TECHNICAL RESEARCH REPORT

ATM/LAN Access Switch (ALAX): System Architecture

by A. Makowski, P. Narayan, J. Kim, M.G. Ryu,
T. Christofili, G. Charleston, S. Rao

CSHCN T.R. 95-14
(ISR T.R. 95-111)

The Center for Satellite and Hybrid Communication Networks is a NASA-sponsored Commercial Space Center also supported by the Department of Defense (DOD), industry, the State of Maryland, the University of Maryland and the Institute for Systems Research. This document is a technical report in the CSHCN series originating at the University of Maryland.

Web site http://www.isr.umd.edu/CSHCN/

ATM LAN Access Switch (ALAX) :System Architecture

Man Geun Ryu ; *mgryu@src.umd.edu, modacom@soback.hana.nm.kr*
Modacom Co. Ltd., Korea

Jangkyung Kim ; *jkkim@src.umd.edu, jkkim@pec.etri.re.kr*
Protocol Engineering Center
Electronics and Telecommunications Research Institute, Korea

Release 1.0
June 1995

Laboratory
for
Advanced Switching Technologies

ISR, University of Maryland
College Park, MD 20742

Project team members:

Dr. Armand Makowski	ISR, UMCP
Dr. Prakash Narayan	ISR, UMCP
Dr. Jangkyung Kim	ETRI, Korea
Mr. Man-Geun Ryu	Modacom Co. Ltd., Korea
Mr. Thomas Christofili	ISR, UMCP
Mr. Giles Charleston	ISR, UMCP
Mr. Sandeep Rao	ISR, UMCP

This document contains the hardware information for the ATM LAN Access Switch (ALAX) that was done as a collaborative research and development effort by the Institute for Systems Research (ISR) in University of Maryland at College Park (UMCP), Protocol Engineering Center (PEC) in Electronics and Telecommunications Research Institute (ETRI), and Modacom Co. Ltd. of Korea.

1. Overview to ATM LAN Access Switch (ALAX)

Figure 1 illustrates the whole system architecture of the ALAX. The key point is that internal main bus type of the ALAX uses the P1355 interface to be proposed as the IEEE standard as shown in the figure. IEEE P1355 is new serial bus to provide the data interconnect between systems as stated in the title 『Standard for Heterogeneous InterConnect (HIC)』. The purpose of this standard is to enable high-performance, scalable, modular, parallel systems to be constructed with low system integration cost; to support communications system fabric; to provide a transparent implementation of a range of high level protocols; and to support links between heterogeneous systems.[1]

According to the first plan, the primary target of design is LAN Emulation Over ATM i.e. especially Ethernet LAN. There is no interrogation that Ethernet have the worldwide users of the LAN today. The first version of the ALAX's general hardware is categorized in three parts as follows:

- ATM to IEEE P1355 Interface,
- IEEE P1355 Virtual Switch Matrix, and
- Ethernet to IEEE P1355 Interface.

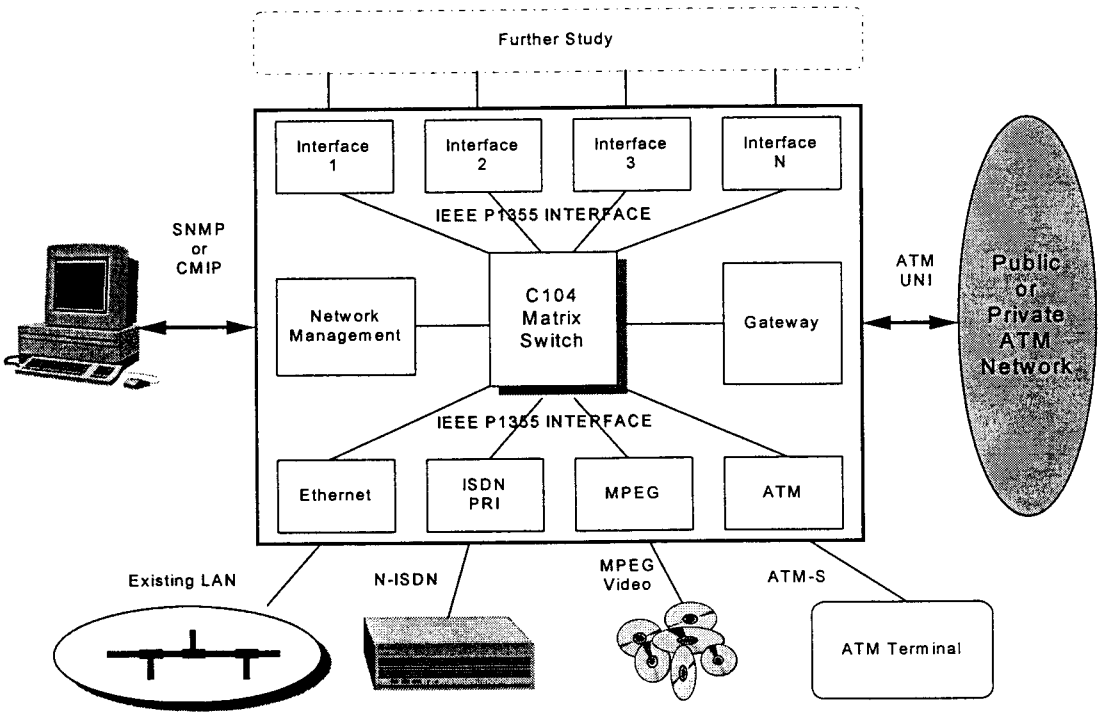


Figure 1. Whole System Architecture of the ALAX

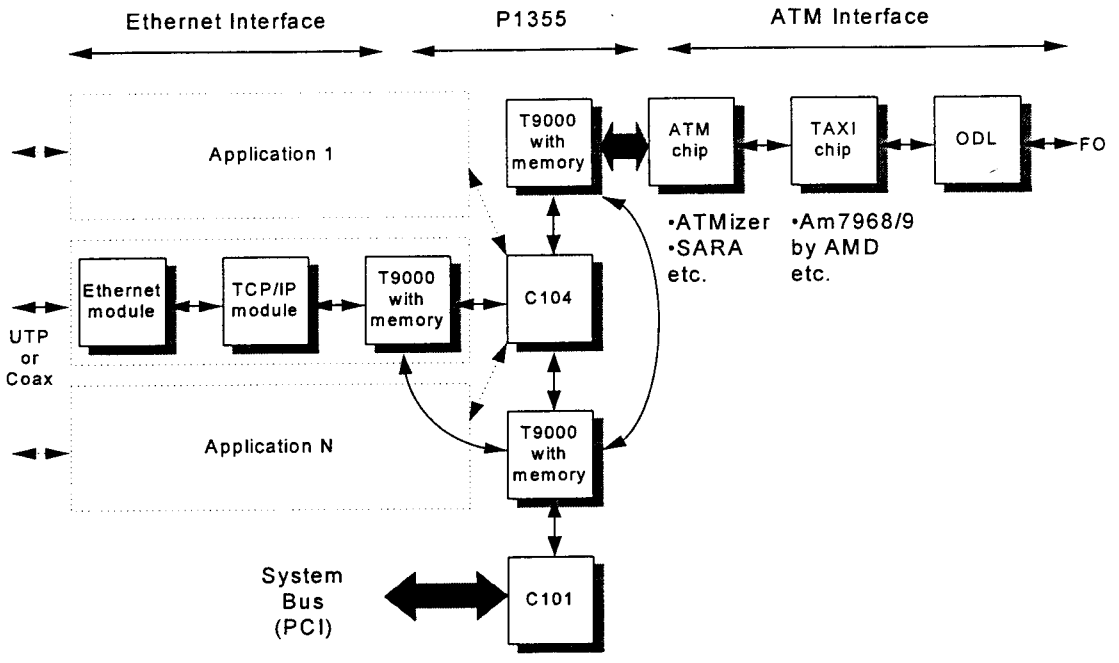


Figure 2. Entire Block Diagram

Figure 2 shows the block diagram for the object. Currently, the first plan has the following conditions;

- ATM User Network Interface applies UNI 3.0 specification as defined in ATM forum.
- ATM Physical Layer applies TAXI for the 100 Mbps multimode fiber.
- Transputer provides full support for parallel processing.
- Data bus applies IEEE P1355.
- Network management and control/monitoring applies PCI bus as defined in PCI Special Interest Group.
- Segmentation and Reassembly is applied by AAL5 and performed by an RISC based CPU.
- Primarily, LAN interface applies Ethernet.

In accordance with above conditions, these researched and developed modules will be inserted into a Pentium PC, where each modules will be inserted into its Peripheral Component Interconnect (PCI) bus. So far as it is designed, the ALAX provides LAN interconnect for up to 15 Ethernet segments because it uses only an ST C104. It is easy to increase the expansion of interconnection without changing of entire architecture, because of the scalable, modular and parallel structure of IEEE P1355 serial bus. Also, in line with the filed application, the system has the hardware architecture which is able to smoothly interconnect and administer the applications such as MPEG 2 over ATM, Multiprotocol over ATM, and Circuit Emulation etc.

2. ATM to IEEE P1355 Interface Module

This module has the capability to convert P1355 data through Virtual Switch Matrix to and from ATM UNI. The following Figure 3 shows the block diagram for the module. As shown in the figure 3, entire block consists of many important chipsets. The following describes each blocks in detail.

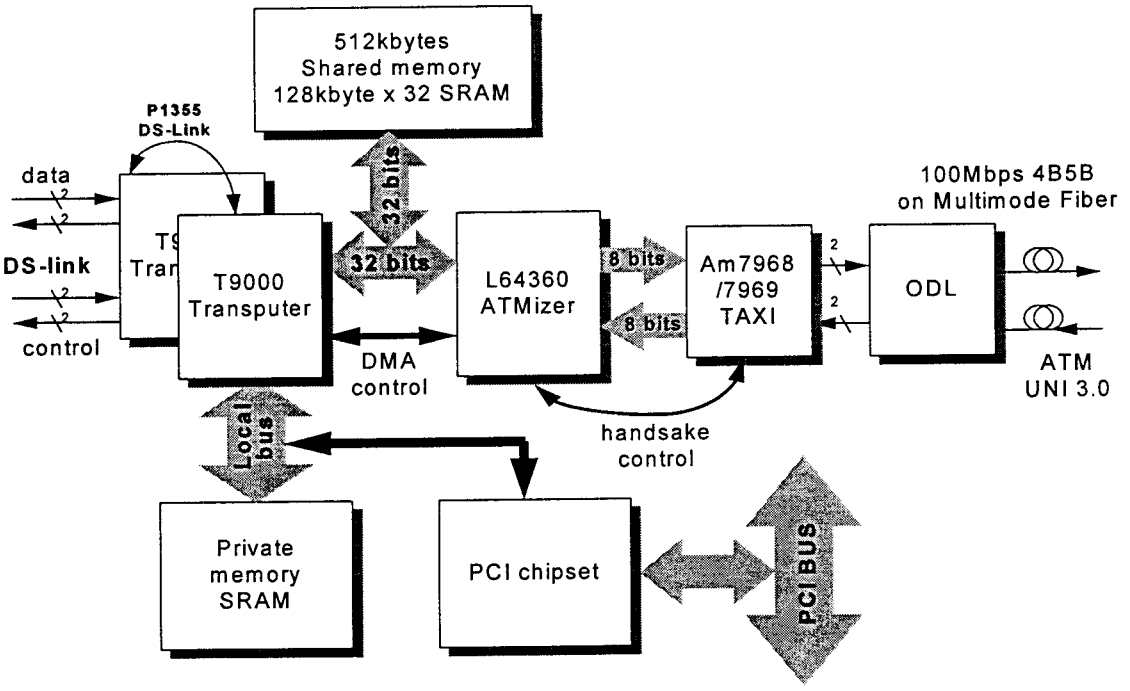


Figure 3. ATM to IEEE P1355 Interface Block Diagram

There are three types of links as defined in IEEE P1355 standard: DS: Data/Strobe encoding, TS: Three-of-six encoding, and HS: High Speed encoding. ALAX uses the DS-link that are intended for use for chip-to-chip and board-to-board interconnection. DS-link connection within the T9000 transputer has two types; the data path and the control path. Both the links connect the Switch Matrix Module. The data path goes from the ATMizer through the shared memory, and the control path uses the control and monitoring for T9000. When the Network Manager requests some data to T9000, it reports upon the result of requested data via the control path. In the emerging field, the communication systems requested the high processing speed, high transmission speed and high speed virtual networking to make the information age come true. Asynchronous Transfer Mode (ATM) is key technology in the dominion. The ALAX also has to be designed with due regard to it. In the presently, a major problem is the bandwidth and flexibility of the bus architecture. Sharing bandwidth over a bus such as the existing bus type is no longer a satisfactory solution in today's applications. IEEE P1355 is a serial interconnect which provides designers an matchless opportunity to break the communications bottleneck, offers an integrated solution to a variety of ATM requirements. Refer to IEEE

P1355 standard document for details. Eventually, a successful guarantee like this system must utilize high performance interconnect and parallel processing processors to support the technology. T9000 transputers are selected because the transputer is able to interconnect to IEEE P1355, and has a pipelined superscalar architecture which allows multiple instructions to be executed every processor cycle.

The main functional blocks of the T9000 are shown in Figure 4. The requirement for processing performance in embedded systems such as the ATM is continuously increasing as control algorithms become more sophisticated and as systems become more complex. In the long term, the only solution to these ever increasing demands for performance is the use of multiple processors to perform independent co-operating system functions. Transputers are the only microprocessor specifically designed to tackle the problems of building multiprocessor systems. There are advantages other than just performance to using multiple transputers in a system. It allows ease of system partitioning, permits scalable systems to be built, where more processor can be added as demand increases, or provides the optimum balance of price versus performance. The ALAX also needs high performance and complex algorithms, it means that the architecture of the system is required to use multiple transputers.

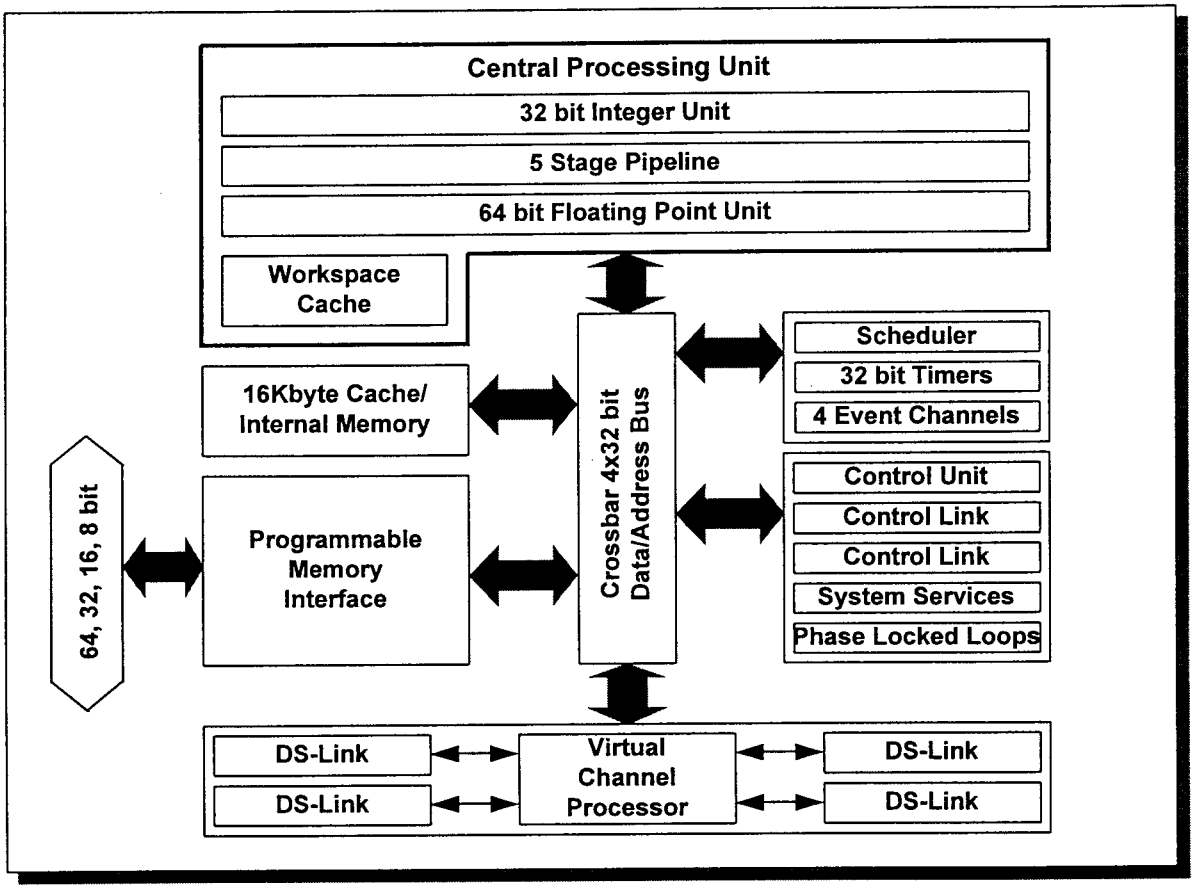


Figure 4. Transputer T9000 block diagram

The key functions of T9000 transputer are the higher layer functions including multiprotocol process, Common Part Convergence Sublayer function in ATM Adaptation layer, signaling function, and local addressing function as defined in ATM UNI 3.0. The following Figure 5 summarizes the processing of data at AAL sublayers. the AAL can be divided into SAR sublayer and CS. That is, the process of converting user-service data units (U-SDU) into ATM cells executed by the AAL is divided into two sublayers. T9000 handles U-SDU that is exchanges the formatted data as specified in IEEE P1355 through the DS-link from Virtual Switch Matrix. The SAR provides the functions associated with the segmentation and reassembly of U-SDUs, and the CS provides the capability for converging specified service-related functions to an upper service layer. The CS accepts U-SDUs from an upper user layer to which it adds a header and trailer related to error handling and data priority preservation to create the SAR-PDUs, which are then sent to the ATM layer. This process also has be in charge of the T9000. After processing CPCS, T9000 makes CS-PDU, and stores it to shared memory. In the opposite direction, the SAR sublayer analyzes the SAR-PDUs transported from the ATM layer, and the SAR-PDUs are collected and assembled together with the CS-PDUs and delivered to the CS. Then CS, action of T9000, analyzes the header and trailer of the transported CS-PDUs and extracts just the U-SDUs, finally delivering them to the upper user layer. Inter entity protocols such as flow control are also handled at the CS. There is no processing the header in ALAX because of using AAL 5 at the first plan.

Also, handling work of T9000 is the processing of Local Addressing concept, because ALAX provides various services and protocols. It will be called "ATM Mapping Layer (AML)"or "MAC Mapping Layer(MML)" in this document. The ALAX is a small virtual switch which needs the local addresses handling through AML.

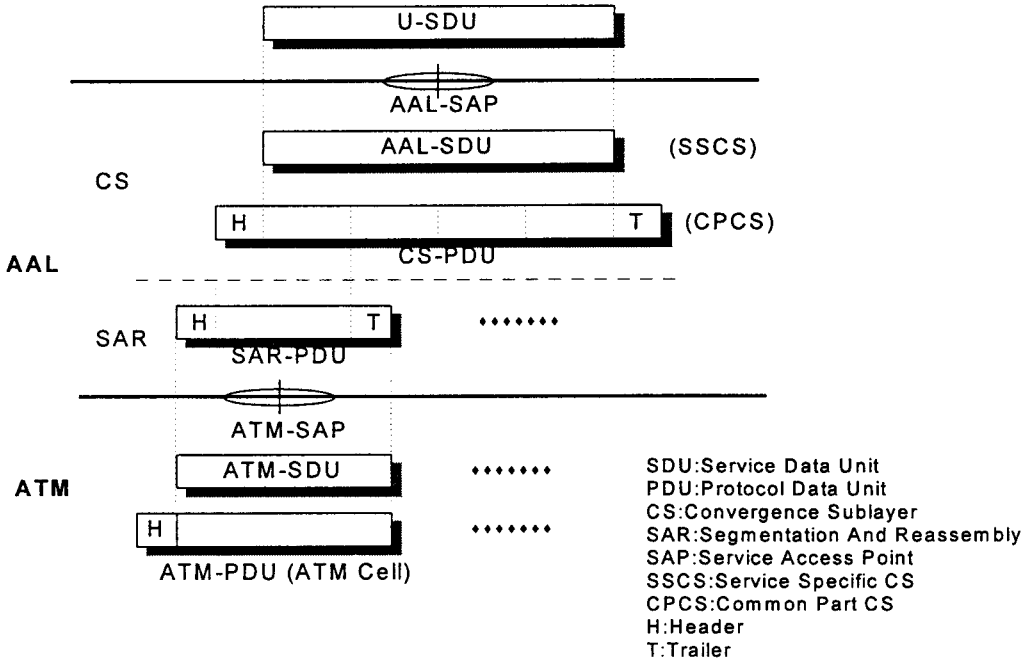


Figure 5. Generic AAL protocol sublayer model

As described on the previous page, the T9000 stores finished CS-PDU of CS processing in shared memory. The reason why there is used the "shared" that the memory is used both T9000 and ATMizer. Tentatively, the size of shared memory is decided as 512 kbytes (128k x 32); it can be changed in conformity to the result of the OPNET simulation. The Direct Memory Access Controller (DMAC) within the ATMizer architecture handles memory transactions between the Virtual Channel RAM (VCR) and shared memory. The DMAC retrieves SAR user payloads from CS-PDUs in shared memory during segmentation operations and writes SAR user payloads back into CS-PDUs during reassembly operations.

Figure 6 shows the block diagram of the ATMizer architecture. The ATMizer from LSI logic corporation is a single-chip segmentation and reassembly ATM network controller. The ATM application interface in "Project Zeus" at the Washington University also uses the ATMizer.

The architecture includes the ATM Processing Unit (APU), a 32 bit RISC CPU, which controls the functional blocks via user firmware. The ATMizer architecture provides the Broadband-Integrated Service Digital Network (B-ISDN) layers as showed below;

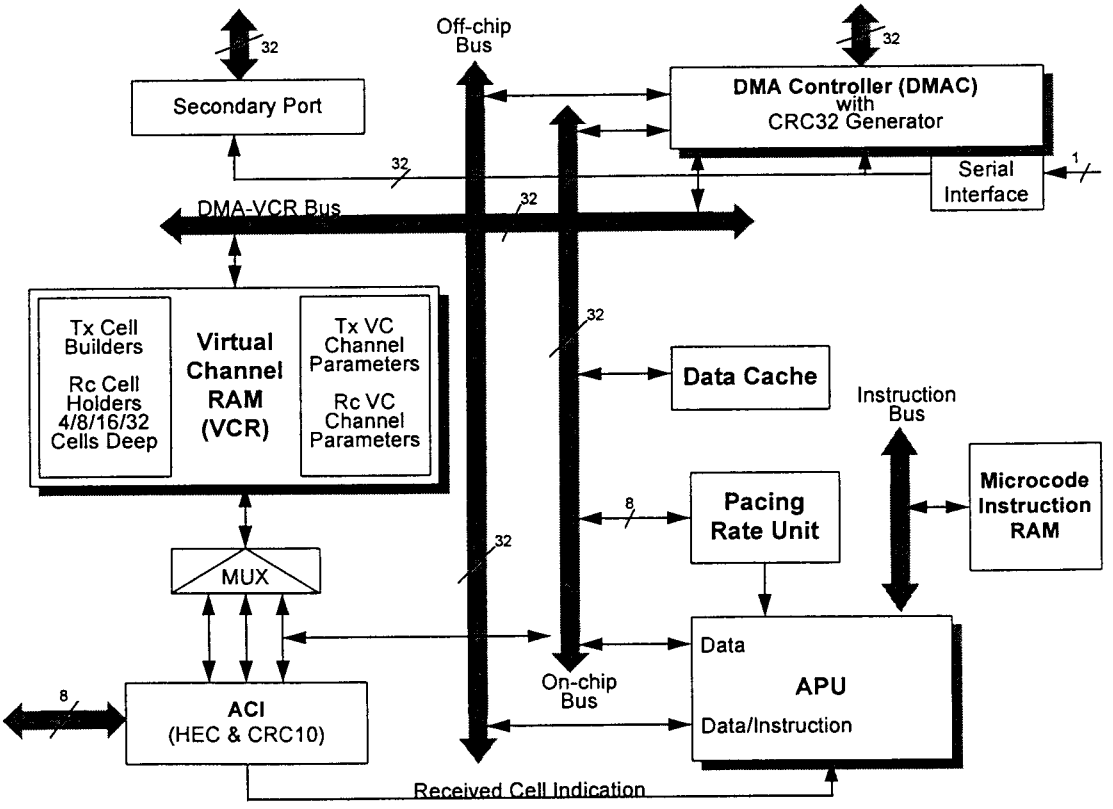


Figure 6. ATMizer Block Diagram

ATM Adaptation Layer

- AAL5 CRC32 Generation and Checking
- Segmentation and Reassembly Sublayer
  - ✓ Type 1, CBR
  - ✓ Type 2, 3/4 VBR
  - ✓ Type 5, VBR

ATM Layer

- Cell Multiplexing and Demultiplexing
- Generic Flow Control
- Cell Header Generation and Extraction
- Cell Rate Pacing
- Cell TX Exceeding the Pacing Rate
- Delay Priority Processing
- CLP Marking, CLP Reduction
- Explicit FC Indication to Higher Layers
- Cell Payload Type Marking, Differentiations
- Cell Relaying
- Cell VPI/VCI Translation
- Peak Rate Enforcement

Physical Layer

- HEC Generation
- Cell Delineation
- Cell Rate Decoupling

The APU within the ATMizer, based on MIPS R3000 architecture, enables customers to immediately realize ATM termination in a single device, and subsequently update signaling, congestion and traffic management algorithms in firmware as the ATM standard evolves. By executing real-time instructions from an internal instruction RAM, the intelligent APU also enables ATM systems to respond immediately to network congestion without T9000 transputer intervention.

Figure 7 illustrates the general loop of firmware within the ATMizer, which verifies for the existence of one of three conditions and takes the appropriate actions if one of the states exists. The LSI logic recommends the flow charts as an example. The firmware checks for Received Cell Indication first since it is more important not to drop a received cell than it is to prevent an Idle Cell from being transmitted. The user firmware should always check for Received Cell Indication before checking for either a Transputer Messaging Request or a Transmit Cell Request. Cells may accumulate in the VCR and firmware may wish to always drain this buffer before proceeding with the segmentation routine, because the ATMizer can be asked to achieve certain complicated functions, the servicing of either a Received Cell Indication, transmit cell request, or message request could take longer than the duration commonly dispensed in steady state operation.

Figure 8 illustrates a typical reassembly routine triggered by the interrupt or condition signals. ATM Cell Interface (ACI) extracts and checks the HEC Field and puts the cell into the VCR, when a cell arrives through the TAXI chipset. Since there is no way of detecting the AAL type yet, the CRC10 circuitry must check the CRC10 Field of every incoming cell regardless of the AAL type. Once the cell is buffered, the ACI generates an interrupt to notify the APU that there is at least one new cell in the buffer. Alternately, the APU can poll a condition signal to find out if there is at least one new cell in the buffer.

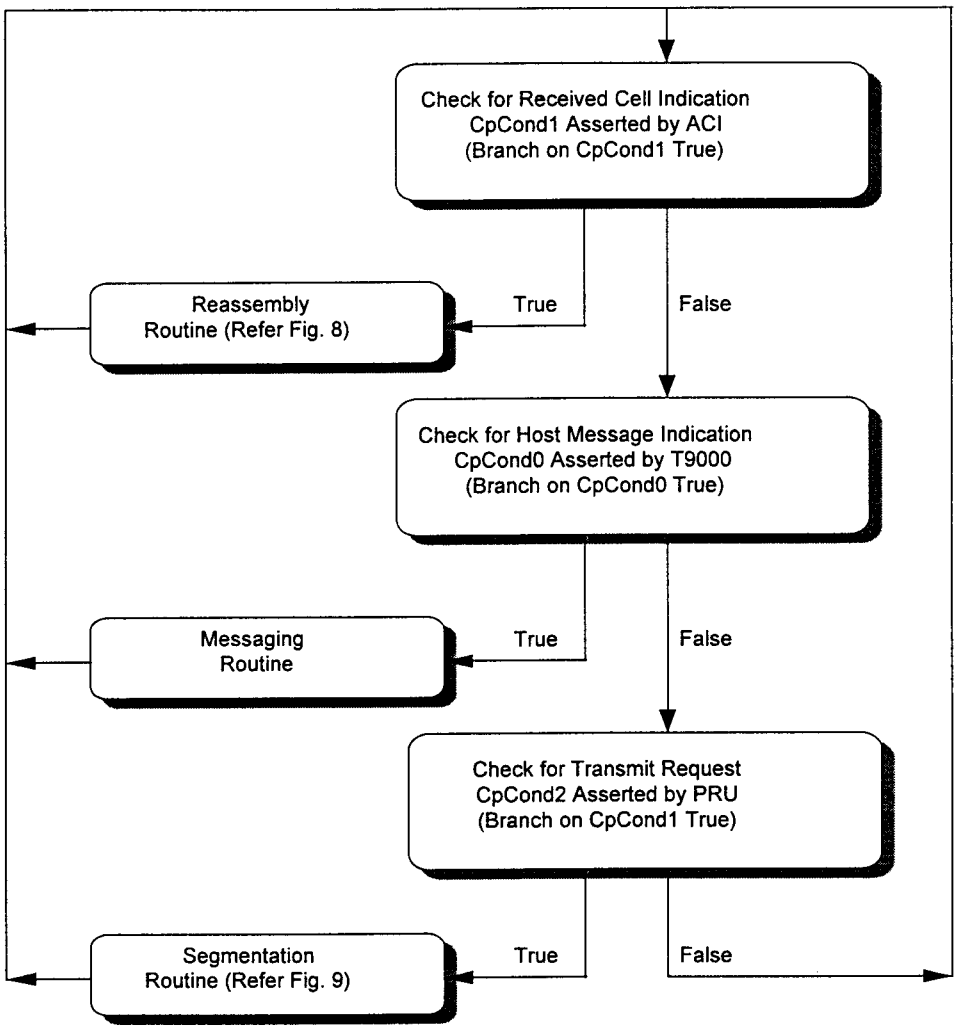


Figure 7. General Loop in Firmware

Firmware can implement an interrupt mechanism using the Received Cell Indicator Register (RCIR) as an up or down counter. Each time a new cell arrives, the counter is incremented by one. If the counter is greater than zero, the ACI asserts interrupts and coprocessor condition signals. Each time the APU processes a cell, at the end of the reassembly routine the APU must write to the PCIR to decrement the counter by one. The first step in processing a cell is to extract the cell header and find the VCI and VPI fields. The VCI/VPI are used to index a data structure which describes the cell.

Once the data structure is retrieved, the APU can find out more information about how to proceed with processing the cell. The cell's data structure (CPE) may contain information such as the AAL type, the T9000 transputer memory address (where the CS-PDU resides), the CRC32 Partial Value for AAL5, and the previous SAR Header for AAL3/4. Next, the CPU may decide whether to translate the VCI/VPI and switch the cell back for transmission or to terminate the cell. In the case of termination, the APU programs the

DMAC with the T9000 memory address found in the data structure. In the case of AAL5, the APU initializes the CRC32 generation circuitry with the CRC32 Partial Value found in the data structure.

For AAL5, it is necessary to have partial CRC32 value because the CRC32 mechanism is calculated over the whole CS-PDU. Once the SAR-SDU is transferred into the shared memory, the CPU retrieves the new CRC32 Partial Value and updates the data structure. If this is the last SAR-SDU for AAL5, the CRC32 value can be checked against the last four bytes of the payload. In the case of an error, the APU informs the T9000 by some messaging protocol. If it is not the last SAR-SDU for the particular CS-PDU, the APU also updates the T9000 memory address for the next SAR-SDU transfer.

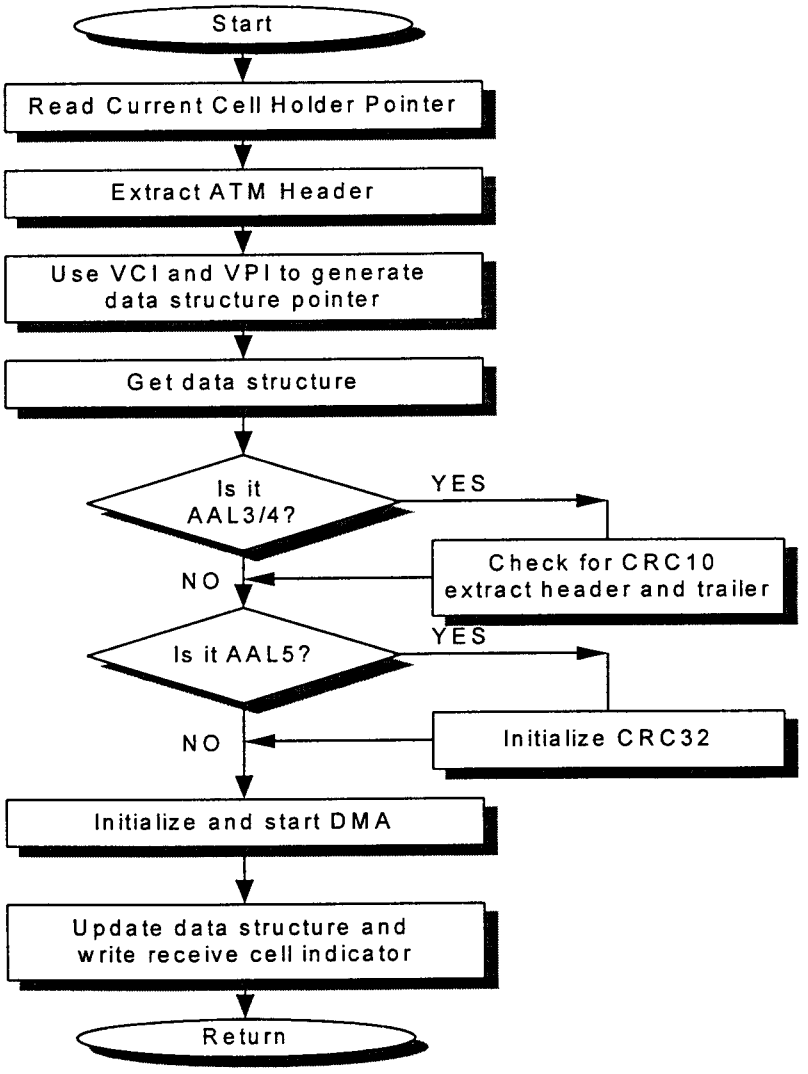


Figure 8. Reassembly Routine

Figure 9 illustrates a typical segmentation routine triggered by the peak rate counters. The event that triggers a segmentation process can be generated when one or more of the Peak Rate Pacing Counters (PRPC) elapses or by a predefined messaging system with the T9000 processor. The PRPC can be used to index a link list of pointers. These pointers point to a data structure describing the particular CS-PDU to be segmented. When one or more counters elapse, the APU can be interrupted and the firmware should vector into the segmentation routine. The first task for the segmentation routine is to use the counter and index a particular link list. The APU can implement a priority mechanism to select which CS-PDU will be segmented first.

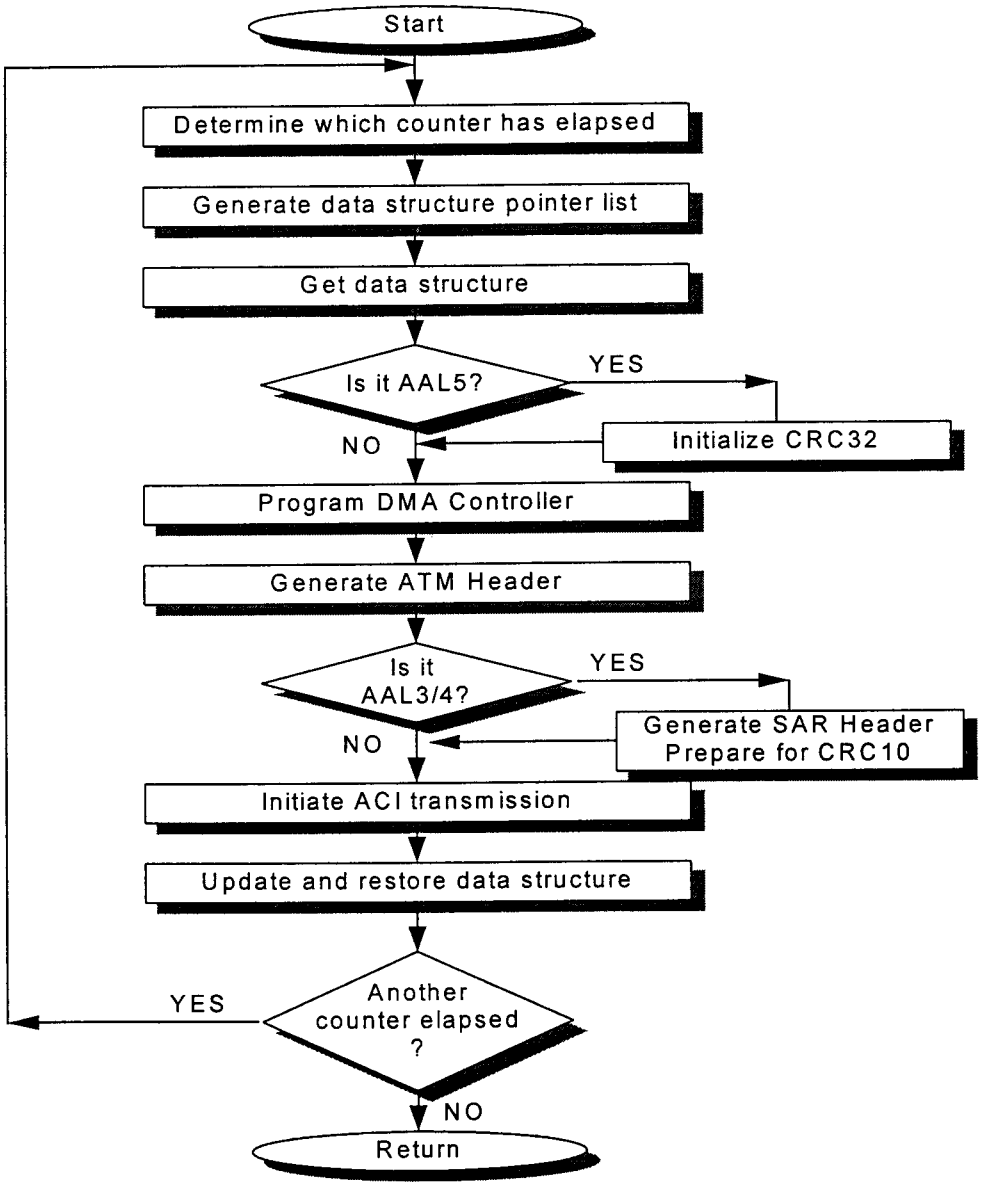


Figure 9. Segmentation Routine

The data structure to describe a CS-PDU for segmentation process is slightly different compared to the data structure for the reassembly process described in the previous page. The data structure for segmentation contains information such as the ATM Header, the CRC32 Partial Value for AAL5, the current T9000 memory address pointing to the SAR-SDU to be transferred, and the transfer count. Other useful control information such as the maximum burst size and priority can be included in the data structure. The maximum burst size can be used to inform the APU to transfer multiple SAR-SDUs from the shared memory into the cell buffer memory for segmentation.

For AAL5 CS-PDU, the CPU needs to initialize the CRC32 circuitry with the CRC32 Partial Value before the transfer begins. Once the SAR-SDU is in the cell buffer memory, the APU needs to generate the cell header. The cell header is retrieved from the data structure and the APU may process some of the fields such as the VCI, the VPI, the GFC, the CLP, and the PT.

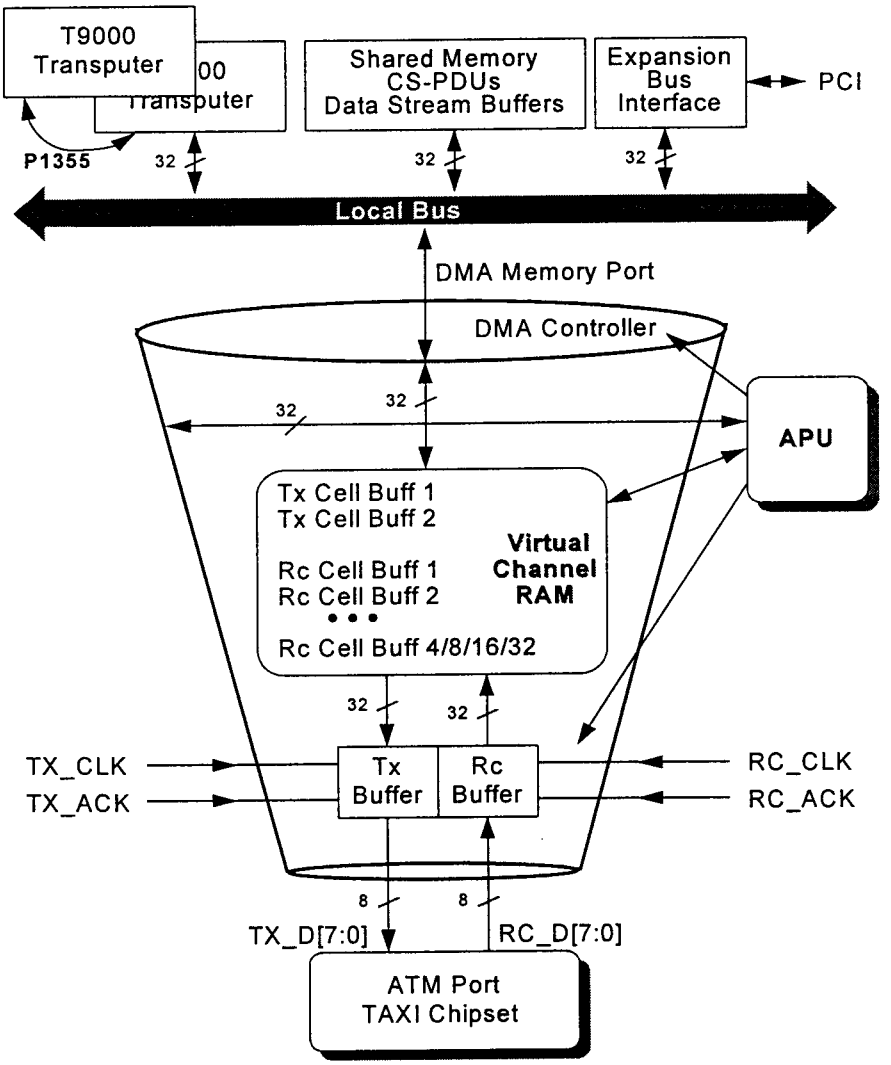


Figure 10. Cell Generation Data Path

Figure 10 shows the cell generation data flow. Almost all aspects of the cell generation process are controlled by the APU under user firmware control. The APU detects external events by periodically polling a Host/DMA Port memory-mapped register that has a bit associated with each possible external event, or by polling the GPINT_TST signal for an indication that an external event has occurred. All incoming cells, arriving over the ACI Receiver through TAXI, are written into the VCR prior to processing. The APU decides whether to terminate a cell or to switch a cell. All outgoing cells are either constructed in the VCR or transferred to the VCR prior to transmission.

The role of the Expansion Bus Interface to be connected to the PCI bus is the network management function. The ALAX will use the graphics-oriented operating systems like Windows 95 and OS/2 for the user interface. This means that the ALAX environment for management demands high bandwidth and performance. The PCI bus is a high performance bus that provides a processor-independent data path between the CPU and high-speed peripherals. Intel corporation announces PCI chipset called the Intel 82420 PCI Chip Set. PCI expects to enhance the ALAX through improved performance and provides the foundation for improving the use for the next version of the ALAX.

For detailed protocol architecture and signaling function in the T9000 and the ATMizer, refer to the ATM LAN Access Switch (ALAX): Protocol Architecture, written by Dr. Jangkyung Kim. The document describes the protocols used in the ALAX like ATM protocol, LAN Emulation, signaling, and local addressing etc..

The ATM forum provides specifications for physical layer ATM interfaces for the User-Network Interface (UNI). DS3 44.736 Mbps, TAXI 100 Mbps and two 155.52 Mbps interfaces are specified in ATM UNI 3.0. The ALAX uses the physical layer for 100 Mbps multimode fiber interface. The functions of the physical layer (U-plane) are grouped into the Physical Media Dependent Sublayer (PMDS) and the Transmission Convergence Sublayer (TCS) as shown in Figure 11.

Transmission Convergence Sublayer	*Cell delineation* *HEC generation/ verification*	● Each cell transmitted shall be preceded by a "TT"(start of cell) code. ● There must be a minimum of 1 JK symbol pair transmitted on the link every 0.5 second. ● The 54 bytes (53bytes of cell plus start of cell code) shall be contiguous on the line. ● The transmitter calculates the HEC value for the first four octets of the cell header, and inserts the result into the HEC field, the last octet of the header. ● The HEC field shall be the remainder of the division (modulo 2) by the generator polynomial x^8+x^2+x+1 of the polynomial x^8 multiplied by the content of the header excluding the HEC field. ● On detection of a header error, the cell shall be discarded.
Physical Media Dependent Sublayer	*Bit timing* *Line coding* *Physical medium*	● This Physical Layer shall operate at 125 Mbaud line rate. ● This Physical Layer shall use the 4B/5B coding based on the ANSI X3T9.5 (FDDI) committee. ● The fiber connector used shall be the MIC duplex connector specified for FDDI in ISO DIS 9314-3. ● The local physical interface shall meet the FDDI PMD specification as defined in ISO DIS 9314-3.

Figure 11. Physical Layer Functions (U-plane)

The ALAX applies Am7968 and Am7969 TAXIchip™ from Advanced Micro Devices for the physical interface. The TAXIchip system provides a mean of connecting parallel data systems over a serial link. The ATM Cell Interface (ACI) within the ATMizer contains both the ATM port-side transmitter and receiver functions and connects directly to the TCS framing circuitry that is the TAXIchip set for PMDS. Cell boundaries are asynchronous. This means they can occur any time the line is idle. Each cell is preceded with the TT code, followed by the 53 byte cell. The TT code denotes the start of cell at the receiver. The 54 bytes must be contiguous on the line. The ACI controls cell delineation and contains HEC generation/verification as described in Figure 11.

3. IEEE P1355 Virtual Switch Matrix Module

The key component of the module is the ST C104, Asynchronous Packet Switch, from SGS-Thomson Microelectronics Group. The ST C104 is a 32 way asynchronous packet switch. It connects 32 serial communication links to each other via a 32 x 32 non blocking crossbar switch, enabling packets to be routed from any of its links to any other link. Each link can operate at up to 100 Mbits/s, providing a bi-directional bandwidth of 19 Mbytes/s. The ST C104 supports a rate of packet processing of up to 200 Mpacket/s. The communication links of this module, DS-link of IEEE P1355, divide the data and control path in half. The DS-links connect with the T9000 communication channel of each interface module. One connects the DS-link for data and the other connects the Control-link for control to the T9000.

The ST C104 uses wormhole routing in which the routing decision is taken as soon as the header of the packet has been input. If the output link is free, the header is output and the rest of the packet is sent directly from input to output without being stored. Wormhole routing is invisible as far as the senders and receivers of packets are concerned. Its major effect is to minimize the latency in the message transmission. The ST C104 is referred to as a dynamic message router because when a message arrives on one of its ports, it is routed automatically to the correct destination.

The following Figure 12 shows the block diagram of this module. It can distinguish three types as shown in Figure.

- Asynchronous Packet Switch, ST C104,
- Local CPU, T9000 with memory, and
- PCI Bus Controller, 82420.

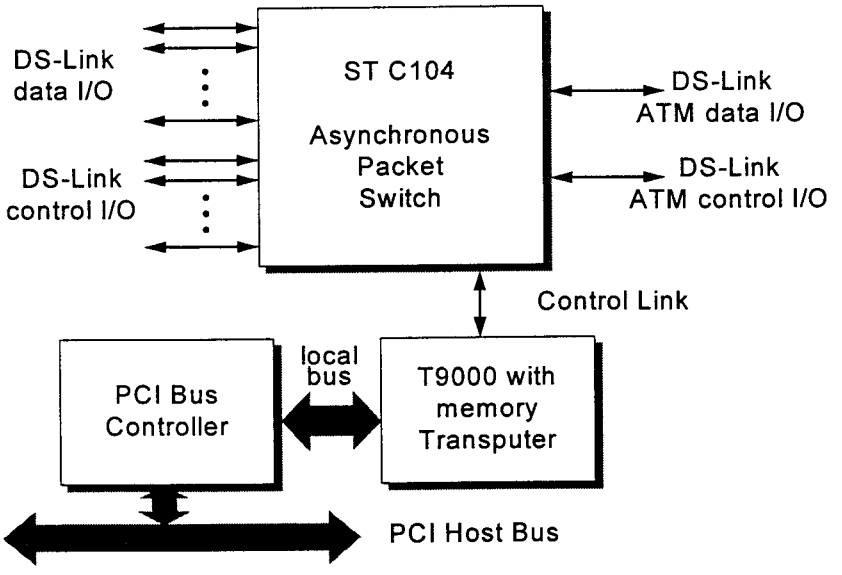


Figure 12. Switch Matrix Module Block Diagram

The ST C104 has the control unit which contains two control links (CLink0 or ControlUp, and CLink1 or ControlDown). Commands can be received on ControlUp from the T9000 and responses and error messages returned to the T9000 in the local transputer. These commands such as CPeek or Cpoke can be used to reset the device and to read and write the configuration registers in the subsystems. ControlDown provides a daisy-chain link, allowing a simple physical connectivity to be used for controlling networks. We have two options to connect control link between Virtual Switch Matrix module (VSMM) and Ethernet or ATM modules; daisy-chained connecting and partitioning. In a daisy-chained connection, the data links of VSMM can extend 32 ports, but the speed and the packet format of control slow down and have complicated structure. On the other hand, a partitioning speeds up a control link and has a simple format. Provisionally, the VSMM on the ALAX is used in a partitioned method to maintain strict separation between the control and data link network. Partitioning allows the single ST C104 on the VSMM to safely implement message routing for both the control and data link networks, without any possibility of an application corrupting the system configuration. Refer to ST C104 data sheet for more information.

The control links use the same electrical and packet level protocols as the DS-links as defined in IEEE P1355. Thus, an ST C104 can be connected by its control link to a data DS-link of a controlling processing node and the node can issue commands to the ST C104.

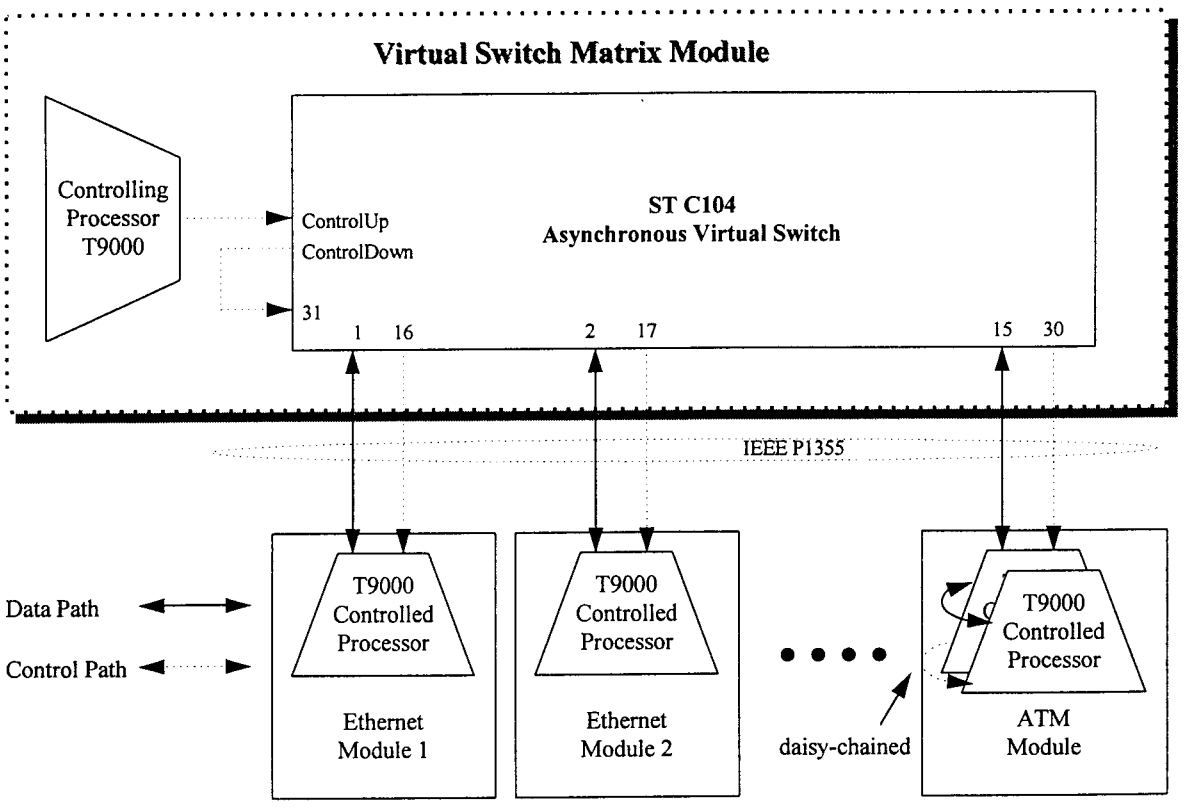
The commands which are received or returned through the control link are as follows;

- Start: To establish the virtual link between the controlling T9000 and the ST C104,
- Reset: To reset the ST C104,
- Identify: To check the contents of a network,
- RecoverError: To be used in error recovery on the control link,
- CPeek: The CPeek command includes a 2 byte address which points to a register in the configuration address space,
- CPoke: To set the value of a configuration register. It writes the data to the configuration space register at the given address, and
- Errors: The ST C104 can send an Error message to the controlling process of the T9000, to indicate that an error has occurred.

After a module of the ALAX is reset it must be configured, and can then be loaded with an application program. This is done by means of a network of control links, separate from the data link network. Each module, Ethernet to P1355 or ATM to P1355, has a ControlUp link connection. On the VSMM, the ControlUp links from all of the modules are connected to the ST C104 link switch as shown in Figure 13. This provides a tree-like connection from the ControlUp connection of the VSMM to each module. This means that a controlling processor, T9000, connected to ControlUp of the ST C104 is able to send a control message to, and monitor each module separately, and monitor each module separately, regardless of whether the other module connections exists. It is also more fault-tolerant than a daisy-chained control link connection.

Figure 13 also shows the data links from each module to the ST C104. A module on the ALAX can communicate with any other module on the ALAX. The ST C104 provides a very flexible interconnection between modules on the ALAX.

The ST C104 on the VSMM have to be configured before an application can be loaded and run. This is done automatically by T9000 development tools such as IMS Dx394 T9000 ANSI C Toolset or IMS Dx395 T9000 occam 2 Toolset. Figure 14 illustrates the program building model for the INMOS T9000 ANSI C Toolset. However, the tools require a description of the ALAX and any modules, in the form of an NDL file.



Finger 13. Control and data link connections via VSMM

The software of the transputers can be developed in standard high-level language using cross-compilers running on a range of host machines. All the languages include extensive support, in the form of run-time libraries for concurrency and communication. It is possible to write a program consisting of many concurrent processes entirely in C.

The Network Description Language (NDL) is used to describe the available hardware - the types of processors, their attributes and how they are connected. For example, processor attributes include a description of its memory map and its link speeds. The NDL also describes any packet routing switch devices in the network and their attributes. Attributes of a routing device include the labeling of the routing device, which indicates how packets from processors should be routed through it.

The network description will not normally change unless the hardware is changed. This NDL description of the system is used by the tools for a variety of purposes, from initializing the hardware to mapping application code onto processors. These tools can either read and check the NDL source directly or read a binary version produced by the NDL compiler, i^{ndl}. Figure 15 shows the NDL network description for the VSMM in Figure 13.

The software description has to specify the linked process image for each process in the system and the procedure interface parameters and their types. The software description must also specify the way that communication channels are used between processes.

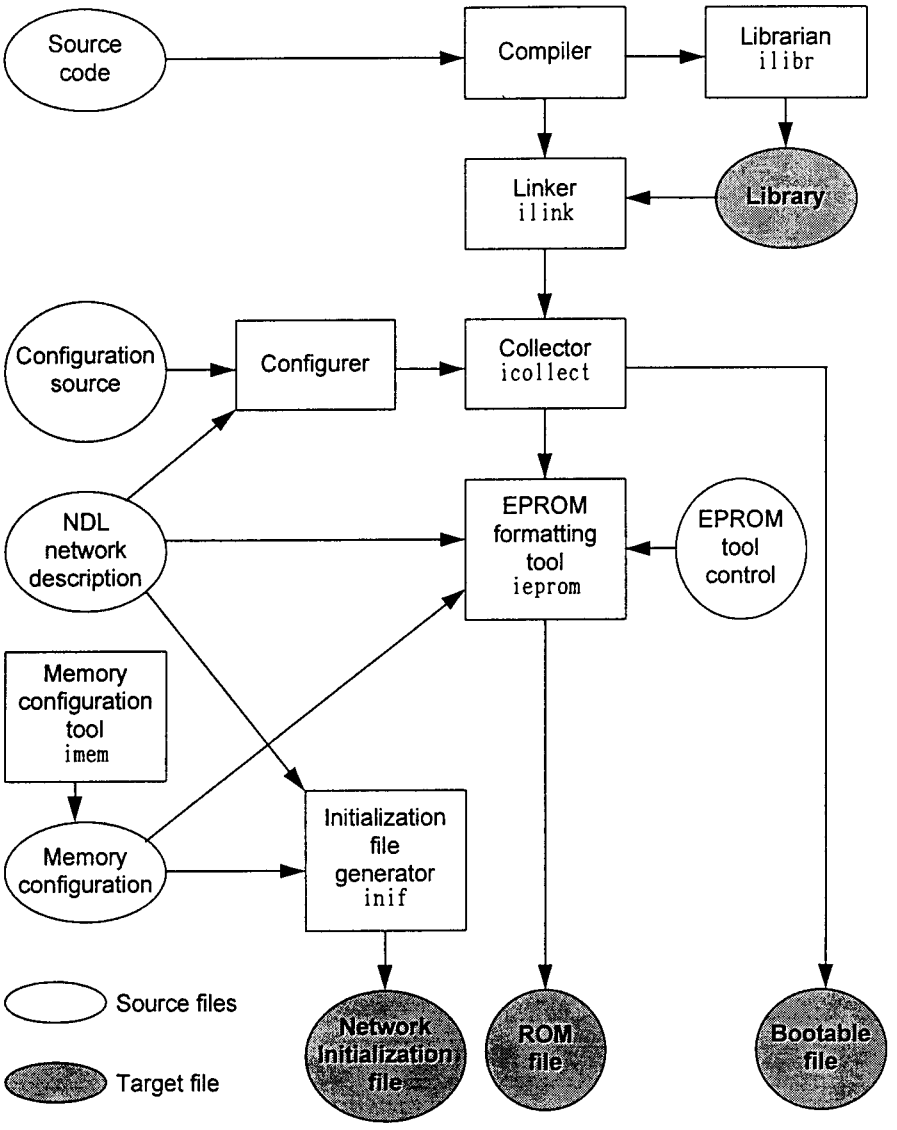


Figure 14. Program build model for T9000 ANSI C Toolset

A controlling processor T9000 network is initialized by first sending control commands to the ST C104 router and the modules connected to the control network and then loading code using the data network. The correct sequence of control network commands is held in a network initialization file, which contains all the information needed to initialize a system through the control links prior to loading the application code. The bootstrap code for each processor, loading code and user application code are all incorporated in the bootable file with the necessary routing information. To load the code onto the network, the bootable file is sent down the data link to the data link network.

```
#INCLUDE "stdndl.inc"

CONTROLPORT host :
HOST IS host[data] :
CONTROL IS host[control] :

ARC HostLink :
[n]NODE Node :                          -- n transputers, in case of the ALAX n = 16 (max)
NODE RootNode :
NODE Switch :
NODE SwitchCONTROL :
NODE SwitchDATA :
NETWORK alaxAtm
 DO
    -- set default attributes
    SET DEFAULT (link.speed.multiply    := 20)        -- set maximum speed 100 Mbits/sec
    SET DEFAULT (link.speed.device      := [1])
    SET DEFAULT (control.speed.devide   := [1])

    -- set default attributes for root node
    SET RootNode (type                  := "T9000")
    SET RootNode (root                  := TRUE)
    SET RootNode (local.rom             := FALSE)
    SET RootNode (pmi.config.inrom      := FALSE)
    SET RootNode (cachesize             := 8)
    SET RootNode (memconfig             := RamMemoryFile)
    SET RootNode (memory                := RamMemoryRegions)

    -- set device attributes for reset of network
    DO i = 0 FOR n
       DO
          SET Node[i] (type               := "T9000")
          SET Node[i] (local.mem          := FALSE)
          SET Node[i] (pmi.config.inrom   := FALSE)
          SET Node[i] (cachesize          := 8)
          SET Node[i] (memconfig          := RamMemoryFile)
          SET Node[i] (memory             := RamMemoryRegions)

    -- set device attributes for switch
    SET Switch (type := "C104")
    SET Switch (link.speed.multiply := 20) -- 2x default to get desired speed due to bug

    SET SwitchCONTROL (type      := "C104PARTITION")
    SET SwitchCONTROL (node      := Switch)
    SET SwitchCONTROL (links     := [[16, 31]])

    SET SwitchDATA (type     := "C104PARTITION")
    SET SwitchDATA (node     := Switch)
    SET SwitchDATA (links    := [[0, 15]])

    #INCLUDE "vsmm.lbl" -- labeling generated automatically by indl

    -- connect control network
    CONNECT Switch[control.up] TO CONTROL
    CONNECT Switch[control.down] TO SwitchCONTROL[link][31]

    DO i = 0 FOR n
       CONNECT SwitchCONTROL[link][i + 16] TO Node[i][control.up]

    -- connect data network
    CONNECT SwitchDATA[link][0] TO HOST WITH HostLink

    DO i = 0 FOR n
       CONNECT SwitchDATA[link][i] TO Node[i][link][0]
:
```

Figure 15. NDL network description example for the VSMM

A configuration source is used to describe the network of processes and channels and the mapping of the processes onto the transputer network. The transputer network is described separately in the network description file. The network description file is referred to by means of a #network configurer directive. This allows the user to map processes in the configuration description onto the processors named in the network description file. The routing of channels may be generated automatically or may be included in the configuration description.

The Virtual Channel Processor (VCP) on the sending transputer packetizes messages to be sent over a link and adds a header to each packet to identify the destination process. At the receiving end, the VCP uses the header to send the data in each packet to the intended process. These headers can also be used for routing packets through a VSMM connecting a number of transputers together. The header still just specifies the destination of the packet.

4. Ethernet to IEEE P1355 Interface Module

This module is an interface gateway for communication between DS-link based T9000 transputer and Ethernet environments. Basically, the ALAX provides a simple and easy mean for running existing applications in the ATM environment. Today, the LAN systems, which were installed to integrate a multiple number of computers and to share data and resources on the premises of universities, research institutions, and businesses, used to be mostly Ethernet. Therefore, we decided to use the Ethernet in the ALAX.

The standards for the LAN protocols, such as logical link control (LLC), carrier-sense multiple access with collision detection (CSMA/CD), token bus, and token ring, were announced in 1985 by the IEEE 802 Committee and they were also adopted by ISO. CSMA/CD is currently the most widely used scheme, and its paradigm product is the Ethernet. In order to use the existing LAN application software, the ALAX contains LAN Emulation that emulates services of existing LANs across an ATM network and can be supported via software layer in end systems as specified in ATM forum, LAN emulation sub-working group. For more detailed information in LAN Emulation over ATM, refer Dr. Jangkyung Kim's paper, ALAX: Protocol Architecture.

Figure 16 shows the block diagram of Ethernet to IEEE P1355 interface module. A standard Ethernet AUI, Twist Pair (TP), or BNC interface is provided on the module. The MAC packets from the Ethernet interface are saved in the memory for the packet conversion of IEEE P1355. Ethernet LANCE informs to T9000 transputer the arrival of the packets that should be converted to the P1355 packets.

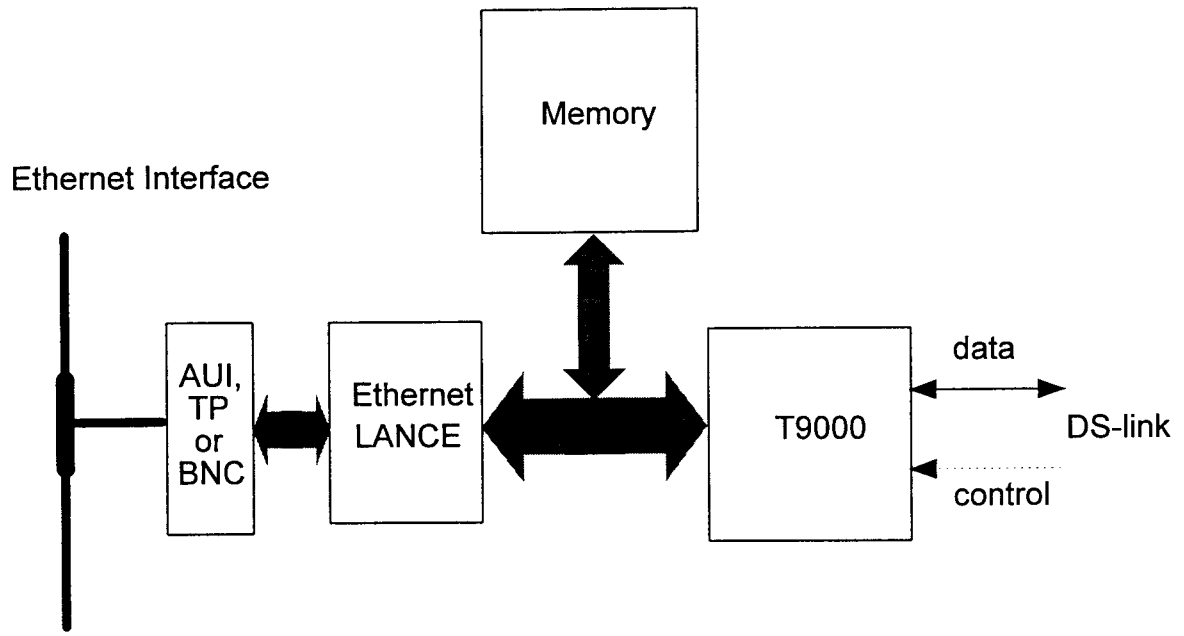


Figure 16. Ethernet to IEEE P1355 block diagram

5. Further Extended Functions

The ALAX has been designed to meet both generic LAN specifications and ATM requirements. It means that the ALAX provides users with a high performance replacement for existing LANs and a modular, scalable and parallel backbone architecture for research institution, campus and business networks.

As described above, the major function of the ALAX is interface between Ethernet and ATM. After we succeed in solving, testing and verifying the ALAX's functionalities, we will extend the function of the ALAX to the following areas.

- 155 Mbps (STS-3c/STM-1) ATM Interface,
- MPEG 2 to IEEE P1355 Interface,
- Multi-protocol over ATM,
- N-ISDN to IEEE P1355 Interface,
- Circuit emulation for DS1, DS3 and n x DS0,
- Applies ATM UNI 4.0,
- Satellite Comunication,
- PCS and
- Enhance software portability to apply common ATM API.

REFERENCES

[1] IEEE, "IEEE Draft Std P1355, Standard for Heterogeneous InterConnect (HIC) (Low Cost Low Latency Scalable Serial Interconnect for Parallel System Construction)", January 1995

[2] ATM Forum, "ATM User-Network Interface Specification, version 3.0", 1993

[3] ATM Forum/94-0035R7, "LAN Emulation Over ATM: Draft Specification - Revision 7", IBM Corp., October 7, 1994

[4] Jangkyung Kim, "ATM LAN Access Switch (ALAX): Protocol Architecture", May 1995

[5] Byeong Gi Lee, Minho Kang & Jonghee Lee, "Broadband Telecommunications Technology", Artech House, 1993

[6] David E. McDysan, Darren L. Spohn, "ATM theory and application", McGraw-Hill, Inc., 1995

[7] Ian Grabam & Tim King, "The Transputer Handbook", Prentice Hall, 1990

[8] Ronald S. Cok, "Parallel Programs for The Transputer", Prentice Hall, 1991

[9] Daniel Minoli & Michael Vitella, "ATM & Cell Relay Service for Corporate Environments", McGraw-Hill, 1994

[10] Jeremy Hinton & Alan Pinder, "Transputer Hardware and System Design", Prentice Hall, 1993

[11] LSI Logic, "ATMizer MegaCore data sheet", November 1993

[12] LSI Logic, "ATMizer™ R/T System Development Platform User's Guide", September 1994

[13] LSI Logic, "L64360 and ATMizer™ Architecture Technical Manual, Preliminary", August 1994

[14] AMD, "Am7968/Am7969 TAXIchip™ Handbook", 1994

[15] SGS-Thomson, "The T9000 Transputer Hardware Reference Manual", 1993

[16] SGS-Thomson, "ST C104, Asynchronous Packet Switch Data Sheet", June 1994

[17] SGS-Thomson, "ST C101, Parallel DS-link Adapter Data Sheet", June 1994

[18] SGS-Thomson,"IMS B104 VME motherboard with DS-Link routing switch datasheet", May 1994

[19] SGS-Thomson,"T9000 Development Tools, Preliminary Datasheets", 1993

[20] Jerome R. Cox, Jr., Michale E. Gaddis & Jonathan S. Turner, "Project Zeus", IEEE Network, March 1993

[21] Jonathan M. Smith & C. Brendam S. Traw, "Giving Applications Access to Gb/s Networking", IEEE Network, July 1993

[22] Edoardo Biagioni, Eric Cooper, & Robert Sansom, "Designing a Practical ATM LAN", IEEE Network, March 1993