
Rupert C. Nieberle
Stalitzerstr. 62, D-1000 Berlin 36
Technical University of Berlin, Computer Science FR 2-2,
Franklinstr. 28, D-1000 Berlin 10
nieberle@opal.cs.tu-berlin.de

ABSTRACT: This paper describes the DSP-Subsystem and its Programming-Tools of the open network CAMP, which allows composition and interactive performance on different Hosts, the usage of high speed communication interface for distributed Sound Synthesis, natural Sound Sampling and Effect Processing, which is based on the Motorola DSP56001.

Introduction

The CAMP-DSP-Subsystem called IDEAL56 for Synthesis, Analysis and Processing of Music is the Signal-Processing-Engine of the Open Music Workstation developed for CAMP (Computer Aided Music Processing). The network like architecture of the CAMP-System allows control of Multi-Devices and high speed sample transfer between Hosts and specialized DSP-Subsystems through SCSI-Interfaces. Connected to Hosts, depending on applications downloaded to the DSPs on the Subsystem, Sample-Devices for natural Sound Sampling and Generation, or algorithms for Music- and Sound Processing are performed. For programming the DSP, a comfortable symbolic debugger can be used to test low level DSP algorithms and high level language constructs are provided to use low level DSP macros, which are embedded in an Object-Oriented and Lisp based Composition Language[1]. The user may use lots of predefined functions or easily extend this environment with own ideas.

Some of the interactive applications are: Recording and Playing Sounds, Filtering, Spectral-Analysis and Resynthesis or Space-Processing. Highlevel network functions in CAMP are implemented, that even novice programmers are able to communicate between various stations easily, which represent the performing musicians. In this environment musicians produce their compositions in realtime. All participants in the network orchestra may use the same distributed musical sources and control interactively all music generating devices throughout the network[2].

After a short introduction of IDEAL56 DSP-Subsystem's hard- and software the high speed communication interface and communication language between the Host and IDEAL56 will be discussed.

IDEAL56 the DSP-Subsystem of CAMP

The modular hard- and software tools of IDEAL56 for programming the DSP56001 allows...
to develop on the very same hardware from test, including I/O, to final applications. This, of course, includes the development of algorithms and testing them in realtime, which is very convenient supported on IDEAL56’s Debugger. Short training and online help, a numerous number of additional utilities help the programmer to get started. Convenience and easy Debugging are the spotlights of IDEAL56.

Hardware Specification of IDEAL56

The basic idea of this development system is easy going from prototype to target system. In the CAMP environment, the DSP-Subsystem consists of one or more DSP-boards, which can be developed using a PC/AT. Typical algorithms for musical applications like recording and playback of sampled Sounds are developed on the DSPC56 board, which can be used as a plugin PC-board, supported by a realtime symbolic debugger. The DSPC56 board and its interfaces to the host, on which the application will be developed is shown in figure 1. There are numerous expansion boards offered for IDEAL56. It connects between the DSPC56 and additional peripheral boards through a 24-bit parallel Bus. The expansion boards, A/D and D/A, AES/EBU, RAM Expansion or SCSI, come in ‘EUROPE’ format, which means, that they build together with DSPC56 a long PC-board.

On the developer board DSPC56 the Processor is running at a clock frequency of 20 or 27 or 33MHz. Three memories P for program, X and Y for data are provided with 32k words (24-bit long) and can be easily expanded to full address space of 64k words.

Reset and programmable interrupt take care of defined program flow after reset. The booting can be done through hostport or - for standalone applications- from an inserted EPROM.

Three cascadeable timers may be used as watchdog or sample rate generators. Eight pre-coded select lines are for customer’s expansion boards, like AD, DA-Converters. Other pre-coded select lines allow memory expansion or user build external multiprocessor memory configurations. All additional expansion boards are connected to the main processor board through a 96-pin-female connector at the inner end of the plug-in-PC-Board. An additional 15pin DSUB-female connector for I/O allows NeXT-DSP-Port devices to be plugged in, an easy to use of the shelf product, like, for example the ‘Digital Microphone’ or ‘ProPort’ from Ariel. Also Midi-IN and OUT can be provided for direct control of generated instrument sounds.

Software Support for IDEAL56

A vast library of functions for digital signal processing, almost every standard problem in the digital domain, can be provided. In this library the functions are written in macros. Standardized calls with handing over of parameters and standardized block data management are offered for convenient programming like in high level languages [3]. Furthermore the library supports mathmatic functions like for example :

\[ \text{TMC} \quad 530 \]
-FFT, IFFT, DCT, IDCT, CCF, Interpolation and Decimation, Fast Convolution
-Windows: Blackman, Triangle, Hanning, Hamming, Kaiser with adjustable alpha for Bessel functions, Tapered with adjustable overlap area, user defined windows
-FIR Filters, IIR Filters: numerous standard configurations of infinite impulse response filter structures
As one of the most flexible macro the FFT will be introduced to you[4]. Only by changing parameters in the calling function it will be adjusted to:
- Optimal to time or space in memory,
- Inplace/Noninplace to be processed.
- result in bitreverse or linear order.
- complex transformation with imaginary part zero or
- 'real FFT' with 2 real no imaginary parts with resorting the data.
- in cases of N, were the number of points is not a power of 2 and not a prime, a Winograd transformation will be provided.

Automatic Code Generation from Signal Flowcharts
The most advanced feature is the Code Generator of IDEAL56. It is part of a graphical surface in order to operate the application on the IDEAL56 system. The Code Generator enables the musician to write realtime compositions without writing a single line of assembler. The Code Generator typically produces programs which are special and deal in general with problems of Digital Signal Processing. Of course a word processing program cannot be coded with this code generating system.

Small Operating-System 'SOS'
The software that runs on the DSP boards, a Small Operating-System 'SOS', was developed to support communication [5,6], code generation and high level commands from hosts. For control and programming of actions executed on the DSP, it behaves similar to a forth interpreter and supports all level of language extensions (Object-Oriented and Lisp), which have been developed in CAMP [2]. The communication language, called F-Code, is easy to use and goes far beyond the programming facilities of forth. For example it is not restricted to certain series of functions because of data exchange on stack, every functions can be allocated to own memory for parallel task execution. Indirect jumps to executable code through tables allow simple code refreshing in case of dynamic loading of actions. There are 'groupfunctions' like Objects of the same type, which incorporate functions, that can be applied to other groupfunctions. SOS is build of smallest code pieces low level words (DSP macros), which exchange data on a DSP-Parameter stack and so called high words, which are build of a list of lowlevel words. The DSP-F-Code looks, besides the different internal data format, exactly like the forth code on the host system. This allows integration of all Forth Music Language [7]constructs.

ICMC 531
The IDEAL56 Basis- and Extension boards used by CAMP

References


