

Washington University in St. Louis

## Washington University Open Scholarship

---

All Computer Science and Engineering  
Research

Computer Science and Engineering

---

Report Number: WUCS-93-1

1993-01-01

### The Washington University Multimedia System

William D. Richard, Jerome R. Cox Jr., Brian Gottlieb, and Ken Krieger

The Washington University Multimedia System (MMS) is a complete multimedia system capable of transmitting and receiving video, audio, and radiological images, in addition to normal network traffic, over the Washington University broadband ATM network. The MMS 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. The video subsystem encodes and decodes JPEG compressed video using two hardware compression engines. The audio subsystem encodes and decodes CD-quality stereo audio. The high-speed radiological image subsystem reformats radiological image data transmitted by a... [Read complete abstract on page 2.](#)

Follow this and additional works at: [https://openscholarship.wustl.edu/cse\\_research](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.; Gottlieb, Brian; and Krieger, Ken, "The Washington University Multimedia System" Report Number: WUCS-93-1 (1993). *All Computer Science and Engineering Research*. [https://openscholarship.wustl.edu/cse\\_research/296](https://openscholarship.wustl.edu/cse_research/296)

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

## The Washington University Multimedia System

William D. Richard, Jerome R. Cox Jr., Brian Gottlieb, and Ken Krieger

### Complete Abstract:

The Washington University Multimedia System (MMS) is a complete multimedia system capable of transmitting and receiving video, audio, and radiological images, in addition to normal network traffic, over the Washington University broadband ATM network. The MMS 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. The video subsystem encodes and decodes JPEG compressed video using two hardware compression engines. The audio subsystem encodes and decodes CD-quality stereo audio. The high-speed radiological image subsystem reformats radiological image data transmitted by a dedicated ATMizer for presentation on a high-resolution monochrome display. Although the MMS can be easily modified to operate with any host, the current implementation is based on a NeXT computer. This paper describes the architecture of the MMS, 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.

**The Washington University Multimedia System**

**William D. Richard, Jerome R. Cox, Jr., Brian Gottlieb and Ken Krieger**

**WUCS-93-01**

**January 1993**

**Department of Computer Science  
Washington University  
Campus Box 1045  
One Brookings Drive  
St. Louis MO 63130-4899**



# The Washington University

## Multimedia System

WILLIAM D. RICHARD<sup>1</sup>, JEROME R. COX<sup>2,3</sup>

BRIAN GOTTLIEB<sup>2,3</sup>, AND KEN KRIEGER<sup>1,3</sup>

Washington University in St. Louis

<sup>1</sup>Department of Electrical Engineering

<sup>2</sup>Department of Computer Science

<sup>3</sup>Electronic Radiology Laboratory

One Brookings Drive

St. Louis, Missouri 63130

**ABSTRACT--**The Washington University MultiMedia System (MMS) is a complete multimedia system capable of transmitting and receiving video, audio, and radiological images, in addition to normal network traffic, over the Washington University broadband ATM network. The MMS 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. The video subsystem encodes and decodes JPEG compressed video using two hardware compression engines. The audio subsystem encodes and decodes CD-quality stereo audio. The high-speed radiological image subsystem reformats radiological image data transmitted by a dedicated ATMizer for presentation on a high-resolution monochrome display. Although the MMS can be easily modified to operate with any host, the current implementation is based on a NeXT computer. This paper describes the architecture of the MMS, 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.

## I. INTRODUCTION

### A. The Washington University ATM Network

Over the past four years, the Applied Research Laboratory at Washington University has developed a prototype Asynchronous Transfer Mode (ATM) switch and deployed a testbed broadband network [1]. The current network configuration is based on four geographically dispersed ATM switches connected by more than 200 miles of single-mode fiber. This network supports the transmission of real-time digital video and audio and the rapid display of high-resolution medical images, as well as other forms of communication, at channel rates of 100 Megabits/second (Mb/s).

The prototype ATM switch [2] has 16 ports each conveying 53-byte ATM cells into and out of the internal switch fabric. A five byte header is used to route a cell through the binary switch elements of the switch fabric. The switch fabric is made up of three circuit boards: two network boards and a broadcast translation board. Each network board contains 32 custom chips implementing the binary switch elements. These boards are identical and function either as a copy network or as a routing network as determined by a programming pin on each chip. The broadcast translation board contains 16 custom broadcast translation circuits. Five additional boards are required to complete the switch: four four-port port processor boards and a switch module interface board.

### B. The Washington University Multimedia System

The Washington University MultiMedia System (MMS) is a complete multimedia system capable of transmitting and receiving video, audio, and radiological images, in addition to normal network traffic, over the Washington University broadband ATM network. Although the system can be easily modified to operate with any host, the current implementation is based on a NeXT computer [3].

The NeXT interface, the interface to the ATM network, and the interface to the multimedia subsystems are implemented by the ATMizer subsystem shown in Fig. 1. This 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], a NeXTBus Interface Chip (NBIC) [6], and interfaces to the ATM network and to the multimedia subsystems. Three dedicated buses, the PBUS, the TxFIFO Bus, and the Rx FIFO Bus, are used to connect the multimedia subsystems to the ATMizer. A fourth dedicated bus, the I<sup>2</sup>C Bus [7], is used to control the video and audio components in the multimedia subsystems.

There are currently three multimedia subsystems used with the ATMizer. The video subsystem, shown in

Fig. 2, digitizes an input color NTSC video signal [8], compresses the digital video data stream using a hardware Joint Photographic Experts Group (JPEG) [9] 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 subsystem 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 [10]. Two hardware JPEG engines are used by the video subsystem so that video can be compressed for transmission and decompressed for display simultaneously.

The second multimedia subsystem, shown in Fig. 3, digitizes an input analog stereo audio signal, i.e., both left and right channels, and produces a CD-quality digital audio data stream for transmission by the ATMizer. This subsystem 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 subsystem, shown in Fig. 4, accepts high-resolution radiological images from the ATMizer via the RxFIFO Bus and reformats them for transmission to a high-resolution monochrome display [11]. The radiological images used with this subsystem are served to the ATM network via a dedicated "mini-ATMizer" transmitter [12]. This transmitter, shown in Fig. 5, accepts a data stream in the format required by the display and reformats it for transmission over the ATM network.

## II. THE ATMIZER

### A. The Local CPU and Host/NeXT Interface

A 16 MHz MC68030 Motorola microprocessor [4] serves as the local CPU on the ATMizer. This CPU has 4 Megabytes of DRAM and 128 Kilobytes of EPROM mapped into its address space. A portion of the DRAM is used to implement a pair of message passing queues for communication between the NeXT host and the local CPU. A portion of the DRAM is also used for segmentation and reassembly of IP (Internet Protocol) packets when the system software is configured to run IP over the ATM network. The DRAM is also used to hold embedded software during debugging. The on-board EPROM is used for long-term, non-volatile storage of embedded code.

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 a serial port. 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 subsystems. The multimedia subsystems are controlled via this bus and the I<sup>2</sup>C Bus [7].

The I<sup>2</sup>C Bus is a multi-master serial bus used to control various components in the audio and video multimedia subsystems. 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<sup>2</sup>C 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<sup>2</sup>C Bus is implemented using a Phillips/Sigmetics PCD8584 I<sup>2</sup>C Bus controller [7]. The PCD8584 serves as the interface between the parallel local CPU bus and the serial I<sup>2</sup>C Bus, and it allows the local CPU or the NeXT host to communicate bidirectionally on a byte-wide basis with the I<sup>2</sup>C 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. As described below, the transmit and receive FIFOs can also be written to and read from, respectively, by the NeXT host via the NeXTBus.

An NBIC [6] is used to provide the interface between the local CPU and the NeXTBus. This 144-pin CMOS device physically resides between the local bus and the NeXTBus. It participates in arbitration on both buses and stores up to two transactions worth of data for pipelined store and forward write transfers in both directions. The NBIC connects directly to the local MC68030 processor and performs all MC68030 transactions, including bursts, except for a few misaligned data transfers. The NBIC is capable of communicating with 8-, 16-, and 32-bit devices connected to the local bus.

Since the NBIC accepts transactions going in both directions, the NBIC can be both a local bus master and a local bus slave. The local memory can be mapped into the address space of the NeXT host, and this feature allows the NeXT host to perform read or write operations to local memory. The transmit and receive FIFOs are also memory mapped, so they too can be accessed directly by the NeXT host if desired.

Embedded code for the local MC68030 CPU is written in C [13] and compiled on the 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. During development, compiled code is downloaded from the NeXT host to local DRAM using the memory-mapped capability described above. A low-level monitor program running on the ATMizer performs memory dumps, allows program execution, etc. Using a terminal, users can interact with this monitor via the on-



board serial port for debugging or diagnostic purposes.

## B. The FIFO Bus and Network Interface

The interface to the ATM network is implemented using an Advanced Micro Devices TAXIchip transmitter/receiver pair [14]. In general, a TAXIchip transmitter/receiver pair provide a general-purpose interface for high-speed point-to-point communications over 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 TAXIchip transmitter and receiver used with the prototype Washington University ATM switch operate at 100 Mb/s. The ATMizer is configured to transmit and receive data via either twisted pair or multi-mode fiber. The transmission protocol specified by the ATM Forum for future ATM systems uses a newer version of the TAXIchip transmitter/receiver pair operating at 155 Mb/s [15].

The local MC68030 CPU interfaces to the ATM network via two FIFOs, as shown in Fig. 1. These FIFOs, the TxFIFO and the RxFIFO, are used to transmit data to and receive data from the ATM network, respectively. Since the TxFIFO and the RxFIFO are memory mapped, they can also be accessed directly by the NeXT host. Data written into the TxFIFO by the local CPU or the NeXT host is transferred over the TxFIFO Bus to the TAXIchip transmitter for transmission.

The TxFIFO Bus is also used by the multimedia subsystems for data transmission. Each subsystem uses 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 video subsystem with the highest priority, the audio subsystem the second-highest priority, the high-speed radiological channel the third-highest priority, and the CPU the fourth-highest priority. A spare transmission channel is also available on the TxFIFO Bus, and it is assigned the lowest priority.

In general, the multimedia subsystems provide 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. Since these tables are memory mapped, they can also be written directly by the NeXT host. The capability also exists for the Header Generation block to provide headers for a data stream from the TxFIFO, although this is typically not done. The local CPU and the NeXT 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 subsystems. 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 VCI (Virtual Channel Identifier) and VPI (Virtual Path Identifier) fields are examined by the Route and Function block to identify the target subsystem. This function is performed using look-up tables initialized by the local CPU or the NeXT host via the PBUS. The Receiver Control block routes each incoming cell to the appropriate subsystem 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 disables delivery of corrupted cells if this feature is enabled.

Data received from the ATM network by the TAXIchip receiver for the local CPU is transferred over the RxFIFO Bus to the RxFIFO. In the current design, the RxFIFO's programmable full flag is used to signal the local CPU that cells have arrived and are available for processing. This flag can also be read directly by the NeXT host.

The tables used by the Route and Function block to route cells also contain information as to which part of the incoming cell is to be delivered to the specified subsystem. 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 subsystem. This allows the RxFIFO to receive a complete cell, including the header, while the multimedia subsystems receive 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 must be processed per cell.

### III. THE MULTIMEDIA SUBSYSTEMS

#### A. The Video Subsystem

The video subsystem is shown in Fig. 2. This subsystem uses Phillips/Signetcs video components [7] to digitize the incoming color video signal and to regenerate the output video signal. The input NTSC video signal is digitized using a Phillips/Signetcs TDA8708 8-bit Analog-to-Digital (A/D) converter in conjunction with a Phillips/Signetcs 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. 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 or

the NeXT host via the I<sup>2</sup>C Bus.

The output of the DMSD is fed into a JPEG Engine implemented using an LSI Logic JPEG chip set [16]. 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 or the NeXT host via the PBUS.

In the transmission portion of the video subsystem, 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 subsystem, the JPEG Coder encodes each video frame as specified in the JPEG standard [9]. 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 or the NeXT host 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 subsystem is controlled by the Tx Control block. The operation of this subsystem, 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 subsystem, a FIFO attached to the Rx FIFO 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 subsystem: 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 is written into a Frame Buffer constructed using Field Memories (FMEMs) [17]. FMEMs are similar to normal FIFOs in that read and write access may occur asynchronously. Unlike a conventional FIFO, however, data may be read as many times as desired after it is written. Addressing is controlled by write address and read address pointers which must be reset to zero before memory access begins.

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 [8]. 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 or the NeXT host via the I<sup>2</sup>C Bus.

The operation of the receive portion of the video subsystem is controlled by the Rx Control block. The operation of this subsystem is data-driven and is asynchronous to the video pixel clock generated by the DENC subsystem. Since the data stream received by this subsystem was created by a remote transmission subsystem with an independent clock, synchronous operation would be very difficult to implement [10]. 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 the possibility of 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 [10]. 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 Subsystem

The audio subsystem is shown in Figure 3. This subsystem digitizes an input analog stereo audio signal, i.e., both left and right channels, and produces a CD-quality digital audio data stream for transmission by the ATMizer. This subsystem 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. An Audio Interface block provides tone and volume, loopback, mixing, and amplification functions. The functions of the Audio Interface block are controlled via the I<sup>2</sup>C Bus.

The audio subsystem digitizes the input stereo audio signal using a Crystal Semiconductor Corporation CS5326 delta-sigma stereo A/D converter [18]. The CS5326 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 a 94 dB signal-to-noise ratio and 0.0015% total harmonic distortion over a 10 Hz to 22 kHz bandwidth. The CS5326 generates 16-bit values for both the left and the right inputs. The resulting serial data stream is parallelized by the Payload Assembler 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 right channel sample pairs. With this sampling rate and payload format, one audio cell is transmitted every 272  $\mu$ s.

In the receive portion of the audio subsystem, 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 Data Rate Adjuster. From the Data

Rate Adjuster, data flows to the output D/A converter. A Crystal Semiconductor CS4328 stereo D/A [18] is used which features an 8x digital interpolation filter followed by a 64x oversampled delta-sigma modulator. The D/A converter achieves a 95 dB dynamic range over the audio band and 0.001 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 subsystem, requires a more sophisticated solution. A solution employing a circular buffer, as is used in the video subsystem, 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 Data Rate Adjuster 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 Data Rate Adjuster. If the FIFO is less than one quarter full, the first sample 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, the last available sample is duplicated and transmitted to the D/A converter until new data arrives.

If the FIFO is more than three quarters full, the Data Rate Adjuster deletes the first sample 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, samples are discarded until space in the FIFO exists for a new left/right channel sample pair. In practice, this situation should not occur.

Initialization and control of the audio subsystem is performed by the local CPU or the NeXT host via the PBUS and the I<sup>2</sup>C Bus. The transmit and receive FIFO flags are programmed via the PBUS, and the transmitter and receiver can be independently turned on and off via a PBUS port. The audio tone and volume settings are adjusted via the I<sup>2</sup>C Bus.

### C. The High-Speed Radiological Image Subsystem

A block diagram of the mini-ATMizer transmitter that creates the ATM radiological image data stream is shown in Fig. 5 [12]. This transmitter accepts a radiological image data stream from a dedicated image server using a TAXIchip receiver operating at 40 Mb/s. The incoming data stream is buffered using a FIFO. This FIFO's

programmable full flag is programmed by a resident finite-state machine at power-up to assert when 48 or more bytes are contained in the FIFO. This ensures that a complete 48-byte data payload is available when transmission of a non-null cell begins. If this check were not performed, the 100 Mb/s transmitter could empty the FIFO during the transmission of an ATM cell. To ensure that the complete image is transmitted, the image server supplies an image that is a multiple of 48 bytes in length by padding with null data. The Header Stuffing block provides an appropriate header, and outbound cells are transmitted using a TAXIchip transmitter operating at 100 Mb/s. Transmission of the header, data payload, and null cells is performed under control of the Control block. Currently, all data cells are transmitted using a single, fixed header. This does not cause difficulty, however, since only one VCI/VPI is ever used on this ATM link.

The high-speed radiological image subsystem is shown in Fig. 4. This subsystem receives a radiological 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 [12].

## IV. SOFTWARE

### A. Overview

The software for the MMS consists of embedded software running on the ATMizer and applications and control programs running on the NeXT host. The Mach Operating System's loadable kernel server feature [19] was used to establish a communication channel between the NeXT host and the ATMizer. This allowed the address space on the ATMizer to be mapped into the NeXT's kernel map through the NBIC, thus creating a shared memory interface between the NeXT and the ATMizer. A pair of message queues was implemented in this shared memory to allow communication between the NeXT host and the ATMizer.

### B. Embedded Software

Currently, the ATMizer has two major embedded software components. The first is a memory manager implemented to manage the four megabytes of on-board DRAM. The memory manager uses the message queue pair to allow the NeXT to allocate or free a block of memory on the ATMizer. The second component is an IP

implementation for the MMS which uses the TxFIFO and the RxFIFO to run IP over the ATM network. The MC68030 is used to handle segmentation and reassembly of the IP packets. The software implements the full AAL5 protocol [20] with only one exception: a 16-bit CRC (Cyclic Redundancy Check) is used instead of the standard 32-bit CRC. The initial implementation of the IP software is quite slow because of operating system overhead and software calculation of the CRC. Improvement in performance is expected in the next version of the system.

### C. MMS Host Software

The software running on the NeXT consists of two loadable kernel servers for control and application software to demonstrate the capabilities of the hardware. The two kernel servers provide the NeXT host with access to the ATMizer hardware. One kernel server provides I<sup>2</sup>C Bus communication for control of the video components. The other server is the ATMizer IP kernel server. This server provides hooks into the IP protocol layer in the NeXT's operating system kernel. This enables the ATMizer to appear to the NeXT as an additional network device so that IP packets may be sent over the ATM network using the AAL5 software described above. If the NeXT were configured as a dual homed host, it could also be used to route packets/cells between Ethernet and the ATM network.

The current applications for the MMS demonstrate the capability of the hardware to support a multimedia physician's workstation. One application, called Video Exchange, provides a multicast video and conference utility. With this program, physicians could call other doctors and communicate with full video-rate, real-time conferencing via the video and audio subsystems. The Radiological Image Viewing Application (RIVA) [21,22] allows doctors to retrieve medical images via the high-speed radiological image subsystem in addition to normal, written reports. Video-taped exams and video reports stored on laser disk can also be retrieved via the video subsystem. This allows physicians at remote locations to conference to discuss patients, and, in particular, images and exams. A third application, ECG Display, is capable of transmitting a real-time ECG (Electrocardiogram) signal over the ATM network via the CPU channel of the ATMizer. This demonstrates the possibility for physicians to observe certain tests, e.g., a treadmill test, without having to travel to the hospital. The physician could view the patient on the treadmill from his office or other remote location using Video Exchange while monitoring the patient via the ECG Display application.

Currently, the Video Exchange, RIVA, and ECG Display applications do not take advantage of the



MC68030 processor on the ATMizer. These applications use the ATMizer for control of the multimedia subsystems and as a network interface for these subsystems. For this low-bandwidth interaction, the local CPU is bypassed and configuration handled directly by the NeXT host.

## V. DISCUSSION

While the MMS uses a large number of fairly complex integrated circuits, all of the components used in the construction of the MMS are commercially available. Physically, each MMS consists of a NeXT computer with two 27.5 cm by 27.0 cm plug-in cards. One of these cards contains the ATMizer subsystem, and the other contains the video subsystem and high-resolution radiology image subsystem. These cards, which are connected by a PBUS/I<sup>2</sup>C Bus

connector, physically plug into the NeXTBus. A third 10.2 cm by 25.4 cm card, which connects to only the PBUS/I<sup>2</sup>C Bus connector, contains the audio subsystem. The MMS has been operational since mid-1992. Six of the systems are currently deployed in the Washington University broadband ATM network and used for demonstration purposes.

In November of 1992, three MMSs were used to demonstrate the future of teleradiology via an infoRAD demonstration at the Radiological Society of North America annual meeting in Chicago, Illinois. This demonstration, which was based on the RIVA and Video Exchange applications, featured audio/video teleconferencing between individuals demonstrating how physicians at remote sites might use the system and included high-speed radiological image service, video clip service via laser disks, and remote database access.

Future plans at Washington University include replacement of the prototype switches in the broadband network with switches from a commercial switch vendor. These switches, which operate at 155 Mb/s, are based on the architecture of the prototype Washington University ATM switch and use the protocol defined by the ATM Forum [15]. The MMS is currently being upgraded to operate at 155 Mb/s using the new protocol.

## VI. CONCLUSIONS

The Washington University MultiMedia System is a complete multimedia system capable of transmitting and receiving video, audio, and radiological images, in addition to normal network traffic, over the Washington University broadband ATM network. The Multimedia System 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 Dr. G. James Blaine, Ian Flannigan, Pete Flugstad, Jason Fritz, Craig Horn, Steve Moore, Kenichi Sato, Bob Whitman, and David Zar for their contributions to the effort described here. In addition, the authors want to thank the original Fast Packet Project sponsors: Southwestern Bell Telephone Company, Southwestern Bell Technology Resources Incorporated, and NEC America, and the current Project Zeus Sponsors: Ascom Timeplex, SynOptics, Southwestern Bell Technology Resources Incorporated, and Bellcore.

## REFERENCES

- [1] J.R. Cox, Jr., and J.S. Turner, "Project Zeus: Design of a Broadband Network and its Application on a University Campus," WUCS-91-45, Department of Computer Science, Washington University, St. Louis, MO, 1991.
- [2] J.S. Turner, "Design of a broadcast packet switching network," *IEEE Trans. Commun.*, vol. COM 36, no. 6, pp. 734-743, June 1988.
- [3] H. Casabona, R. West, K. O'Day, H. Ross, C. Ross, and S. Augarten, *NeXT User's Reference*, NeXT, Inc., Product Number N6002, 1990.
- [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] C. Mikkelsen, *NeXTBus Interface Chip Specification*, NeXT, Inc., Product Number N6011, 1990.
- [7] *The Video Data Handbook*, Signetics Company, 1991.
- [8] K.B. Benson, ed., *Television Engineering Handbook*. New York: McGraw Hill, 1986.
- [9] G.K. Wallace, "The JPEG still picture compression standard," *Commun. ACM*, vol. 34, no. 4, pp. 31-44, Apr. 1991.
- [10] 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.
- [11] 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.
- [12] 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.
- [13] B.W. Kernigan and D.M. Ritchie, *The C Programming Language*, Englewood Cliffs, NJ: Prentice-Hall, 1988.
- [14] *TAXIchip Integrated Circuits Technical Manual*, Advanced Micro Devices, 1989.
- [15] *ATM User-Network Specification Version 2.0*, The ATM Forum, June 1, 1992.
- [16] *JPEG Chipset Technical Manual*, LSI Logic Corporation, December, 1991.
- [17] *MOS Memory Data Book*, Texas Instruments, Inc., 1991.
- [18] *SMART Analog Data Book Volume 2: Communications ICs*, Crystal Semiconductor Corporation, 1990.
- [19] K. Walrath, *Writing Loadable Kernel Servers*, NeXT, Inc., Product Number N6007B, 1990.

- [20] ANSI T1S1.5, "AAL 5 - A new High-Speed Data Transfer AAL," Nov. 1991.
- [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.

#### CAPTIONS

Fig. 1. The ATMizer subsystem.

Fig. 2. The video subsystem.

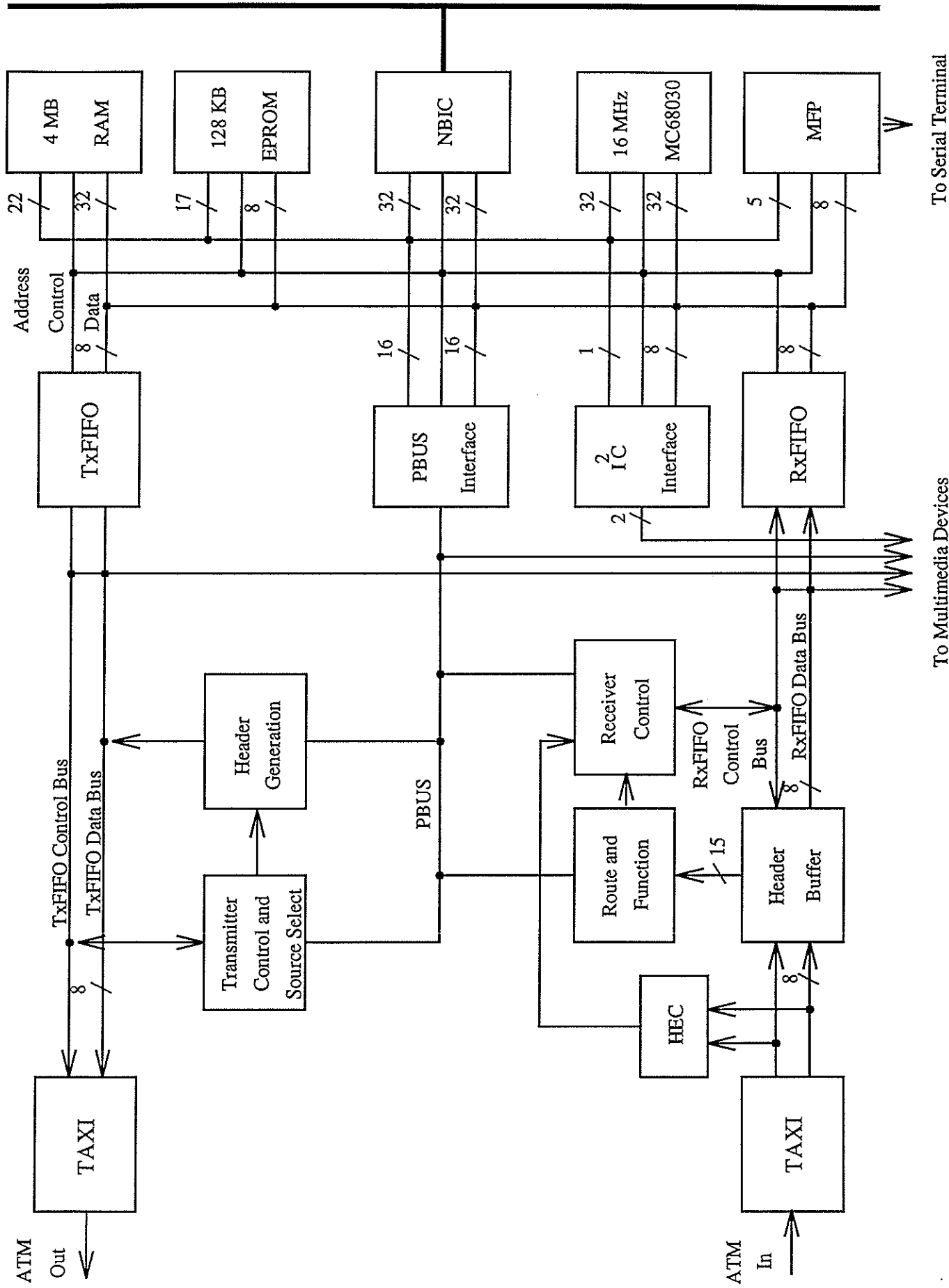
Fig. 3. The audio subsystem.

Fig. 4. The high-speed radiological image subsystem.

Fig. 5. The mini-ATMizer transmitter.

Fig. 6. The JPEG Engine.

NeXTBus



To Multimedia Devices

To Serial Terminal

Fig. 1. The ATMizer subsystem.

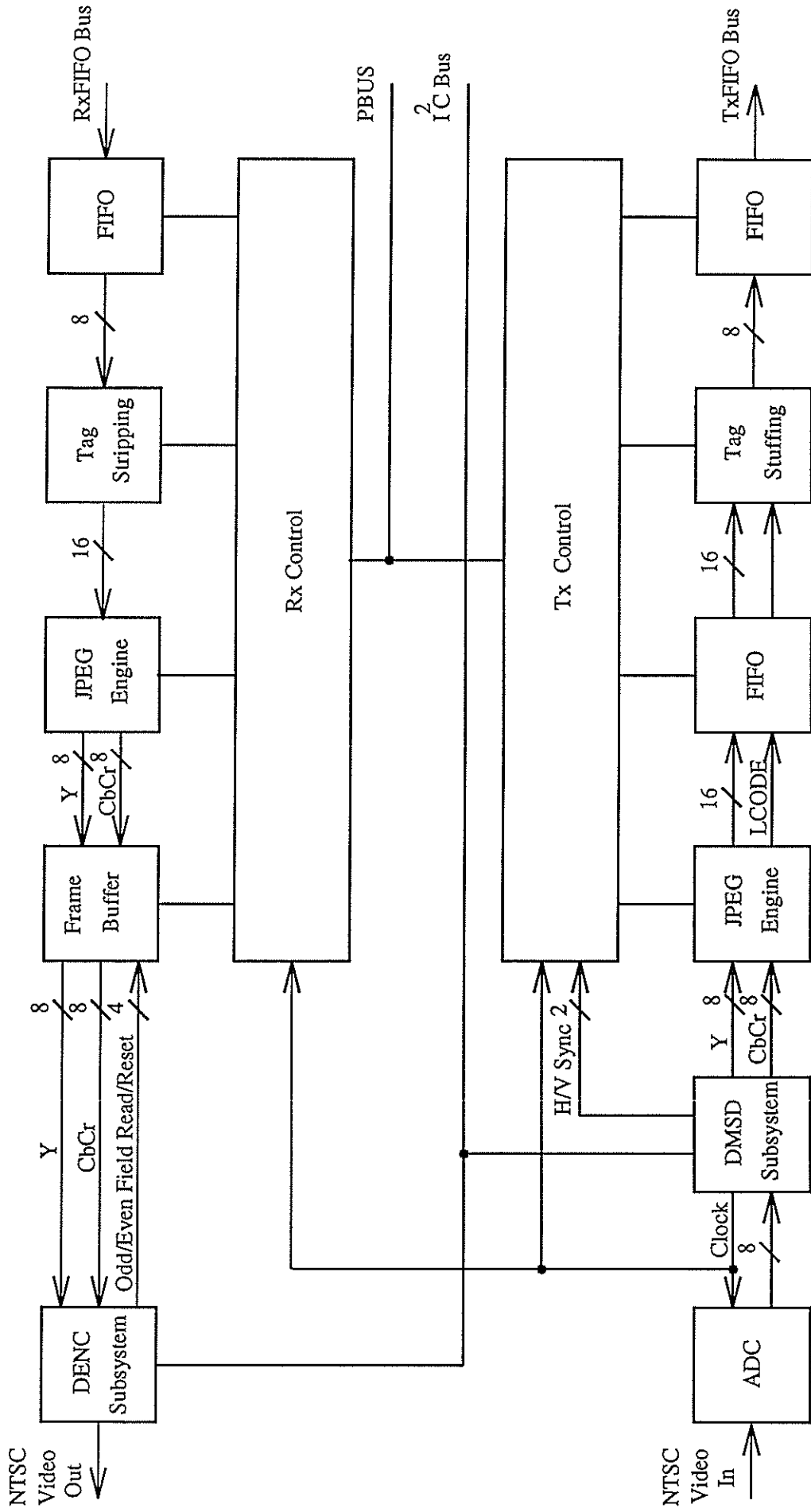


Fig. 2. The video subsystem.

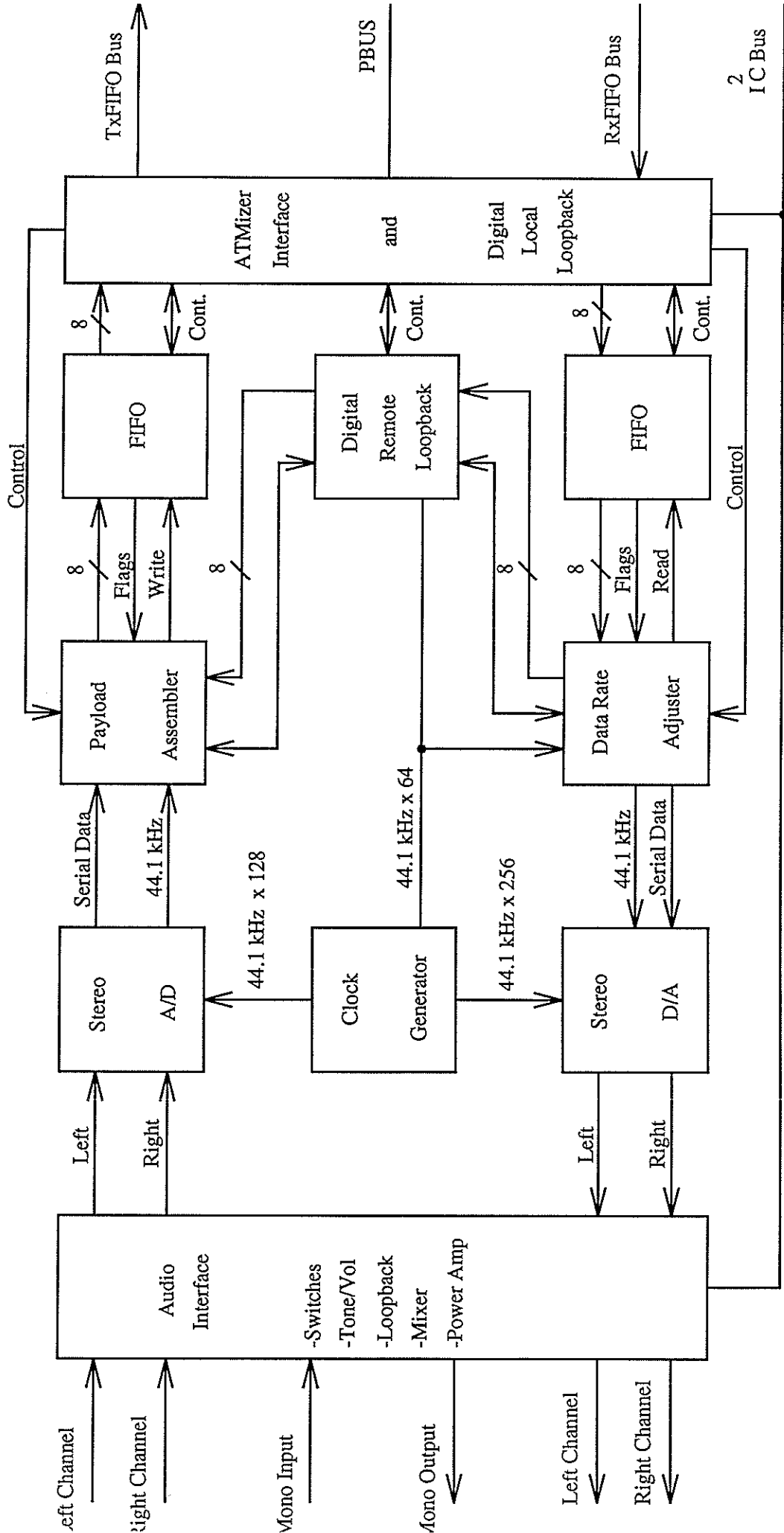


Fig. 3. The audio subsystem.

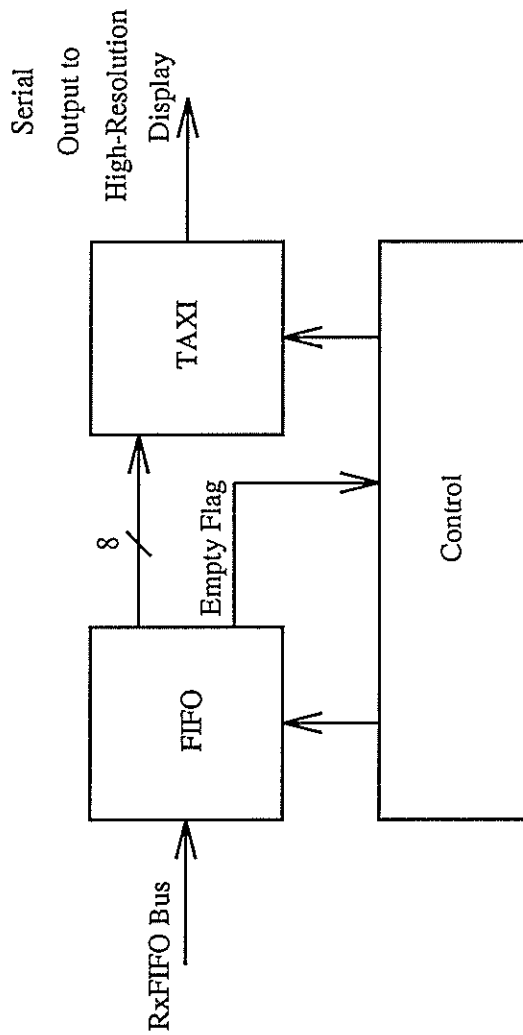


Fig. 4. The high-speed radiological image subsystem.



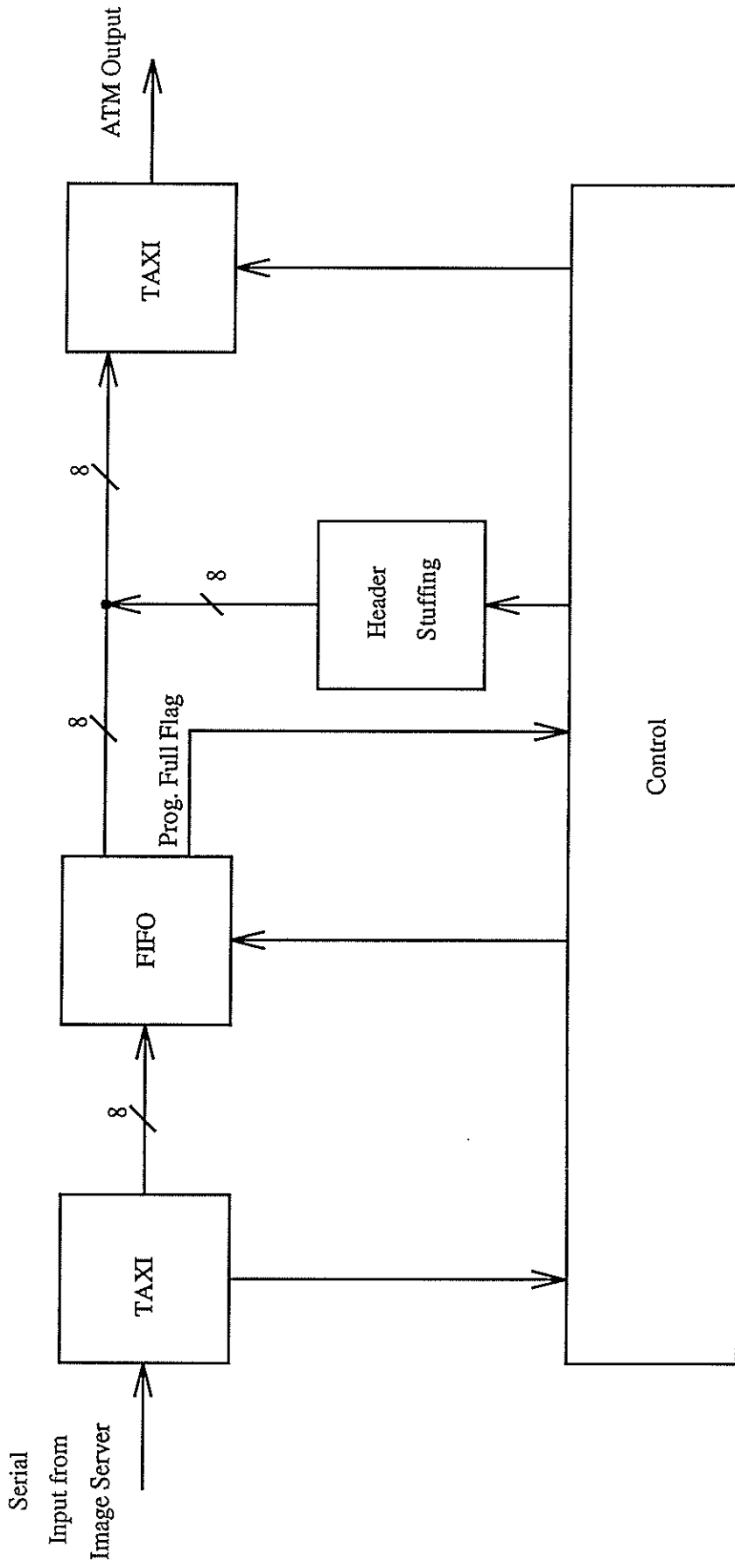


Fig. 5. The mini-ATMizer transmitter.

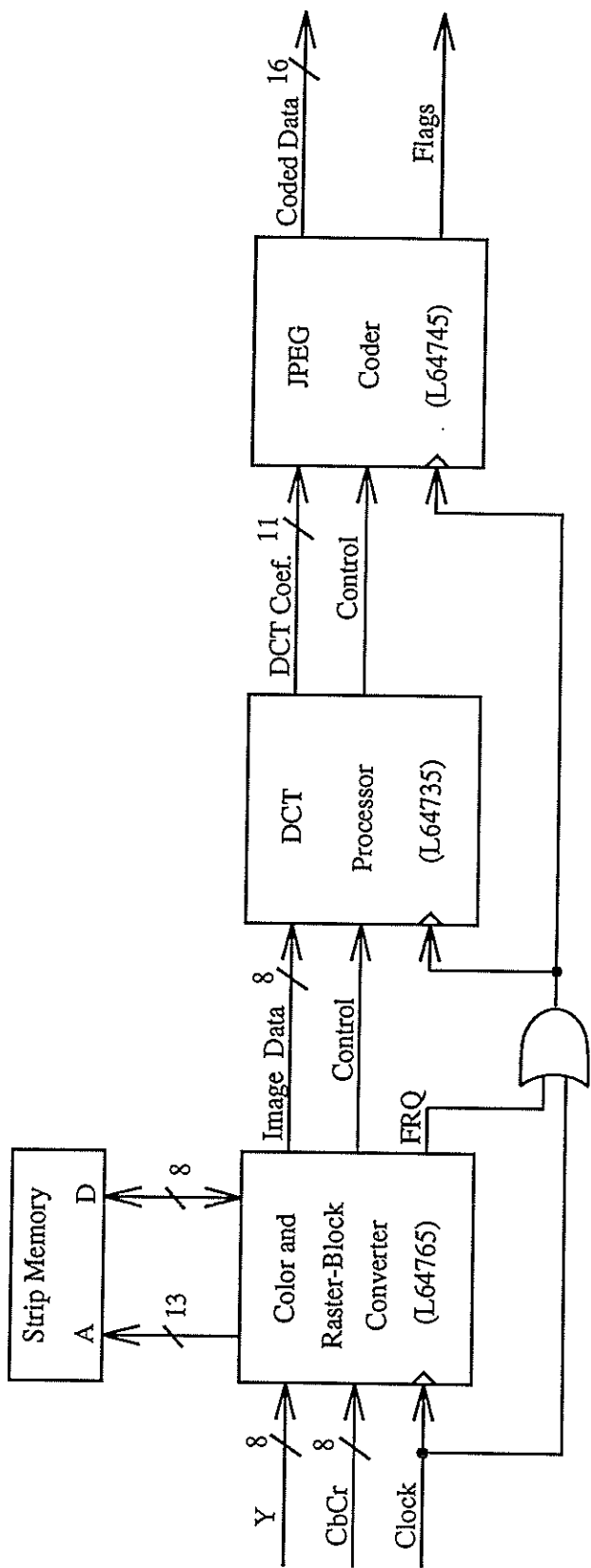


Fig. 6. The JPEG Engine.