

A DIGITAL PROCESSING SYSTEM FOR SOURCE LOCATION AND SOUND CAPTURE BY LARGE MICROPHONE ARRAYS *

Harvey F. Silverman[†]

William R. Patterson III[†]

James L. Flanagan[†]

Daniel Rabinkin[†]

[†]LEMS, Division of Engineering, Box D, Brown University, Providence, RI 02912

[‡]CAIP Center, Rutgers University, CoRE Building, Piscataway, NJ 08855-1390

ABSTRACT

The Huge Microphone Array(HMA) project started in February 1994 to design, construct, and test a real-time 512-microphone array system and to develop algorithms for use on it. Analysis of known algorithms showed that signal-processing performance of over 6 Gigafllops would be required; at the same time, there was a need for "portability", i.e., fitting into a small van. These tradeoffs and many others have led to a unique design in both hardware and software. This paper presents the design and its justifications. Performance data for a few important algorithms relative to usage of processing-capability, response latency, and difficulty of programming are discussed.

1. INTRODUCTION

The scope of the Huge Microphone Array (HMA) project goes substantially beyond the capabilities of earlier systems. No all-digital, real-time, intelligent, 512-microphone-array system has ever been designed, and the number of microphones is even larger than the 400 used in the analog system built at AT&T Bell Laboratories in the mid 1980's [1]. The system is nearing completion and will be initially used in three different sites. The reader is referred to [2] for a more detailed description, [3] for a recent general reference, [4] for the application of matched filtering, and to [5, 6] for location determination.

2. CRITERIA FOR DESIGN

A typical hierarchy for processing the signals from a microphone array is shown in Figure 2. Many of the algorithms to be implemented operate on frequency-domain data. Thus, the first processing after A/D conversion is the accumulation of short time segments, conversion to the frequency domain, and coding for transmission to other processors. The need to minimize cabling and noise problems, led to an early design decision to place the ADCs for groups of sixteen microphones in small boxes, called *microphone modules*, mounted near the array itself, remote from the console. Each box connects by a dual optical-fiber cable to the central console. As a result, only 32 duplex-fiber cables connect to the console, rather than 512 long, and potentially noisy, microphone cables (see Figure 1).

The sampling rate, frame length and the frame advance

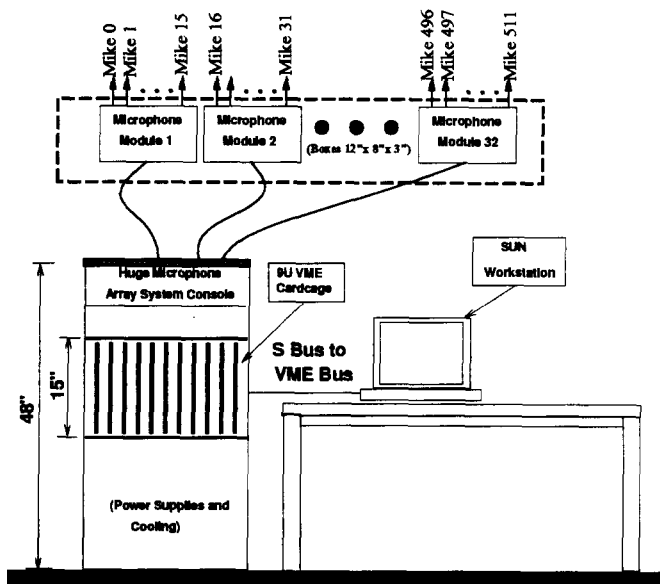


Figure 1. The Packaging of the HMA

must be selected to make this remote microphone-module design feasible. Sampling at 20kHz, a frame length of 51.2ms (1024 samples), and a frame advance of 25.6ms(512 samples) were chosen empirically as suitable for the algorithms and the architecture.

Figure 2 suggests that data from one microphone will usually be used by more than one processor. To meet this need and to minimize acquisition latency, the data transmission system simultaneously writes the data into local buffer memories on each processing board in the console. Any processors on the board can share one such frame buffer. Data transfers in and out of this memory are fairly fast (17.92MW/s). Thus all microphone data are available to all processors, and data selection time is minimal.

Packetization in the microphone modules introduces the first delay(*latency*) between the input acoustic signal and any output acoustic signal. Summing the nominal delays given in Figure 2 indicates that over 125ms of latency is evident. This calculation also assumes that the architecture does not impose further delay because of the use of multiple processors and the need to pass data between processors within the single frames of delay shown. It is unfortunate that the time needed to produce an improved audio signal is

*RESEARCH SUPPORTED BY NSF GRANT MIP-9314625

so significant. While an expected latency of from 100-500ms is likely to be acceptable for teleconferencing, or even for input to a recognizer, this delay would not be acceptable in direct applications such as audience pickup or direct public address. In real systems there is an additional source of latency should it be necessary to split the computation of any of the blocks of Figure 2 among several processors.

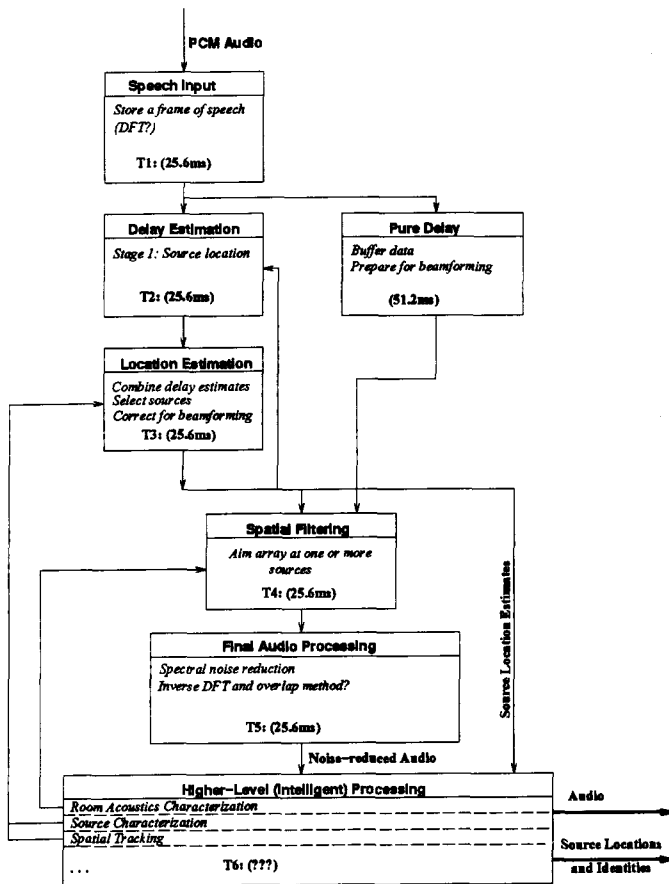


Figure 2. A Typical Data-Flow Structure in a Microphone-Array System (with Nominal Delay Times)

Frame-by-frame processing has another implication. In all the above it has been assumed that there is no delay due to **communications**. Each frame (here 25.6ms) must include time for the processor to receive its input and to return its output, during which the DSP microprocessors cannot do calculations. For computational efficiency, it must be possible to transfer one or more full frames of transformed audio signal in a time that is short compared to the frame interval. The HMA design uses high-speed serial links that can connect multiple pairs of boards simultaneously. The link rate is 2.38MW/s of 32-bit words. Transfers between processors on a single board are at 8MW/s.

Writing **software** for multiprocessor real-time systems is very difficult so a great deal of systems software is needed.. The decision was made to make the hardware "smarter" and develop a *load and go* form of an operating system. The main hardware feature that affects the operating system is

that data flow is not controlled by the DSP processors, but, instead, a set of supervisory processors sets up and starts DMA transfers independently of the DSP programs.

The HMA(Figure 3) meets the following design criteria:

1. At least 6Gflops/s of DSP computing performance is needed to do matched filtering in real-time for 512 microphones implying about 100 ADSP21020 processors.
2. Large, fast local memories are needed; at least 128K words each of data and program memory per processor.
3. Control channels are required; a low speed uplink to command the modules and a fast path to and from the workstation for booting the system, recording data, and controlling the system at a high level.
4. To minimize latency, there must be few limitations on the scheduling of data transfers.

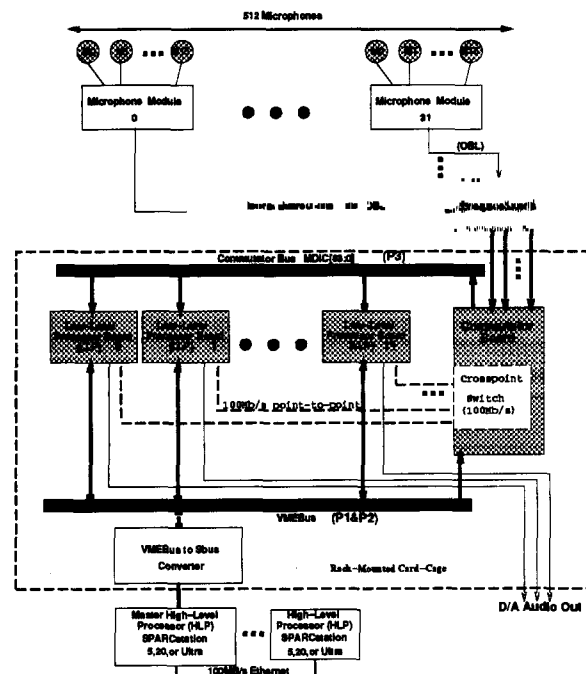


Figure 3. HMA System Block Diagram

3. HMA HARDWARE

The **Microphone Module** for the HMA is shown in Figure 4. Each module does, 1) the preamplification and biasing for 16 electret microphones, 2) analog-to-digital conversion of the microphone signals, 3) DFT transformation and compression or coding of the data, and 4) packetization and transmission of the data to the console unit. One should note that the module circuit board may also be populated for use as a personal computer attachment. In this mode, the system has the capability of a single ADSP21020, 33MHz signal processor with up to 16 microphones of input. This second mode of operation has proven to be quite valuable inside and outside of the University in a package called the **Brown Megamike**[7].

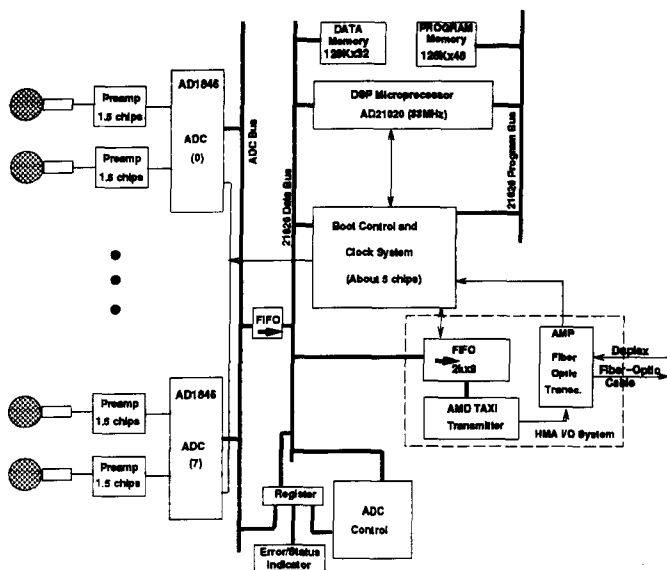


Figure 4. Block Diagram of Module

The **Commutator** system is the traffic manager for the microphone data. Its principle function is to communicate with the microphone modules and to multiplex their data outputs onto the commutator bus. Multiplexing is implemented with two, large Xilinx FPGA's. The commutator bus itself is a VME P1 backplane that has been modified to easily handle the one-way, 64-bit 17.92MHz transfers. The single board fits a standard VME Bus 9U rack (366.7mm high and 277mm deep) and also hosts an SGS-Thompson crosspoint switch that allows transfers between Low-Level Processing(LLP) boards at 2.38MW/s. With 12 LLP boards, this device can support up to 6 pairs communicating at the same time.

The **Low-Level Processing(LLP)** system consists of twelve boards (Figure 5). Each LLP board hosts eight DSP processing systems on daughter boards that mount on the rear. The daughter boards (7.2cmx11.9cm) are ten layers and contain an ADSP-21020 DSP processor, 10Mb of fast SRAM and support logic. The mother/daughter arrangement reduces the complexity of the LLP board and provides processor-system interchangeability.

The dataflow control on the LLP board is the HMA's most idiosyncratic feature. The ADSP-21020 has independent ports for its two memories allowing simultaneous I/O; the design uses a separate DMA controller and a supervisor processor to manage these data transfers. After powerup, the high-level processor (a SPARCstation) loads the supervisor static RAM with routing tables that the processor uses to control the DMA communication system. When the supervisor starts a DMA transfer that affects one of the ADSP21020 processors, the DMA controller requests both buses of the target processor, strobes its program memory in or out, and, simultaneously, loads microphone data into its data memory. The DSP processor programs have no influence on what data they receive or when, which implies that programs may be generic; exact duplicates may be used on the set of processors doing the same task but on different data.

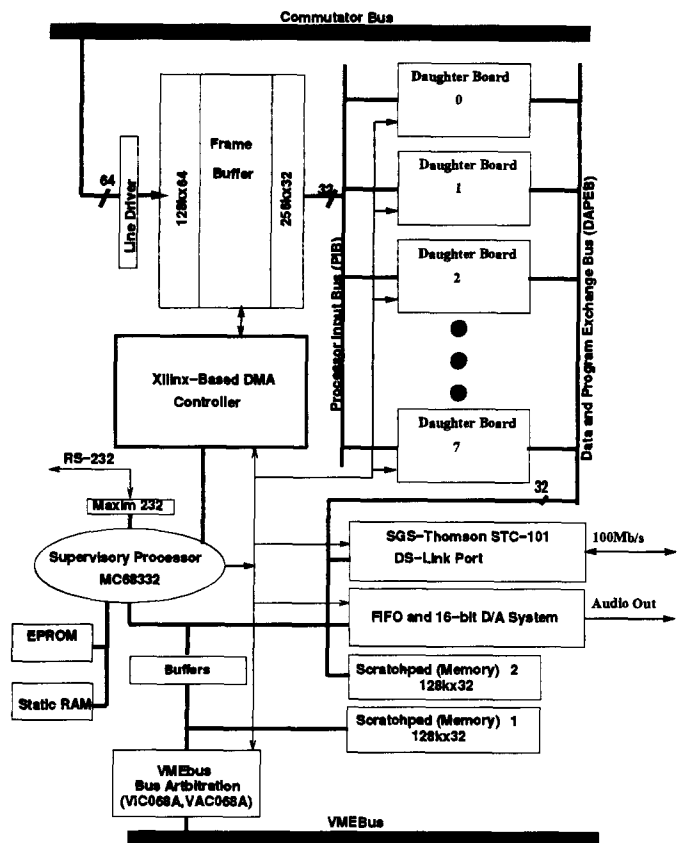


Figure 5. Block Diagram of Low-Level Processing Board

Each LLP board also has two scratchpad memories. Memory 1 is tightly coupled to the VME Bus port and is used as a mailbox for data entering or leaving that port. The VMEbus interface offers full bus master capability. Memory 2 is used for transfers between DSP processors on the same board at a net rate of 8MW/s. The timing of a frame is indicated in Figure 6. All the processors on all the boards run on the data of Frame $n-1$ (or earlier if buffered) while the data for Frame n is being loaded into the frame buffer. Any time an ADSP21020 DSP processor is not the target of a data transfer, it can do calculations. Since the transfer of data from a microphone takes only $28.2\mu s$ of the 25.6ms frame time, computational efficiency is very high.

4. SYSTEM SOFTWARE

The operating system for HMA has two major components. The first is an interactive set of tools (Application Tools for the HMA (ATHMA)) that develops a set of files that describe an application. These files contain 1) a program library (all the binary files for programs to run on LLP processors - one program from the library may be run on many LLP processors), 2) a mapping of programs to processors, 3) routing tables to be loaded into the memory of the supervisory processors to control DMA data transfers, 4) parameters to be passed to the DSPs at run-time, and 5) optionally, a program to run on the workstation after the HMA starts. The details of ATHMA are beyond the scope

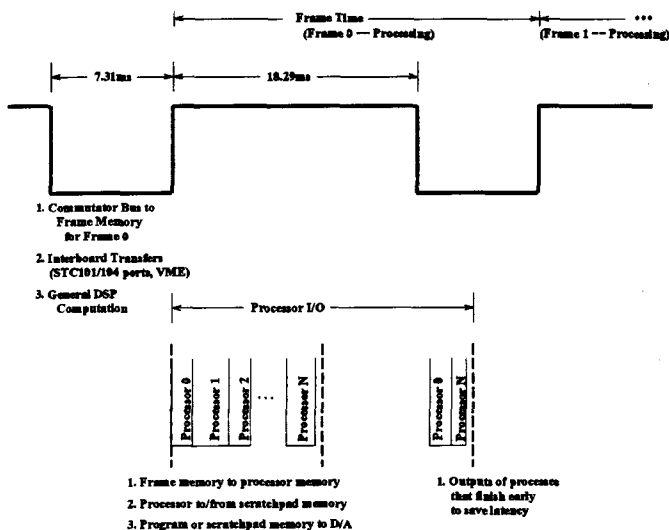


Figure 6. Typical Frame Timing on the LLP Boards

of this paper, but the reader is referred to [2]. The second component is a loader that uses the output files from ATHMA as input, loads the appropriate memories and registers of the HMA, starts the table execution on the supervisors, and, if needed, starts any required workstation programming.

5. SOME NUMBERS FOR A TYPICAL APPLICATION

An important application, locationing and beamforming, has been implemented on the Brown Megamike system for 16 microphones, giving the actual times for the implementation of this application on the 21020 DSP microprocessor [7]. In extrapolating to 512 microphones the following assumptions have been made:

- Brandstein's LI estimator [8, 5], which has a significant computational advantage for microphones grouped as *orthogonal quads* or sets of four microphones arranged as two pairs having the lines between the pairs intersect at right angles.
- The 512 microphones are spread in sub-arrays around the room and not just on a single wall. Experiments suggest that four orthogonal quads per sub-array is sufficient for accurately locating sources.
- The LI estimator, which computes from bearing lines, obtains its best estimates from the intersections of bearing lines from different sub-arrays.
- All microphones contribute to the beamformed signal.

Using the known times for all the algorithms, 36 processors are required (5 LLP boards) for this relatively simple application. Sixteen processors do time-delay estimation, four do the locationing and one computes the final position estimate. Two processors are used to check and correct the delay estimates. Twelve processors are needed for the beamforming in a three-deep tree, and one collects the final output. The latency is kept to under 150ms by the architecture, suitable for teleconferencing and other applications.

6. CONCLUSION

The design of a system that supports the use of an array of 512 microphones in real time has been described. One-hundred twenty-eight fast, floating-point signal processors give real-time performance for moderately complex algorithms. A straightforward example has illustrated the reasons for the decisions. After the hardware and software system is functional, the next step is to develop a set of useful applications. These include software for recording a database from 512 microphones, a practical matched-filtering and/or inverse-filtering beamformer, improved adaptive beamformers, talker-characterized trackers, and systems with advanced methods of noise-cancellation and reverberation reduction.

A project of this magnitude always involves many people. The authors would especially like to acknowledge the contributions of John Adcock, Michael Brandstein and Joseph DiBiase to the system and software design.

REFERENCES

- [1] J. L. Flanagan, J. D. Johnson, R. Zahn, and G. W. Elko. Computer-steered microphone arrays for sound transduction in large rooms. *J. Acoust. Soc. Amer.*, 78(5):1508-1518, November 1985.
- [2] H. F. Silverman, W. R. Patterson III, and J. L. Flanagan. The huge microphone array (HMA). LEMS Technical Report 156, LEMS, Division of Engineering, Brown University, Providence, RI 02912, November 1996.
- [3] J. L. Flanagan and H. F. Silverman. Material for third biennial roundtable of microphone-array technology. LEMS Technical Report 151, LEMS, Division of Engineering, Brown University, Providence, RI 02912, October 1996.
- [4] E. Jan, P. Svaizer, and J. Flanagan. Matched-filter processing of microphone array for spatial volume selectivity. In *Proceedings of ISCAS95*, pages 1460-1463. IEEE, 1995.
- [5] M. S. Brandstein, J. E. Adcock, J. H. DiBiase, and H. F. Silverman. A closed-form method for finding source locations from microphone-array time-delay estimates. In *Proceedings of ICASSP-1995*, pages 3019-3022, Detroit, Michigan, May 1995.
- [6] M. S. Brandstein, J. E. Adcock, and H. F. Silverman. A localization error criterion for microphone-array sensor placement. In *Proceedings of ICASSP-1996*, pages 901-904, Atlanta, GA, May 1996.
- [7] J. E. Kessler. On the implementation of a real-time microphone-array audio source location algorithm with applications to video teleconferencing. Master's thesis, Brown University, Providence, RI, May 1996.
- [8] M. S. Brandstein, J. E. Adcock, and H. F. Silverman. A practical time-delay estimator for localizing speech sources with a microphone array. *Computer, Speech and Language*, 9(2):153-169, April 1995.