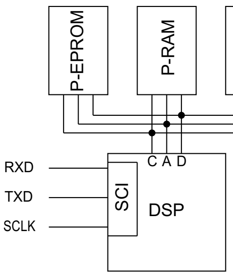

Figure 12: Control over a serial interface (RS232, RS422).

4 Multiprocessor Systems

The design of multiprocessor systems can be carried out by linking signal processors by serial or parallel interfaces. Besides purely multiprocessor DSP systems, an additional connection to standard bus systems can be made as well.

4.1 Connection via Serial Links

In connecting via serial links, signal processors are cascaded so that different program segments are distributed over different processors (see Fig. 13). The serial output data is fed into the serial input of the following signal processor. A synchronous bit clock and a common synchronization SYNC control the serial interface. With the help of a serial time-multiplex

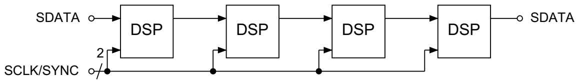

Figure 13: Cascading and pipelining (SDATA = serial data, SCLK = bit clock, SYNC = synchronization).

mode (Fig. 14) a parallel configuration can be designed which, for instance, feeds several parallel signal processors with serial input data. The serial outputs of signal processors provide output data in time-multiplex. A complete time-multiplex connection via

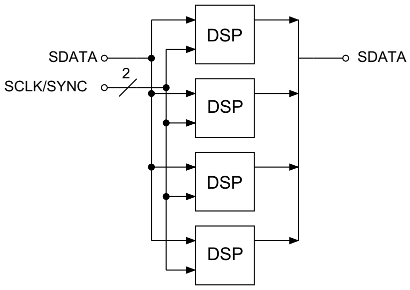

Figure 14: Parallel configuration with output time-multiplex.

the serial interface of the signal processor is shown in Fig. 15. The allocation of a signal processor at a particular time slot can either be fixed or carried out by an address control ADR.

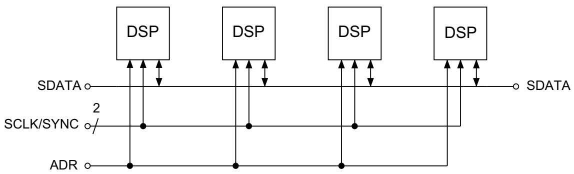

Figure 15: Time-multiplex connection (ADR = address at a particular time).

4.2 Connection via Parallel Links

The connection via parallel links is possible with dual-port processors as well as with dual-port RAMs (see Fig. 16). A parallel configuration of signal pro-

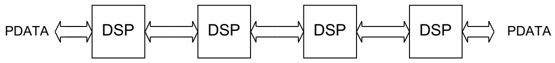

Figure 16: Cascading and pipelining.

cessor systems with a local bus is shown in Fig. 17. The connection to the local bus is done either over a dual-port RAM or directly with a second signal processor port. Another possible configuration is the

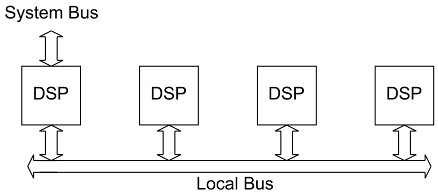

Figure 17: Parallel configuration.

use of a 4-port RAM as shown in Fig. 18. Here, one processor serves as a connector to a system bus and feeds three other processors over a 4-port RAM with control and data information.

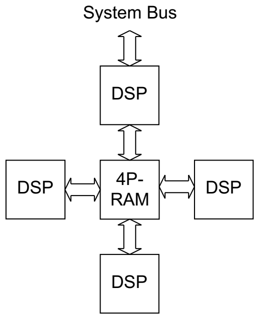

Figure 18: Connection over a 4-port RAM.

4.3 Connection via Standard Bus Systems

The use of standard bus systems (VME bus, MULTIBUS, PC bus) to control multiprocessor systems is presented in Fig. 19. The connection of signal processors can either be carried out directly over a control bus or with the help of a special data bus. This parallel data bus can operate in time-multiplex. Hence control information and data are separated. A few of the criteria for standard bus systems are data transfer rate, interrupt request and processing, the option of several masters, auxiliary functions (power supply, bus error, battery buffer) and mechanical requirements.

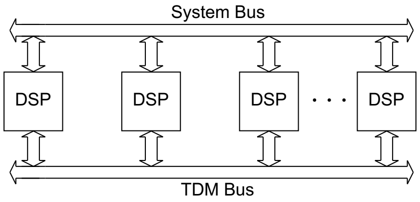

Figure 19: Signal processor systems based on standard bus system.

4.4 Scalable Audio System

The functional segmentation of an audio system into different stages, the analog, interface, digital and man-machine stages, is shown in Fig. 20. All stages

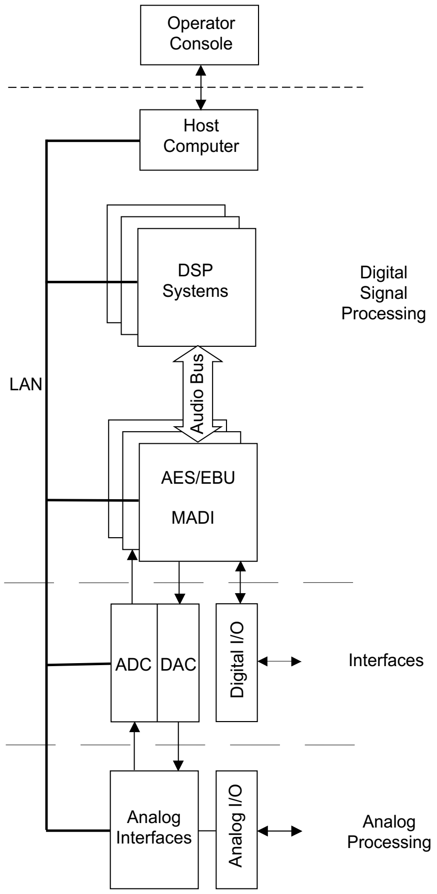

Figure 20: Audio system.

are controlled by a LAN (Local Area Network). In the analog domain, crosspoint switches and microphone amplifiers are controlled. In the interface domain AD/DA converters and sampling rate converters are used. The connection to a signal processing system is done by AES/EBU and MADI interfaces. A host computer with a control console for the sound engineer serves as the central control unit. The realization of the digital domain with the help of a standard bus system is shown in Fig. 21. A central mixing console controls several subsystems over a host. These subsystems have special control comput-

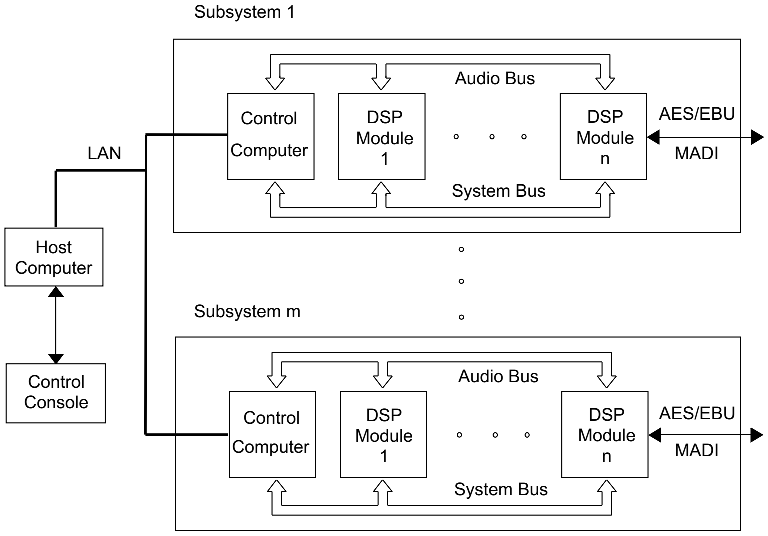

Figure 21: Scalable digital audio system.

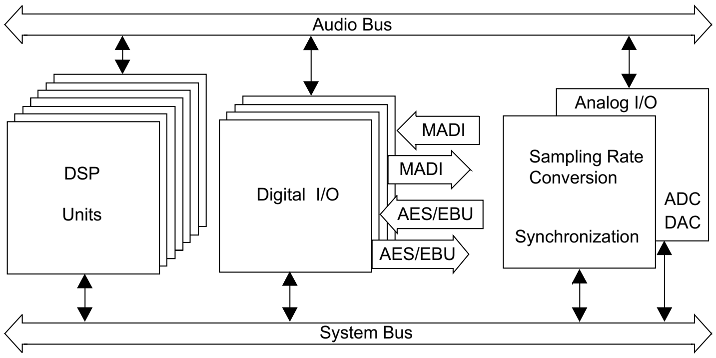

Figure 22: Subsystem.

ers which control several DSP modules. The system concept is scalable (extendable in modules) within a subsystem and by extension to several subsystems. Audio data transfer between subsystems is performed by AES/EBU and MADI interfaces. The segmentation within a subsystem is shown in Fig. 22. Here, besides DSP modules, digital interfaces (AES/EBU, MADI, sampling rate converters, etc.) and AD/DA converters can be integrated.

5 Conclusion

In this tutorial several methods for designing single-processor and multiprocessor DSP systems are discussed. Most systems make use of the specialized external peripherals of DSPs. These are the host interface and the serial interfaces for single-processor systems. Multiprocessor systems can be based on serial or parallel communication links. The special choice is dependent on the number of DSPs and the special application. Most of the coefficient calculations and the control algorithms can be performed by the DSP or in combination with simple 8-bit microprocessors providing interfaces to knobs and displays. High-level programming languages will speed up the development time for DSPs and multiprocessor systems in the future.

References

[1] U. Zölzer, *Digital Audio Signal Processing*, J. Wiley & Sons, Chichester, 1997.

[2] AES10-1991 (ANSI S4.43-1991): AES Recommended Practice for Digital Audio Engineering - Serial Multichannel Audio Digital Interface (MADI).

[3] AES3-1992 (ANSI S4.40-1992): AES Recommended Practice for Digital Audio Engineering - Serial Transmission Format for Two-Channel Linearly Represented Digital Audio.