View metadata, citation and similar papers at core.ac.uk

brought to you by CORE

provided by Washington University St. Louis: Open Scholarship

Washington University in St. Louis

Washington University Open Scholarship

All Computer Science and Engineering Research

Computer Science and Engineering

Report Number: WUCS-93-40

1993

The Washington University MultiMedia eXplorer

William D. Richard, Jerome R. Cox Jr., A. Maynard Engebretson, Jason Fritts, and Craig Horn

The Washington University MultiMedia eXplorer (MMX) is a complete, host-independent multimedia system capable of transmitting and receiving JPEG-compressed video, CD-quality audio, and high-resolution radiographic images over the Washington University broadband ATM network. If the host is equipped with an ATM interface card, normal network traffic is supported via "T" and "Y" connections. The MMX consists of an ATMizer and three multimedia subsystems. The ATMizer implements the host interface, the interface to the ATM network, and the interface to the three multimdeia channels. This paper describes the architecture of the MMX, the software used with the system, and the applications... **Read complete abstract on page 2.**

Follow this and additional works at: https://openscholarship.wustl.edu/cse_research

 Part of the Computer Engineering Commons, and the Computer Sciences Commons

Recommended Citation

Richard, William D.; Cox, Jerome R. Jr.; Engebretson, A. Maynard; Fritts, Jason; and Horn, Craig, "The Washington University MultiMedia eXplorer" Report Number: WUCS-93-40 (1993). *All Computer Science and Engineering Research.*
https://openscholarship.wustl.edu/cse_research/535

Department of Computer Science & Engineering - Washington University in St. Louis
Campus Box 1045 - St. Louis, MO - 63130 - ph: (314) 935-6160.

This technical report is available at Washington University Open Scholarship: https://openscholarship.wustl.edu/cse_research/535

The Washington University MultiMedia eXplorer

William D. Richard, Jerome R. Cox Jr., A. Maynard Engebretson, Jason Fritts, and Craig Horn

Complete Abstract:

The Washington University MultiMedia eXplorer (MMX) is a complete, host-independent multimedia system capable of transmitting and receiving JPEG-compressed video, CD-quality audio, and high-resolution radiographic images over the Washington University broadband ATM network. If the host is equipped with an ATM interface card, normal network traffic is supported via "T" and "Y" connections. The MMX consists of an ATMizer and three multimedia subsystems. The ATMizer implements the host interface, the interface to the ATM network, and the interface to the three multimdeia channels. This paper describes the architecture of the MMX, the software used with the system, and the applications which have been developed to demonstrate the capability of broadband ATM networks for multimedia applications.

The Washington University MultiMedia eXplorer

William D. Richard, Jerome R. Cox, Jr., A. Maynard Engebretson,
Jason Fritts, Brian L. Gottlieb and Craig Horn

WUCS 93-40

August 17, 1993

Department of Computer Science
Applied Research Laboratory
Campus Box 1045
Washington University
One Brookings Drive
St. Louis, MO 63130-4899

Abstract

The Washington University MultiMedia eXplorer (MMX) is a complete, host-independent multimedia system capable of transmitting and receiving JPEG-compressed video, CD-quality audio, and high resolution radiographic images over the Washington University broadband ATM network. If the host is equipped with an ATM interface card, normal network traffic is supported via "T" and "Y" connections. The MMX consists of an ATMizer and three multimedia subsystems. The ATMizer implements the host interface, the interface to the ATM network, and the interface to the three multimedia subsystems. This paper describes the architecture of the MMX, the software used with the system, and the applications which have been developed to demonstrate the capability and applicability of broadband ATM networks for multimedia applications.

ABSTRACT

The Washington University MultiMedia eXplorer (MMX) is a complete, host-independent multimedia system capable of transmitting and receiving JPEG-compressed video, CD-quality audio, and high-resolution radiographic images over the Washington University broadband ATM network. If the host is equipped with an ATM interface card, normal network traffic is supported via "T" and "Y" connections. The MMX consists of an ATMizer and three multimedia channels. The ATMizer implements the host interface, the interface to the ATM network, and the interface to the three multimedia channels. This paper describes the architecture of the MMX, the software used with the system, and the applications which have been developed to demonstrate the capability and applicability of broadband ATM networks for multimedia applications.

Key words: Broadband ATM, JPEG Compression, Multimedia.

I. INTRODUCTION

A. The Washington University ATM Network

Over the past five years, the Applied Research Laboratory at Washington University has developed a prototype Asynchronous Transfer Mode (ATM) switch, a set of demonstration network peripherals, and a testbed broadband network [1]. The current network is based on four geographically dispersed, 16-port, 100 Mb/s, prototype ATM switches connected by more than 200 miles of single mode fiber.

During the 1994 academic year, the network will be expanded, and the prototype switches will be replaced by 155 Mb/s commercial versions whose designs were derived from our prototype and licensed by Washington University. As a first step in this upgraded network, we have developed a Computer Science Department network with more than 20 multimedia workstations. These workstations did not originally have multimedia capability and had been purchased in the past from several different vendors. We were anxious to demonstrate the full capability of our ATM network and have sought a solution that allows the use of existing computing platforms yet provides high-performance multimedia communication.

B. The Washington University Multimedia Explorer

The Washington University MultiMedia eXplorer (MMX) is a complete multimedia system capable of transmitting and receiving video, audio, and radiographic images over the Washington University broadband ATM network. The MMX is a second-generation multimedia system based on the earlier, host-based Washington University Multimedia System (MMS) [2]. A block diagram of two typical MMX configurations are shown in Fig. 1 and Fig. 2. The MMX is designed to operate with any host. A standard RS-232C interface [3] is used to control the MMX from an application program running on the host. If the host is equipped with an ATM interface card, normal network traffic is supported via "T" and "Y" connections. The "T" connection copies in-bound ATM cells and transmits them to the host ATM card. The "Y" connection places cells from the host ATM card into unassigned cells originating from the MMX to form a single cell stream. An overview block diagram of the MMX system is shown in Fig. 3.

The serial interface, the "T" and "Y" connections, the interface to the ATM network, and the to the multimedia subsystem are implemented by the MMX system shown in block diagram form in Fig. 3. As shown in

Fig. 4., the ATMizer subsystem consists of a 16 MHz Motorola MC68030 CPU [4] with 4 Megabytes of DRAM, 128 Kilobytes of EPROM, a Multi-Function Peripheral Chip (MFP) [5], and interfaces to the ATM network and to the multimedia subsystem. Three dedicated buses, the PBUS, the TxFIFO Bus, and the RxFIFO Bus, are used to connect the multimedia subsystem to the ATMizer. A fourth dedicated bus, the I^2C Bus [6], is used to control the video components in the multimedia subsystem. The audio components in the multimedia subsystem are controlled via the PBUS. Multimedia data streams pass between the multimedia subsystem and the ATM network, via the ATMizer, over the TxFIFO and RxFIFO buses. Local CPU bus bandwidth is not used for multimedia data transfer.

There are currently three multimedia channels used with the ATMizer. The video channel, shown in Fig. 5, digitizes an input color NTSC video signal [7], compresses the digital video data stream using a hardware Joint Photographic Experts Group (JPEG) [8] compression engine, stuffs the appropriate framing tags into the data stream, and routes the resulting data stream to the ATMizer via the TxFIFO Bus for transmission. The video channel also accepts a compressed video data stream from the ATMizer via the RxFIFO Bus, strips the framing tags, decompresses the data stream, and produces an output NTSC video signal. A frame buffer is used to eliminate the synchronization problems caused by separate transmitter and receiver clocks [9]. Two hardware JPEG engines are used by the video channel so that video can be compressed for transmission and decompressed for display simultaneously.

The second multimedia channel, shown in Fig. 6, digitizes an input analog stereo audio signal, i.e., both left and right stereo channels, and produces a CD-quality digital audio data stream for transmission by the ATMizer. Both microphone and line-level inputs are available. This channel also accepts a digital audio stream from the ATMizer and generates an output analog stereo audio signal. A monaural input and a monaural output are also supported by the system.

The third multimedia channel, shown in Fig. 6, accepts high-resolution radiographic images from the ATMizer via the RxFIFO Bus and reformats them for transmission to a high-resolution monochrome display [10]. The radiographic images used with this channel are served to the ATM network via a dedicated "mini-ATMizer" transmitter [11]. This transmitter accepts a data stream in the format required by the display and reformats it for transmission over the ATM network.

An MMX is typically used with an NTSC video camera, microphones, and amplified stereo speakers. The

analog NTSC video signal generated by the video channel is usually fed into a "video-in-a-window" card installed in the host computer as shown in Fig. 1. In this configuration, the received video is displayed in a window on the normal workstation display eliminating the need for a separate video monitor. Other configurations are possible, including the use of a an auxilliary high-resolution monochrome display, a separate video monitor, as shown in Fig. 2, or the use of a laser disk player or other audio/video source.

II. THE ATMIZER AND "T-Y" CONNECTIONS

A. The Local CPU and Host Interface

A 16 MHz MC68030 Motorola microprocessor [4] serves as the CPU on the MMX. This CPU has 4 Megabytes of DRAM and 128 Kilobytes of EPROM mapped into its address space. The EPROM holds the embedded control software for the MMX. The DRAM is used for serial I/O buffering and CPU channel ATM cell management. Communication between the MMX and the host computer is handled through an RS232 serial port [3]. This serial port is used for control of the MMX devices.

A Multi-Function Peripheral Chip (MFP) [5] is also connected to the local CPU bus. This chip provides timing and interrupt services to the local CPU, as well as the serial port used to communicate with the host. A dedicated peripheral bus, the PBUS, is derived from the local CPU bus. This 16-bit multiplexed address/data bus is used to connect the ATMizer to the multimedia subsystem. The multimedia subsystem is controlled via the PBUS and the I^2C Bus [6].

The I^2C Bus is a multi-master serial bus used to control various components in the video multimedia channel. Two wires, serial data (SDA) and serial clock (SCL), carry information between devices connected to the bus. Both SDA and SCL are bidirectional lines, and data on the I^2C Bus can be transferred at a rate of up to 100 Kilobits per second (Kb/s) in standard mode, or up to 400 Kb/s in fast mode. The I^2C Bus is implemented using a Phillips/Signetics PCD8584 I^2C Bus controller [6]. The PCD8584 serves as the interface between the parallel local CPU bus and the serial I^2C Bus, and it allows the local CPU to communicate bidirectionally on a byte-wide basis with the I^2C Bus.

The local CPU interfaces to the ATM subsystem via a pair of FIFOs. These FIFOs are mapped into the address space of the CPU. To transmit an ATM cell, the CPU writes the cell, including header, to the TxFIFO. The

output of this FIFO is connected to the TxFIFO Bus which is described below. Incoming ATM cells are written into the RxFIFO from the RxFIFO Bus. The empty flag from this FIFO is polled by the CPU. Interrupt-driven operation, while possible, is not currently used.

Embedded code for the local MC68030 CPU is written in C [12] and currently compiled using the native C compiler on a NeXT host. A library of I/O function calls, e.g., *printf* and *scanf,* was written to replace the *stdio* C library to facilitate code development and debugging. A low-level monitor program, which is part of the embedded code, performs memory dumps, allows program execution, etc. For debugging or diagnostic purposes, it may be convenient to use a terminal and the on-board serial port to interact with this monitor.

B. The FIFO Bus and Network Interface

The interface to the ATM network is implemented using an Advanced Micro Devices TAXIchip transmitter/receiver pair [13]. In general, a TAXIchip transmitter/receiver pair provide a general-purpose interface for high-speed point-to-point communications over twisted-pair, coaxial, or fiber-optic media. The pair emulate a pseudo-parallel register: they load data into one side and output it on the other. With TAXIs, however, the input and output are separated by an arbitrarily long serial link. The ATMizer is configured to transmit and receive data via either twisted pair or multi-mode fiber using the transmission protocol specified by the ATM Forum for ATM systems operating at 155 Mb/s [14]. The "T" connection, using analog circuits, copies incoming signals and transmits them to both the local TAXI receiver and to the ATM link to the host. As described below, the ATM interface on the MMX can be programmed to ignore cells not specifically directed to either the local CPU or one of the multimedia channels. Similarly, the host must ignore cells bound for the MMX.

The local MC68030 CPU interfaces to the ATM network via two FIFOs, as shown in Fig. 4. These FIFOs, the TxFIFO and the RxFIFO, are used to transmit data to and receive data from the ATM network via the TxFIFO and RxFIFO Buses, respectively.

The TxFIFO Bus is also used by the multimedia channel for data transmission to the ATM network. In addition, this bus is also used to multiplex cells from the "Y" connection into the output ATM stream. Each multimedia channel, and the "Y" connection, use a FIFO to supply data to the TAXIchip transmitter. The programmable full flag from each FIFO connected to the TxFIFO Bus is sampled by the Transmitter Control and

Source Select block at the beginning of each ATM cell transmission cycle. A priority scheduling scheme is implemented by the Transmitter Control and Source Select block that provides the audio channel with the highest priority, the video channel the second-highest priority, the high-speed radiographic image channel the third-highest priority, the "Y" connection with the fourth-highest priority, and the local CPU the fifth-highest priority.

In general, the multimedia subsystem provides a data stream to the TxFIFO Bus that does not include the appropriate ATM header for transmission. The Transmitter Control and Source Select block operates in conjunction with the Header Generation block to insert the required headers. These headers are stored in tables in the Header Generation block by the local CPU via the PBUS. The capability also exists for the Header Generation block to provide headers for a data stream from the TxFIFO and the "Y" connection, although this is typically not done. The local CPU and the host normally generate complete cells for transmission that include the appropriate header and header error control (HEC) byte.

The RxFIFO Bus is used to supply data to the RxFIFO and to the multimedia subsystem. As the bytes of a cell are received from the ATM network, they are clocked into a byte-wide shift register, or Header Buffer. Once the entire header is in the Header Buffer, the VPI (Virtual Path Identifier) and VCI (Virtual Channel Identifier) fields are examined by the Route and Function block to identify the target channel. This function is performed using look-up tables initialized by the local CPU via the PBUS. The Receiver Control block routes each incoming cell to the appropriate channel by asserting the appropriate FIFO select signal to one of the FIFOs attached to the RxFIFO Bus. The Receiver Control block also monitors the output of the HEC circuit and can, optionally, disable delivery of corrupted cells.

Data received from the ATM network by the TAXIchip receiver for the local CPU is transferred over the RxFIFO Bus to the RxFIFO. The RxFIFO's programmable full flag is used to signal the local CPU that cells have arrived and are available for processing.

The tables used by the Route and Function block to route cells also contain information as to where the incoming cell is to be delivered. For cells received on a given virtual channel/path, any combination of the four-byte header, the header error check byte, or the 48-byte payload can be routed to the target channel. This allows the RxFIFO to receive a complete cell, including the header, while the multimedia subsystem receives only a data stream. This functionality also allows the RxFIFO to receive just the 4-byte header and the 48-byte payload or just

the header error check byte. The latter can be useful when measuring data throughput since only one byte need be processed per cell.

III. THE MULTIMEDIA SUBSYSTEM

A. The Video Channel

The video channel is shown in Fig. 5. This channel uses Phillips/Signetics video components [6] to digitize the incoming color video signal and to regenerate the output video signal. The input NTSC video signal is digitized using a Phillips/Signetics TDA8708 8-bit Analog-to-Digital (A/D) converter in conjunction with a Phillips/Signetics SAA9051 Digital Multistandard Decoder (DMSD). The DMSD produces a 12.2727 Megabyte per second Y:Cb:Cr (Luminance, Chroma Blue, Chroma Red) data stream during the active portion of each video line. The Y:Cb:Cr data is subsampled and multiplexed in 4:2:2 form. That is, four 8-bit luminance samples are multiplexed with two each subsampled chroma blue and chroma red samples. The DMSD is programmed by the local CPU via the I^2C Bus.

The output of the DMSD is fed into a JPEG Engine implemented using an LSI Logic JPEG chip set [15]. This three-chip set, shown operating in the encode mode in Fig. 6, consists of an L64765 Color and Raster-Block Converter, an L64735 Discrete Cosine Transform (DCT) Processor, and an L64745 JPEG Coder. The Color and Raster-Block Converter is used to convert the digital video data stream, which is produced by the DMSD in raster format, into block format. This format is required by the DCT Processor. Since the data is already in the Y:Cb:Cr format required by the DCT Processor, the L64765's color space conversion function and resampling filters are bypassed. The Raster-Block Converter consumes data as it is generated by the DMSD and starts feeding data to the DCT Processor after an eight-video-line delay. The Color and Raster-Block Converter's internal configuration registers are programmed by the local CPU via the PBUS.

In the transmission portion of the video channel, the DCT Processor is used to compute the forward DCT over the 8 x 8 data blocks generated by the Color and Raster-Block Converter. The DCT Processor accepts 8-bit unsigned pixel data and generates 11-bit signed DCT coefficients. The DCT Processor accepts data at video rates and has a 168-clock pipeline delay. Processing is suspended during horizontal blanking intervals, however, since new data is not available from the Raster-Block Converter. A Freeze Request (FRQ) output from the Raster-Block Converter is used to gate the clock on both the DCT Processor and the JPEG Coder during horizontal blanking

intervals. The coefficient data stream produced by the DCT Processor is fed to the JPEG Coder.

In the transmission portion of the video channel, the JPEG Coder encodes each video frame as specified in the JPEG standard [8]. The Coder performs quantization and has Differential Pulse-Code Modulation (DPCM) coding, run-length coding, and variable length (Huffman) coding capabilities. The Coder has eight internal tables: four quantization tables, two AC coding tables, and two DC coding tables. The JPEG Coder has a minimum encoding latency of 16 clock cycles. Not every DCT coefficient generates a code word, so the actual latency between a particular input and its corresponding output depends on the actual input data. The last DCT coefficient always causes a 32-bit code word to appear in the Coder's output buffer exactly 16 clock cycles after it is clocked into the Coder. The Coder's internal tables and the other operating parameters are downloaded from the local CPU via the PBUS.

The output of the JPEG Coder is buffered using a FIFO. Code words are transferred to this FIFO over a 16-bit bus. Data from the FIFO are processed by the Tag Stuffing block. This block inserts into the data stream the appropriate JPEG codes for start of field one, start of field two, and end of field. The information required to perform this function, i.e., the Last Code (LCODE) signal from the JPEG Coder, is passed to the Tag Stuffing block via an extra bit in the FIFO buffer.

The output of the Tag Stuffing block is again buffered in a FIFO. The output of this FIFO is connected to the TxFIFO Bus. This FIFO's programmable full flag, which is programmed via the PBUS to assert when the FIFO contains 48 bytes, is used to signal the ATMizer that a full cell payload is ready for transmission.

The operation of the transmission portion of the video channel is controlled by the Tx Control block. The operation of this channel, with the exception of the FIFO connected to the TxFIFO bus, is synchronous to the video sample clock created by the DMSD.

In the receive portion of the video channel, a FIFO attached to the RxFIFO Bus is used to buffer the incoming data stream from the ATMizer. Data passes from this FIFO through a Tag Stripping block to the JPEG Engine. The Tag Stripping block strips the start of field one, start of field two, and end of field tags from the data stream. This information is used by the Rx Control block to control the data flow through the JPEG Engine.

The JPEG Engine consists of the same three components used in the transmission portion of the video channel: the JPEG Coder, the DCT Processor, and the Color and Raster-Block Converter. Here, however, the

components are used in the decompression mode, rather than the compression mode.

The video data stream produced by the Raster-Block Converter in the JPEG Engine is written into a Frame Buffer constructed using Field Memories (FMEMs) [16]. FMEMs are similar to normal FIFOs in that read and write access my occur asynchronously. Unlike a conventional FIFO, however, read and write address pointers increment independently, possibly passing one another multiple times. Both address pointers can be reset to zero on initialization.

A Phillips/Signetics SAA7199 Digital Encoder (DENC) produces the output NTSC video signal from the Y:Cb:Cr data in the FMEM frame store. The DENC has all the circuitry necessary to perform the modulation required to convert the digital Y:Cb:Cr data into standard NTSC composite video [7]. The device has three 256 x 8-bit color look-up tables and built-in triple 9-bit Digital-to-Analog (D/A) Converters. The SAA7199's look-up tables and control registers are programmed by the local CPU via the I^2C Bus.

The operation of the receive portion of the video channel is controlled by the Rx Control block. The operation of this channel is data-driven and is asynchronous to the video pixel clock generated by the DENC subsystem. Since the data stream received by this channel was created by a remote transmission subsystem with an independent clock, synchronous operation would be very difficult to implement [9]. Over a period of time, the receive FIFO buffer would either overflow (transmitter clock too fast) or underflow (transmitter clock too slow). This problem would be compounded by cell-to-cell jitter introduced by the ATM network. The Rx Control block uses a gated clock scheme to operate the JPEG Engine only when data is available. The JPEG Engine is not synchronized with the DENC and runs at its peak rate when data is available. The overhead associated with the horizontal and vertical blanking intervals at the transmitter ensures that the JPEG Engine has sufficient bandwidth to eliminate data overflow problems. The frame buffer, which can be read and written independently, eliminates the problems caused by asynchronous image decoding/generation.

While it is possible for the DENC to display portions of two transmitted video frames during one display frame time, the use of a frame buffer as a solution to the synchronization problems involved in transmitting video over an ATM network has been found to be a very acceptable solution [9]. Since two video frames typically differ by only a small amount, any possible artifacts due to the use of a frame buffer are not perceptible.

B. The Audio Channel

The audio channel is shown in Fig. 7. This channel digitizes an input analog stereo audio signal, i.e., both left and right audio channels, and produces a CD-quality digital audio data stream for transmission by the ATMizer. This channel also accepts a digital audio stream from the ATMizer and generates an output analog stereo audio signal. A monaural input and a monaural output are also supported by the system. A TMS320C25 Digital Signal Processor (DSP) [17] provides volume, loopback, mixing, and amplification functions. The functions of the audio channel are controlled via a 2 Kilobyte two-port memory. One port of this memory is connected to the local DSP bus, and the other port is connected to the PBUS.

The audio channel digitizes the input stereo audio signal using a Crystal Semiconductor Corporation CS4216 stereo audio codec [18]. The CS4216 oversamples at 64 times the output word rate of 44.1 kHz and uses a three-stage digital finite impulse response (FIR) filter to achieve greater than 80 dB signal-to-noise ratio over a 10 Hz to 22 kHz bandwidth. The CS4216 generates 16-bit values for both the left and the right inputs. The resulting serial data stream is parallelized by the DSP and buffered in a FIFO attached to the TxFIFO Bus. The 48-byte payload of each transmitted cell contains twelve digital audio data words, i.e., twelve 16-bit left and 16-bit right channel sample pairs. With this sampling rate and payload format, one audio cell is transmitted every 272 us.

In the receive portion of the audio channel, a FIFO connected to the RxFIFO Bus is used to buffer the incoming data stream from the ATMizer. Data passes through this FIFO to the DSP. From the DSP, data flows to the output D/A converter. The D/A portion of the Crystal Semiconductor CS4216 includes an on-chip 8x digital interpolation filter followed by a 64x oversampled delta-sigma modulator. The D/A converter achieves greater than 80 dB dynamic range over the audio band and 0.2 dB of passband ripple.

Since the audio sampling frequency at the transmitter can be slightly different from the playback frequency at the receiver, a mechanism is needed to compensate for data overflow or underflow. The problem here, while similar to that described above for the video channel, requires a more sophisticated solution. A solution employing a circular buffer, as is used in the video channel, would result in degraded audio performance when the write and read pointers crossed due to differing transmitter and receiver clock frequencies.

The solution to the synchronization problem implemented by the DSP involves the controlled duplication or deletion of audio samples. The required duplication or deletion is based on the amount of audio data contained in

the FIFO buffer. If this FIFO is between one quarter and three quarters full, as indicated by the FIFO's flags, data is neither duplicated nor deleted as it passes through the DSP. If the FIFO is less than one quarter full, one left-right sample pair of each cell is duplicated as data is passed from the FIFO to the D/A converter. This action results in a build-up of data in the FIFO over time, and, eventually, the FIFO will become one quarter full. If the FIFO is empty, zero sample values are transmitted to the D/A converter until new data arrives.

If the FIFO is more than three quarters full, the DSP deletes the first left-right sample pair of each cell. This action results in a reduction of data in the FIFO over time, and, eventually, the FIFO will become three-quarters full. If the FIFO is full, an error message is sent to the local CPU and the FIFO is reset. In routine operation, this situation should not occur.

Initialization and control of the audio channechannel is performed by the local CPU via the PBUS and the two-port memory. The transmit and receive FIFO flags are programmed by the DSP at power-up and upon command from the local CPU via the PBUS. The transmitter and receiver can be independently turned on and off by the DSP upon command from the local CPU via the PBUS. The audio volume settings are adjusted via memory locations in the two-port memory as is the sampling rate. Eight sampling rates from 44.1 KHz to 7.35 KHz are supported as are monaural and stereo payloads.

C. The High-Speed Radiographic Image Channel

The high-speed radiographic image channel is shown in Fig. 8. This channel receives a radiographic image data stream from RxFIFO Bus. This data stream does not contain the four-byte ATM header or the header error check byte: these bytes are stripped by the ATMizer subsystem. The data stream is buffered in a FIFO as it is received. The empty flag from this FIFO is monitored by the Control block. When data is available, it is transmitted to the high-resolution monochrome display via a TAXIchip transmitter operating at a data rate of 40 Mb/s using the required protocol [11].

Currently, a dedicated mini-ATMizer transmitter [11] is used in conjunction with dedicated radiographic image server to create the ATM data stream used by the radiographic image channel. The required data stream could also be generated by an image server equipped with an ATM interface card.

IV. SOFTWARE

A. Overview

The software for the MMX consists of embedded software running on the local CPU and applications running on the host computer. The embedded control program uses interrupt driven I/O over the serial port to process control commands from the host.

B. Embedded Software

The embedded software on the local CPU provides the host with control functions for the various devices on the MMX. For ATM communication, it provides the ability to filter and route incoming packets to the various devices based on VPI and VCI pairs. For the video channel, there are functions to change the JPEG Q-factor in both the encoder and decoder. The audio functions provide volume control and mixing of the audio outputs.

The memory map of the MMX is compatible with that of the Washington University Multimedia System (MMS) [2,19]. Thus, an MMS provides a convenient development environment. Software can be developed and tested in the DRAM on the MMS being burning the EPROMs for the MMX.

C. MMX Host Software

The software for the host computer demonstrates the capability of the MMX hardware to support a multimedia workstation. The main application in this suite is VideoExchange [20], a video conferencing tool. This program is also used with the Washington University Multimedia System, although it has been modified to control the MMX through the serial port, rather than across the NeXTBus.

Another application for the MMX is RIVA, the Radiology Image Viewing Application [21,22]. RIVA allows physicians to retrieve medical images, via the high-speed radiographic image channel, along with the appropriate written reports. Medical video used for ultrasound and flouroscopic examinations and video reports stored on laser disk can also be retrieved via the video channel. This allows physicians at remote locations to conference to discuss patients, and, in particular, examine their radiographic images and medical video clips.

V. RESULTS AND DISCUSSION

The MMX has been operational since mid-1993. Twenty of the systems are currently deployed in the Department of Computer Science ATM network and are being used for demonstration and research purposes. The

issues associated with both video conferencing and teleradiology in a broadband ATM environment are being studied using the installed base of systems.

Bus bandwidth limitations on current workstation-class machines make real-time, bus-based, full-rate NTSC video applications difficult, if not impossible, to implement. In addition, bus-based applications consume CPU cycles that could be used for other purposes. One typical method used to address this limitation is to reduce the spatial and temporal resolution of the transmitted video signal. The Washington University MultiMedia eXplorer achieves real-time, full-rate video, audio, and radiographic image service by bypassing the host bus. This frees the host bus for normal network traffic. Until host bus and CPU speeds increase significantly, the approach used by the MMX can be used to achieve a high-quality multimedia solution.

While the MMX uses a large number of fairly complex integrated circuits, all of the components used in its construction are commercially available. Physically, each MMX consists of an expansion chassis with two 27.5 cm by 27.0 cm cards and two daughter cards. One of the large cards contains the ATMizer, and the other contains the video and image channels. These cards are connected by a PBUS/I^2C Bus connector. A third 11.8 cm by 26.6 cm card, which connects to only the PBUS/I^2C Bus connector, contains the audio channel. A fourth 14.9 cm by 27.0 cm card contains the protocol-specific portion of the ATM interface, as well as the "T" and "Y" connections. This allows one MMX design to work in networks with different speeds with only a change of this daughter card.

VI. CONCLUSIONS

The Washington University MultiMedia eXplorer is a complete multimedia system capable of transmitting and receiving full-rate, JPEG-compressed, NTSC video, CD-quality audio, and high-resolution radiographic images over the Washington University broadband ATM network. Real-time, full-rate operation is supported through the use of dedicated hardware which does not require host bus bandwidth for operation. High-speed host communication is supported in conjunction with the MMX via the "T" and "Y" functionality of the system. The MultiMedia eXplorer demonstrates in a dramatic manner the capability and applicability of broadband ATM networks for multimedia applications. In particular, the physician's workstation application clearly demonstrates the advantages of such a system when used in a remote radiology environment.

VII. ACKNOWLEDGEMENTS

The authors would like to thank, G, James Blaine, Tom Chaney, Pierre Costa, Pete Flugstad, Steve Moore, Kurt Smith, and Bob Whitman for their contribution to the effort described here. The MMX development was supported by Project Zeus sponsors SynOptics, Ascom Timeplex, Southwestern Bell and Washington University. The MMX was based upon a design developed in the physician's workstation project with the sponsorship of Southwestern Bell, Washington University and the Mallinckrodt Institute of Radiology.

REFERENCES

[1] J.R. Cox, Jr., M.E. Gaddis, and J.S. Turner, "Project Zeus," *IEEE Network.*, vol 7, no. 2, pp. 20-30, March 1993.

[2] W.D. Richard, J.R. Cox, Jr., B. Gottlieb, and K. Krieger, "The Washington University Multimedia System," *ACM Multimedia Systems* (in press).

[3] *Interface Between Data Terminal Equipment and Data Communication Equipment Employing Serial Binary Data Interchange.* Washington, D.C.: Electronic Industries Association, 1969.

[4] *Motorola MC68030 Enhanced 32-Bit Microprocessor User's Manual.* Englewood Cliffs, NJ: Prentice-Hall, 1990.

[5] *MC68901 Multi-Function Peripheral Data Sheet,* Motorola Semiconductor Products, Inc., Jan., 1984.

[6] *The Video Data Handbook,* Signetics Company, 1991.

[7] K.B. Benson, ed., *Television Engineering Handbook.* New York: McGraw Hill, 1986.

[8] G.K. Wallace, "The JPEG still picture compression standard," *Commun. ACM,* vol. 34, no. 4, pp. 31-44, Apr. 1991.

[9] W.D. Richard, "ATM video transmission: Considerations and an implementation," *Proc. 1st Int. Conf. Comp. Commun. and Networks,* San Diego, CA, June 1992, pp. 207-210,

[10] J.R. Cox, Jr., R.G. Jost, T. Monsees, S. Ramamurthy, and M. Karlsson, "An inexpensive electronic viewbox," *SPIE,* vol. 914, pp. 1218-1224, 1988.

[11] D.M. Zar and W.D. Richard, "Mini-ATMizer User's Guide and Technical Manual," WUCS-92-50, Department of Computer Science, Washington University, St. Louis, MO, 1992.

[12] B.W. Kernigan and D.M. Ritchie, *The C Programming Language,* Englewood Cliffs, NJ: Prentice-Hall, 1988.

[13] *TAXIchip Integrated Circuits Technical Manual,* Advanced Micro Devices, 1989.

[14] *ATM User-Network Specification Version 2.0,* The ATM Forum, June 1, 1992.

[15] *JPEG Chipset Technical Manual,* LSI Logic Corporation, December, 1991.

[16] *MOS Memory Data Book,* Texas Instruments, Inc., 1991.

[17] *TMS320C2X User's Guide,* Texas Instruments, Inc., 1987.

[18] *Crystal Semiconductor CS4216 Stereo Audio Codec Data Sheet,* Crystal Semiconductor Corporation, March,

1992.

[19] K. Krieger, "An ATM Multimedia Interface for the MDWS," WorkNote_ERL_93_2, Electronic Radiology Laboratory, Washington University, St. Louis, MO, 1993.

[20] B. Gottlieb, "VideoExchange: A Multimedia Conferencing Tool," ARL Working Note 93-10, Applied Research Laboratory, Department of Computer Science, Washington University, St. Louis, MO, 1993.

[21] J.R. Cox, Jr., S.M. Moore, R.A. Whitman, G.J. Blaine, R.G. Jost, L.M. Karlsson, T.L. Monsees, G.L. Hassen, T.C. David, "Rapid Display of Radiographic Images", *SPIE Medical Imaging V: PACS Design and Evaluation,* vol. 1446, pp. 40-51, 1991.

[22] M.S. Wimmer, G.J. Blaine, R.G. Jost, S.M. Moore, and J.W. Studt, "A Distributed Approach to Integrated Inquiry and Display for Radiology", *Journal of Medical Systems,* vol. 15, no. 4, pp. 299-309, 1991.

Biographies

William D. Richard received the B.S., M.S., and Ph.D degrees in electrical engineering from the University of Missouri-Rolla in 1983, 1985, and 1988, respectively. He joined Department of Electrical Engineering at Washington University in St. Louis as an Assistant Professor in 1988. His research interests include ultrasonic imaging, medical instrumentation, and computer architecture.

Dr. Richard is a member of the Institute of Electrical and Electronics Engineers, the American Society for Engineering Education, and the Society of Automotive Engineers. He is also a member of the Sigma Xi, Phi Kappa Phi, Phi Eta Sigma, Eta Kappa Nu, and Tau Beta Pi honoraries.

Jerome R. Cox, Jr., received the S.B., S.M., and Sc.D. degrees in electrical engineering from the Massachusetts Institute of Technology, Cambridge, in 1947, 1949, and 1954, respectively.

He is the Welge Professor of the Department of Computer Science at Washington University in St. Louis. Joining the faculty in 1955 as an Assistant Professor of Electrical Engineering, he was appointed Associate Professor in 1958 and Professor in 1961. From 1964 to 1975, he was Director of the Biomedical Computer Laboratory. He also served as Chairman of the Department of Computer Science from 1975 to 1991. He is currently Director of the Applied Research Laboratory and Research Director of the Electronic Radiology Laboratory. His research interests are digital communication, computer systems design, and electronic radiology.

Dr. Cox is a Senior Member of the Institute of Medicine of the National Academy of Sciences and is a Fellow of the Acoustical Society of America, the Institute of Electrical and Electronics Engineers, and the American Medical Informatics Association.

Maynard Engebretson received a BEE degree from the University of Minnesota in 1958 and MSEE and D.Sc. degrees from Washington University in 1963 and 1970, respectively. His research interests have focussed on the application of computers and digital signal processing methods to problems of hearing and deafness at Central Institute for the Deaf where he was Associate Director of Research from 1987 to 1992. He is a member of the faculty in the Departments of Speech and Hearing, Computer Science, and Otolaryngology at Washington University in St. Louis, Missouri. His current interests include multimedia, speech processing, acoustic array processing, and active noise control. He is a member of the Institute of Electrical and Electronics

Engineers, the Acoustical Society of America, Sigma Xi, and Eta Kappa Nu.

Jason Fritts received the B.S. degree in electrical engineering from Washington University in St. Louis in 1992. His graduate work involves hardware development for the MMX multimedia and ATMizer subsystems.

Brian L. Gottlieb received the B.S. and M.S. degrees in computer science from Washington University in St. Louis in 1991 and 1993, respectively. He also received the B.S. and M.S. degrees in electrical engineering from Washington University in St. Louis in 1991 and 1993, respectively. His graduate work included software development for the MMX system.

Craig Horn received the B.S. degree in electrical engineering from Washington University in St. Louis in 1992. His graduate work involves hardware development for the MMX multimedia and ATMizer subsystems.

FIGURE CAPTIONS

Fig. 1. An MMX is typically configured with a video camera, stereo microphones, and amplified stereo speakers. The video output from the MMX is fed into a "video-in-a-window" card in the workstation in the configuration shown.

Fig. 2. An MMX can also be configured with a separate NTSC video display, a separate high-resolution radiographic image display, or both.

Fig. 3. The MMX system block diagram.

Fig. 4. The ATMizer and "T-Y" connections.

Fig. 5. The video channel.

Fig. 6. The JPEG Engine.

Fig. 7. The audio channel.

Fig. 8. The high-speed radiographic image channel.

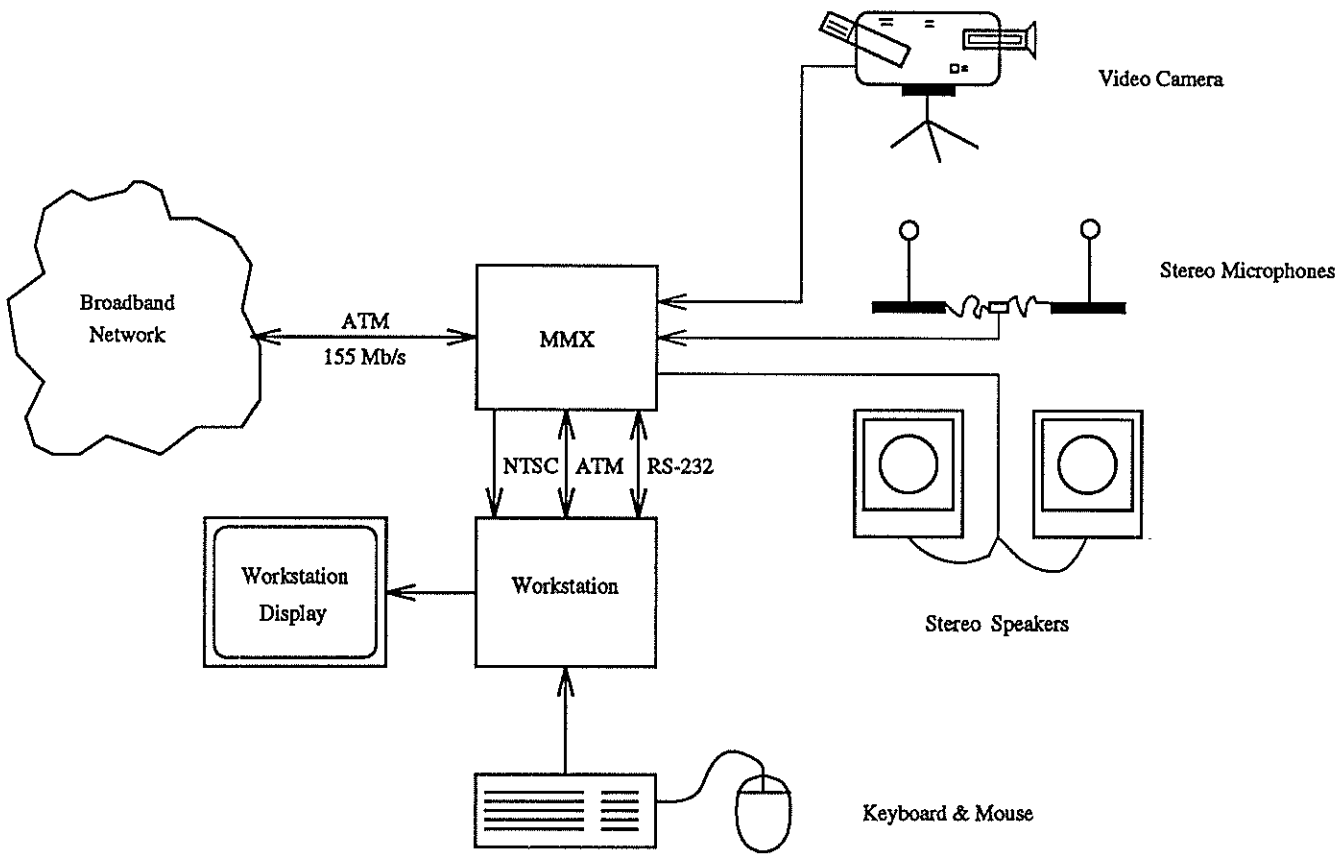

Fig. 1. The MMX is typically configured with a video camera, stereo microphones, and amplified stereo speakers. The video output from the MMX is fed into a "video-in-a-window" card in the workstation in the configuration shown.

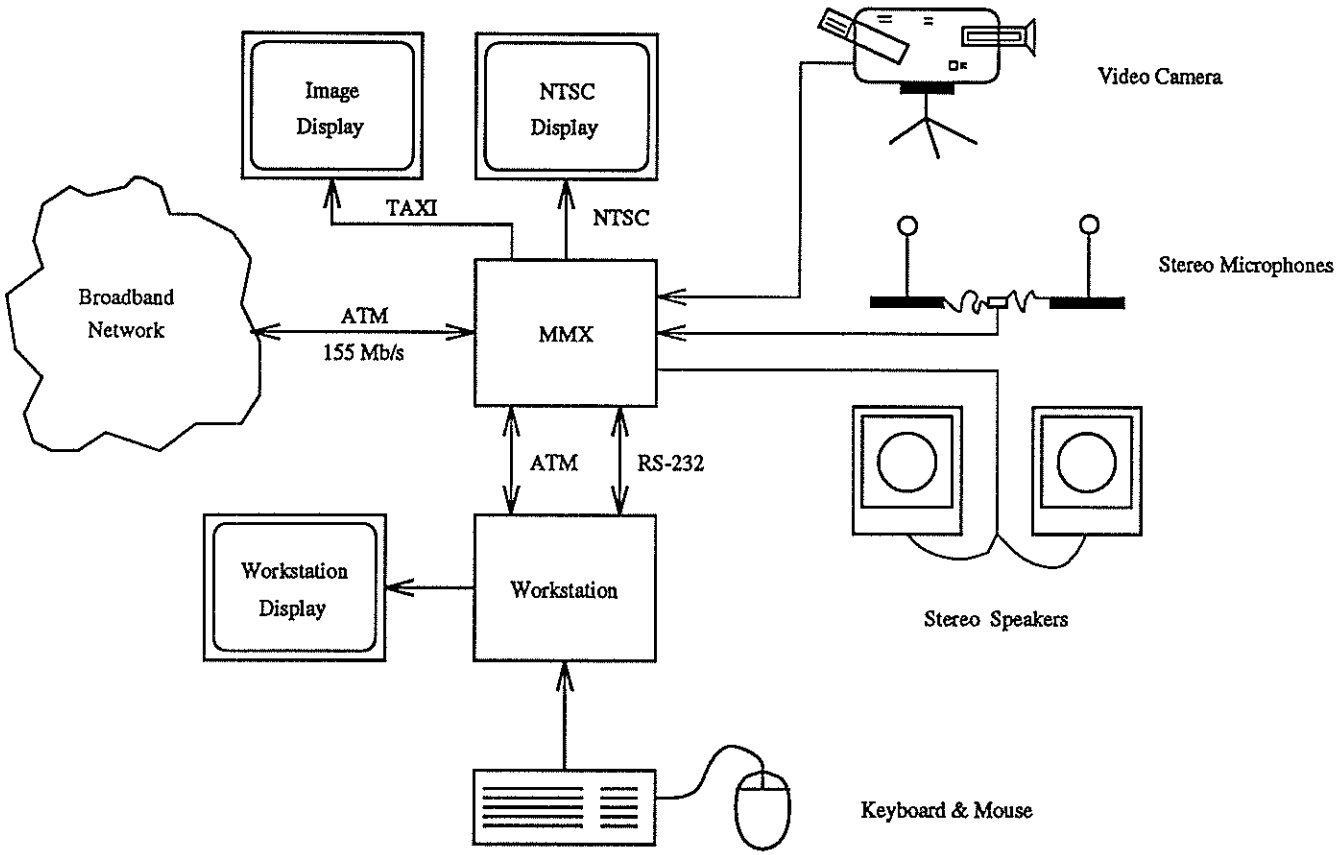

Fig. 2. An MMX can also be configured with a separate NTSC video display, a separate high-resolution radiographic image display, or both.

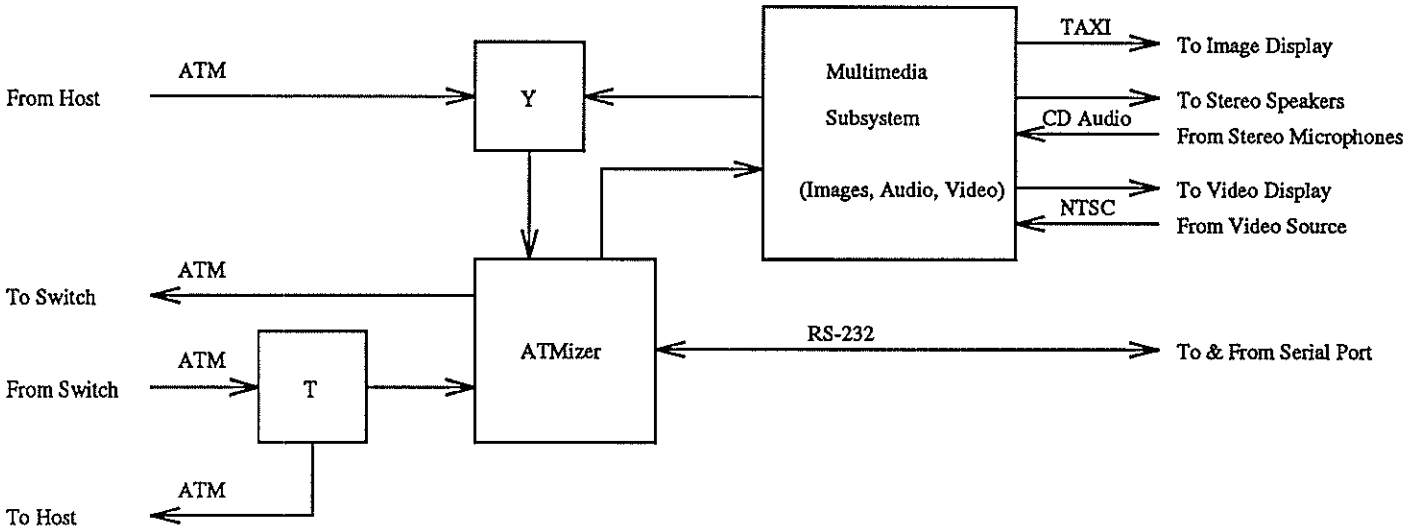

Fig. 3.. The MMX system block diagram.

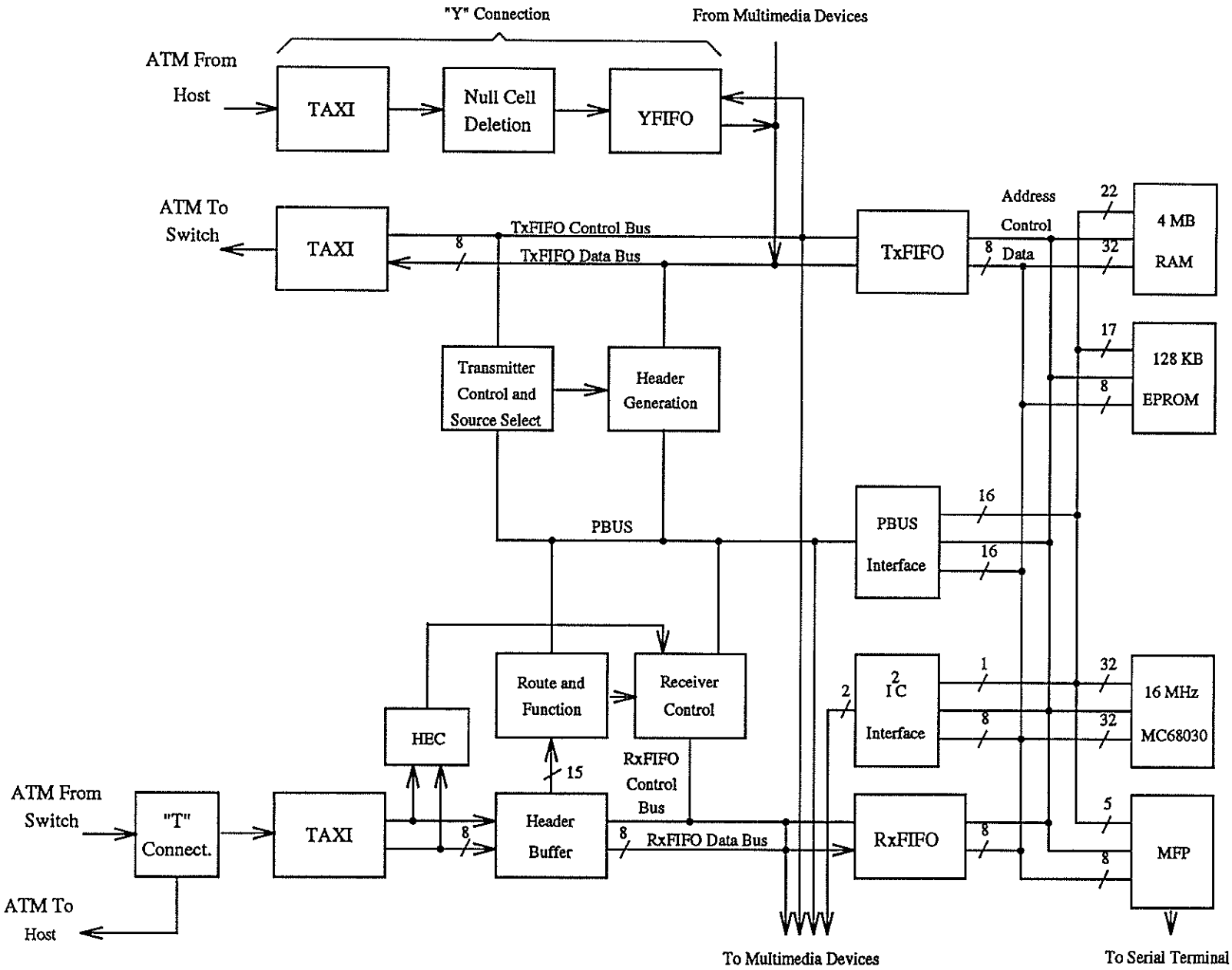

Fig. 4. The ATMizer and "T-Y" connections.

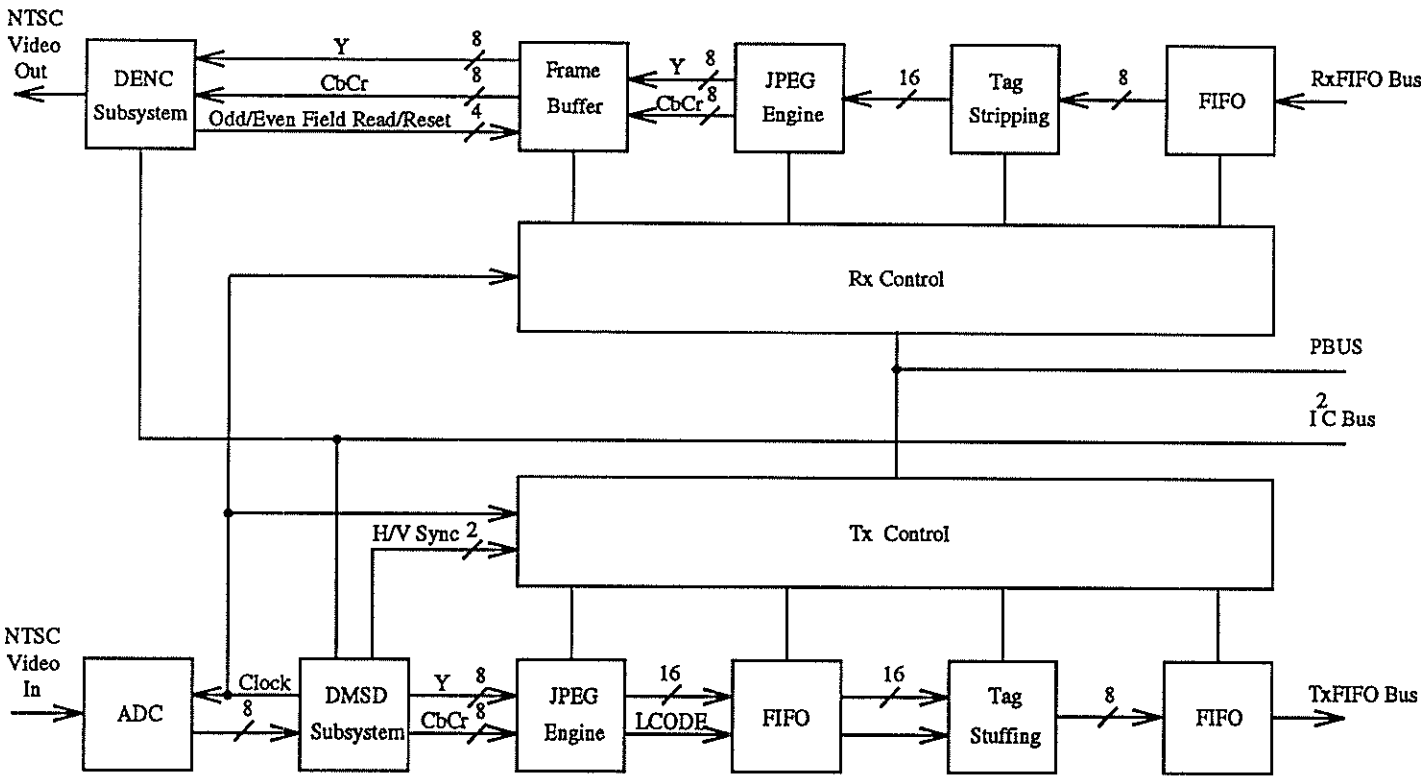

Fig. 5. The video channel.

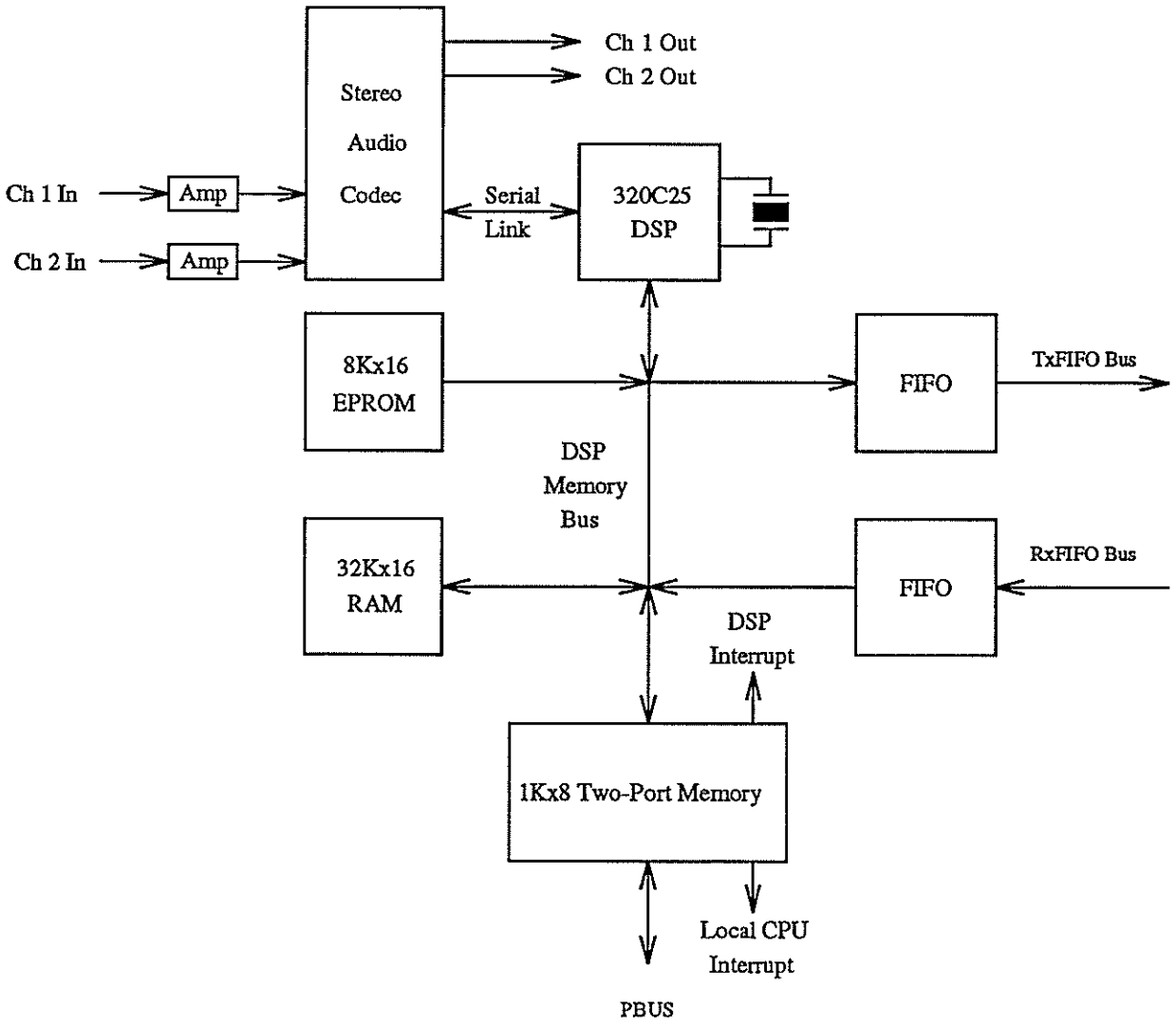

Fig. 7. The audio channel.

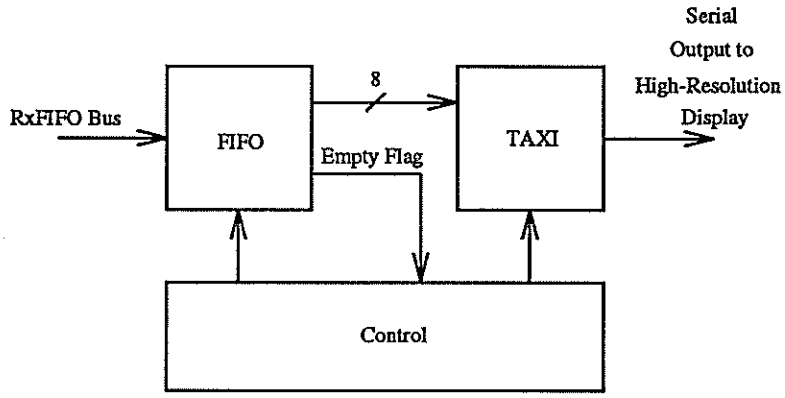

Fig. 8. The high-speed radiological image channel.

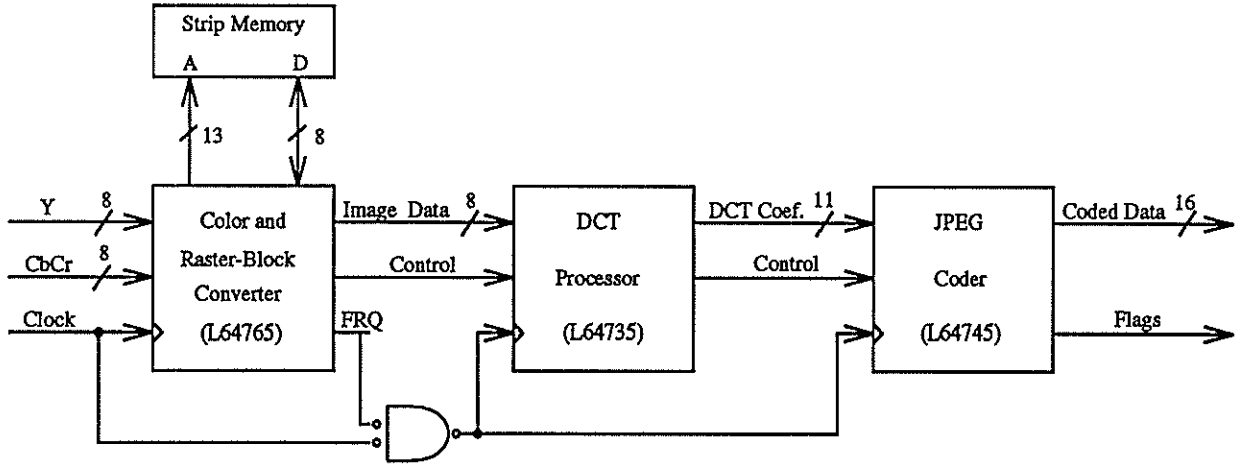

Fig. 6. The JPEG Engine.