NASA Technical Memorandum 106328

On-Board Packet Switch Architectures for Communication Satellites

Mary Jo Shalkhauser and Jorge A. Quintana
Lewis Research Center
Cleveland, Ohio

September 1993

NASA

On-Board Packet Switch Architectures for Communication Satellites

Mary Jo Shalkhauser and Jorge A. Quintana
National Aeronautics and Space Administration
Lewis Research Center
Cleveland, Ohio 44135

Summary

NASA Lewis Research Center is developing an on-board information switching processor for a multichannel communications signal processing satellite. The information switching processor is a flexible, high-throughput, fault tolerant, on-board baseband packet switch used to route user data among user ground terminals. Through industry study contracts and in-house investigations, several packet switching architectures have been examined for possible implementation. We studied three contention-free switching architectures in detail, namely the shared memory approach, the shared bus approach, and the shared memory per beam approach. This paper discusses these three switching architectures and examines the advantages and disadvantages of each approach.

Introduction

NASA Lewis Research Center is developing a meshed, very small aperture terminal (VSAT) satellite communications system which could provide low data rate, direct to user, commercial communications services for the transmission of data, voice, facsimile, datagram, video, and teleconferencing information. Such a system will enhance current communication services and enable new services. The focus of current space segment developments is a flexible, high-throughput, fault tolerant, on-board information switching processor (ISP). This ISP is a baseband space and time switch that routes user data among various user ground terminals.

The satellite architecture is part of a flexible, low-cost, meshed VSAT network (ref. 1) which provides contiguous United States coverage through eight fixed uplink antenna beams and eight hopping downlink antenna beams. The ISP onboard the satellite provides connectivity among the uplink and the downlink beams which enables thousands of low-rate users to communicate with each other. Two types of user traffic, packet data (packet switched) and nonpacket data (circuit switched), are supported by the satellite network. The ISP will support only packetized data. It requires user ground terminals to create packets from circuit switched data.

This paper examines the architectures for three packet switching approaches for the ISP: the shared memory, the shared bus, and the shared memory per beam architectures. It discusses the advantages and disadvantages of each of the three approaches.

Background

Data packets are fixed in size and contain 2048 bits. To reduce the amount of on-board storage, packets are divided into 16 subpackets of 128 bits each. The first subpacket of a packet contains header information indicating the packet source, the destination downlink beam, and dwell for point-to-point communication traffic. The header also indicates multiple destination beams or dwells for multicasting. Figure 1 shows the packet structure.

Uplink data is transmitted using a multifrequency time division multiple access (MF–TDMA) technique. Downlink data is transmitted using a time division multiplexed (TDM) format. The uplink and the downlink frames are divided into 16 subframes to take advantage of the subpacket structure. The ISP processes one subframe at a time so the switch stores only a single subframe of data. The first subframe of a frame contains all the header subpackets. The switch uses these to set up the routing for the entire frame. Each downlink frame contains eight dwell times. The duration of each dwell is variable

according to the needs of the current satellite traffic. Figure 2 shows the structure of both the uplink and the downlink frames. Reference 1 contains a detailed description and analysis of the packet and frame structures.

Shared Memory Switch Architecture

The shared memory packet switch utilizes a single data memory for storing and switching packets onboard the satellite. Incoming packet data is written into the shared memory at the first available unused memory location. Switching of the packet data to the correct destination beam and dwell is achieved by controlling the order in which data are read from the memory.

A block diagram of the shared memory packet switch architecture is shown in figure 3. The input receives data from the uplink ports and combines the data onto a single high-speed time division-multiplexed (TDM) bus. As subpackets arrive, they are stored into first-in first-out memories (FIFO's), one for each uplink port. During the read cycles, each of the FIFO's is accessed one at a time sequentially, allowing all the data to be multiplexed onto a single TDM bus.

The memory-based switch routes input subpackets to their correct downlink beam and dwell. The first subframe of the uplink frame contains all the packet headers. The routing control is set up during the first subframe. As each subpacket arrives at the shared memory via the TDM bus, the data are written into the first available shared memory address location. Simultaneously, a pointer to the shared memory address location is written into an address control memory. The shared memory is designed in a dual port configuration so that reading and writing can occur simultaneously. At startup, a full frame of data is written into the shared memory before any data are read out. Subsequently, data received during frame M will be written into the shared memory while data received during the previous frame (M-1) are read out.

Subpacket data switching is controlled by the beam and address control memories. The beam control memories consist of a pair of ping-ponged FIFO's for each downlink dwell (64 FIFO pairs). During the first subframe of each frame, the subpacket data are routed to the header decoder which determines the packet destination and creates an enable signal to the proper beam control FIFO. A pointer to an address control memory location is then written into the FIFO. The control memory location contains the pointer to the shared memory location of the stored subpacket. For multicasted subpackets, only one copy of the subpacket is stored in the shared memory. The header decoder enables multiple beam control FIFO's to take care of the multiple destinations of the subpacket. The contents of the beam control FIFO's remain unchanged for the remainder of the frame. For subframes 2 through 16 of the frame, subpackets continue to be written into the shared memory. These subpackets are received in the same order as in subframe one and the address control memories are sequentially accessed to write new pointers to the shared memory location of the new subpackets.

The address pool FIFO contains a list of all currently unused shared memory addresses. At startup, the address pool FIFO contains all the shared memory addresses. When a shared memory address is used, it is removed from the address pool buffer. As each packet exits the data memory, the shared memory address becomes available and is written into the address pool buffer.

The beam control FIFO's are accessed sequentially to read the downlink data out of the data memory. First, the beam control FIFO reads the address memory corresponding to the first packet in downlink beam one and dwell one. The beam control FIFO contents point to an address control memory location which in turn points to the shared memory address containing the correct data subpacket. The beam control memories are rotated through sequentially until all data for that subframe have been read. The beam control memories and the address control memories are arranged in a ping-pong configuration so that one memory is written to while the other is read from.

The switch output buffers the downlink data for transmission. The packet data on the high-speed TDM bus from the output of the shared memory are demultiplexed to the correct destination beams by rotating the writes sequentially to each of the downlink FIFO's.

Two important issues in packet switches are contention and congestion. Contention occurs when two or more packets are routed to the same switch output at the same time. (Blocking is contention internal to the switch.) Congestion occurs when too many data are routed to a buffer causing the buffer to overflow. The shared memory architecture is nonblocking and free of output contention (ref. 2). Switch congestion, however, can cause the beam control memory FIFO's to overflow. Therefore, congestion control algorithms must be considered with this approach. The architecture readily supports both point-to-point and multicast communication traffic. Both types of data are handled in virtually the same manner.

The biggest advantage of the shared memory architecture over the other two approaches investigated is the efficient use of the data storage memory that is shared by all of the downlink ports and that is sized for the uplink capacity. Because incoming subpackets are always stored in the first available shared memory location, and because the memory is dual-ported, additional memory savings are realized. This approach uses one-half the memory of a ping-pong data memory approach.

The control memories—beam control memories, address control memories, and the address pool FIFO—consume a large portion of the total onboard memory. Since these control memories store pointers rather than data, the total memory used in the shared memory architecture is still low. The shared memory architecture requires wide, high-speed data buses. The bus size and memory access time limits the ultimate throughput and speed of the switch. Switching control, which requires double pointers to store and switch data, is relatively complex.

Shared Bus Architecture

The shared bus packet switch uses a shared, high-speed TDM bus to interconnect the uplink and downlink ports onboard the satellite. The architecture uses one downlink FIFO per dwell (64 total) to store the data transmitted to that dwell.

A block diagram of the shared-bus architecture is shown in figure 4. The input multiplexer combines the lower data rates input from the uplink ports into a high speed TDM shared bus to be processed by the switch. The shared bus interconnects the input ports with modules such as the control processor, 64 FIFO memories, and a dwell processor (ref. 3). These modules are described in the following sections.

The control processor receives data from the high speed shared bus. From the packet headers, the control processor generates the appropriate strobes (enables) for the dwell FIFO memories. These strobes enable the correct destination FIFO(s) so packet data can be stored. This process, and the fact that the bus is faster than all the input and output data rates, makes the architecture nonblocking and contention free.

The FIFO memories provide temporary storage while the downlink dwell duration is adaptively controlled. The eight dwell memories associated with a downlink beam share a single output data bus. The dwell processor clocks the data out from each dwell and writes it onto the output bus.

The dwell processor is responsible not only for clocking out the data from the appropriate FIFO memory, but also plays an important role in congestion control. The fill level threshold for each FIFO is programmable and the dwell processor monitors this level to prevent overflows in the system. The dwell processor also controls the duration and order of the dwells. The simplicity of this approach and the convenience of adaptive control compensate for the fact that one processor is needed per downlink beam.

Two different random access memories (RAM's) are used to accommodate multicasting services. The first is a high speed RAM that reads and stores the destination addresses for each of the uplink channels. The second RAM contains a look-up table used to create enables to the appropriate downlink FIFO's. The first RAM provides the writing address for the second RAM. The data bus width of the second RAM represents the 64 enable signals, one for each downlink FIFO. A logic "1" indicates that a specific FIFO is part of the multicasting address. Different multicast "groups" can be created and stored in the second RAM where they will be accessed according to the header destination address.

The shared bus architecture has many advantages and disadvantages when compared to the shared memory architecture. A shared bus packet switch is simpler to design and to test. By eliminating additional

processing for multicasting, it also reduces the processing complexity, which in turn increases the reliability of the system. Contention and congestion are handled with no further processing in this architecture, which further increases its desirability.

A major drawback of the shared bus architecture is the inefficient use of data storage memory because the data memory is not shared among the downlink dwells or beams. The amount of memory required for each downlink FIFO depends on the projected message traffic. In the worst case scenario, all the traffic could go to a single downlink dwell. Therefore, each FIFO memory must be sized for a dwell duration nearly equal to the full downlink frame duration, and memory is used very inefficiently.

Another major drawback of this system is that each dwell has a specific FIFO assigned to it. This means that if a memory fails onboard one area on the ground illuminated by the corresponding antenna beam dwell will be totally disabled. To avoid this situation, this architecture may require additional programming to dynamically switch to an under-utilized downlink FIFO in the case of a single FIFO failure.

Shared Memory Per Beam Architecture

The shared memory per beam architecture is a combination of the shared memory and the shared bus architectures. In this architecture, each downlink beam has a single data memory that is shared by each of the eight downlink dwells within that beam. Switching is performed by first routing the subpackets to their correct destination beam and then reordering the subpackets into their correct dwell locations.

Figure 5 contains a block diagram of the shared memory per beam architecture. As in the shared memory approach, the input module receives data from the uplink ports and multiplexes all the data onto a single TDM bus. The switching portion of the shared memory per beam architecture differs from that in the shared memory architecture. In the shared memory per beam architecture, each downlink beam has a beam memory module which contains a data memory shared by all eight dwells in the beam and the associated control circuits. This means that all the processing required for a fully shared memory architecture is distributed among eight downlink beam modules. These beam memory modules are each a scaled down version of the shared memory switching modules.

The beam header decoder receives a TDM data bus from the input portion of the architecture. The header decoder consists of a look-up table which examines the packet headers and creates enable signals to the packet's destination beam memory module. The TDM bus is routed to all of the beam memory modules and the enables determine which beam memory module should process the subpacket. Each beam memory module contains a dwell header decoder which determines the destination dwell for the received subpacket and then enables the correct dwell control memory to store pointers to the address control memory.

The beam memory module also contains a shared memory, an address pool FIFO, and address control memories. These blocks are identical in function to the blocks with the same names in the fully shared memory architecture. However, because only the data destined for a single downlink beam must be stored, all the memories (both data and control) are smaller.

The output portion of the shared memory per beam architecture is different from that of the shared memory architecture because no output demultiplexer is needed. One output FIFO is used per downlink beam module. These FIFO's are used to assemble the downlink data into the proper downlink TDM transmission format.

The shared memory per beam architecture is a hybrid of the shared memory and the shared bus architectures, and is a good compromise between the two approaches. The shared memory per beam approach is a much more modular approach than that of the shared memory architecture and allows modularity of memory, distributed processing, slower bus speeds, and small bus sizes. However, the total data memory requirement is greater than that required by the shared memory approach because the total memory is based on the downlink capacity rather than the uplink capacity, as in the shared memory approach. The shared memory per beam approach still uses significantly less data memory than the

shared bus approach, however. Control of the shared memory per beam approach is similar to that of the shared memory control, but more complicated than the shared bus control.

The modularity of the shared memory per beam approach, and the compromise it represents between processing complexity and memory efficiency, make this approach the best choice of the three for the packet switch architecture in the described application.

Concluding Remarks

Three different approaches to the implementation of a packet switch have been offered and described. The shared bus packet switch is clearly simpler to design and involves less processing than either shared memory design, but its memory inefficiency is a major drawback that onboard processing packet switches cannot afford. A fully shared memory packet switch is significantly more memory efficient than a shared bus approach because the memory can be shared and reconfigured. The shared memory packet switch is also more efficient in handling the worst case traffic load because the shared memory is sized for the maximum uplink capacity, rather than the maximum downlink capacity as in the shared bus approach. However, it will increase the amount of processing and complexity onboard the satellite.

A good compromise between these two approaches is the shared memory per beam architecture. As discussed, the memory efficiency of this approach is much higher than that of the shared bus architecture and only slightly worse than that of a shared memory. Additionally, all the processing tasks associated with the shared memory are distributed among the eight downlink beams.

The amount of processing and memory onboard the satellite should be as efficient as possible. NASA has chosen a hybrid architecture that decreases the requirements for fast processing and on-board memory, and also increases the efficiency of onboard memory utilization.

References

1. Ivancic, W., et al.: Destination Directed Packet Switch Architecture for a Geostationary Communications Satellite Network. IAF Paper 92–0413, 1992.

2. Shyy, D.; and Inukai, T.: Information Switching Processor (ISP) Contention Analysis and Control. NASA CR–191130, 1992, pp. 13–15.

3. Technical Support for Digital Systems Technology Development: Task Order 1, ISP Contention Analysis and Control. Space Systems/Loral, Palo Alto, CA; NASA Contract NAS3–25934, 1992.

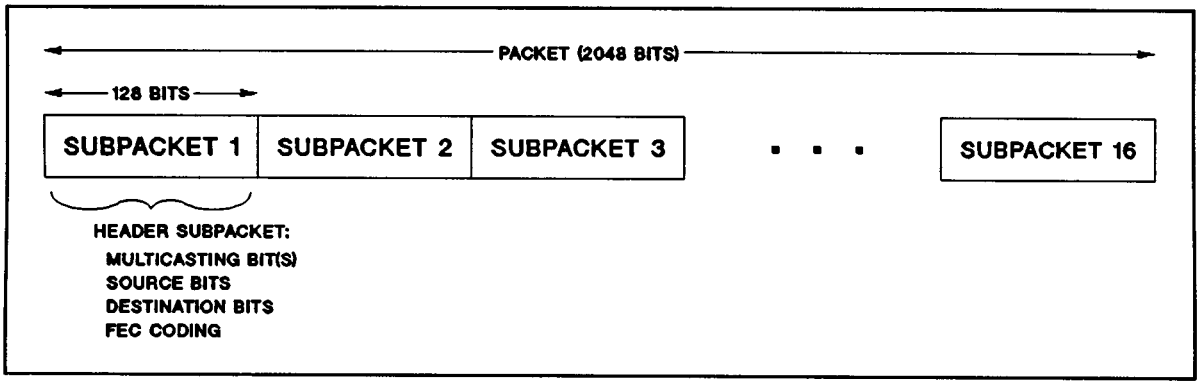

Figure 1.—ISP packet structure.

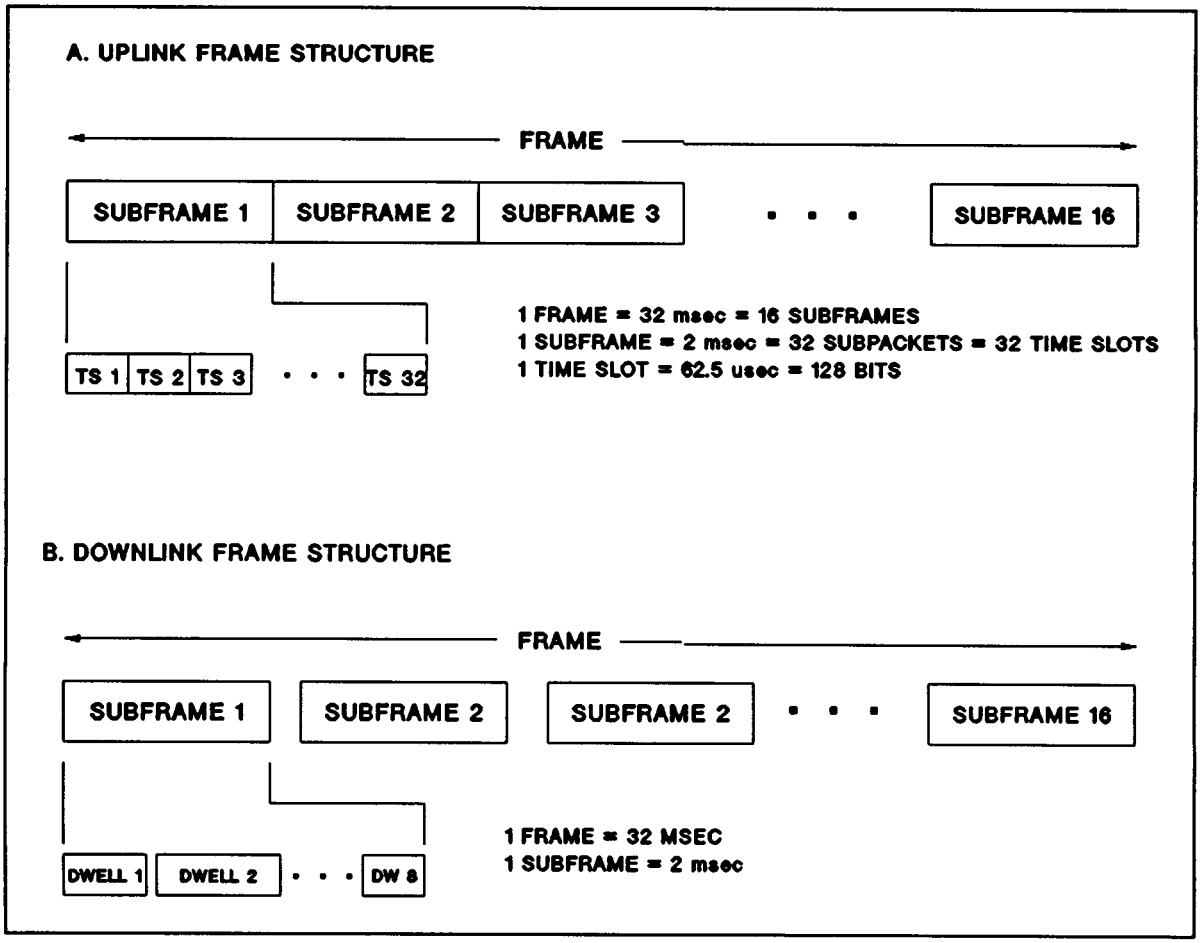

Figure 2.—Packet structures.

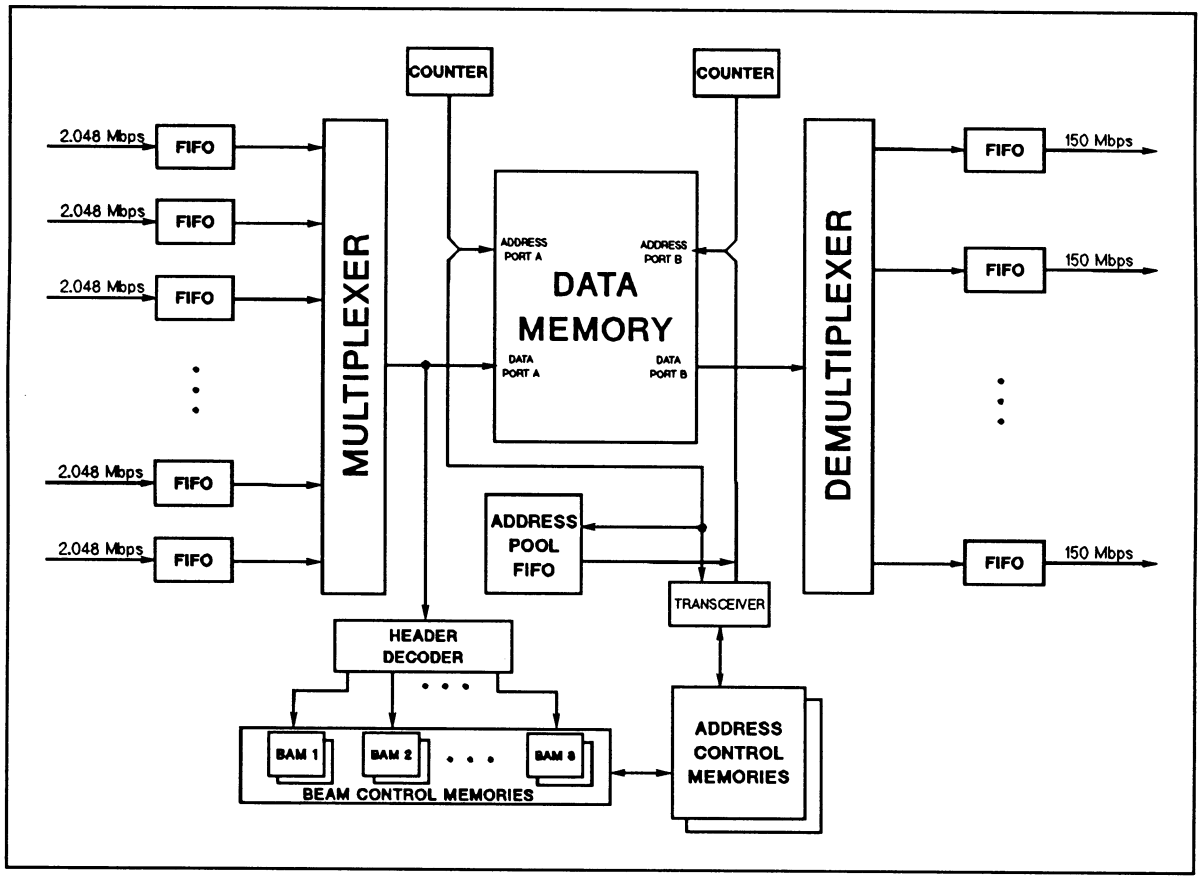

Figure 3.—Shared memory packet switch architecture.

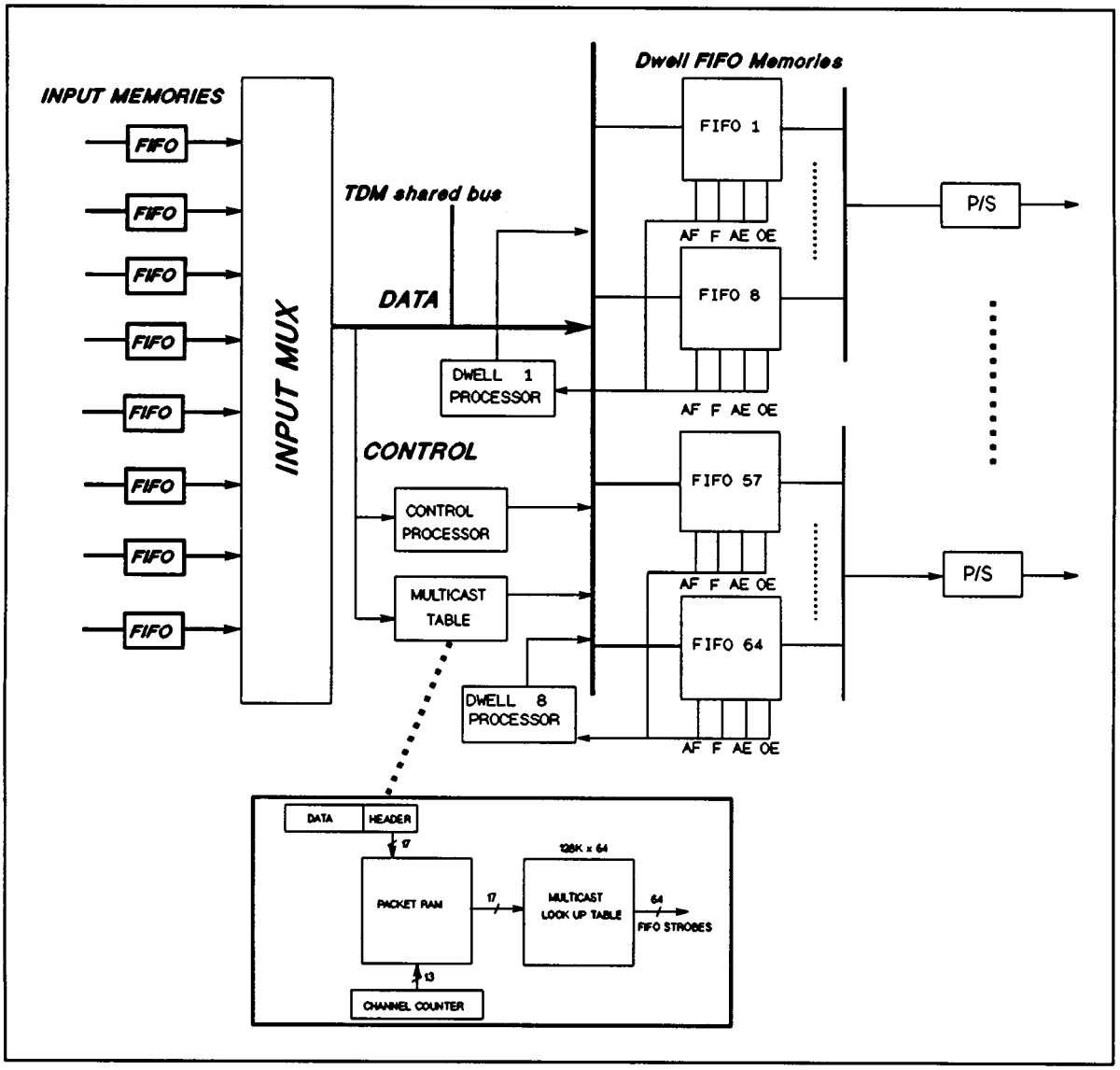

Figure 4.—Shared bus packet switch architecture.

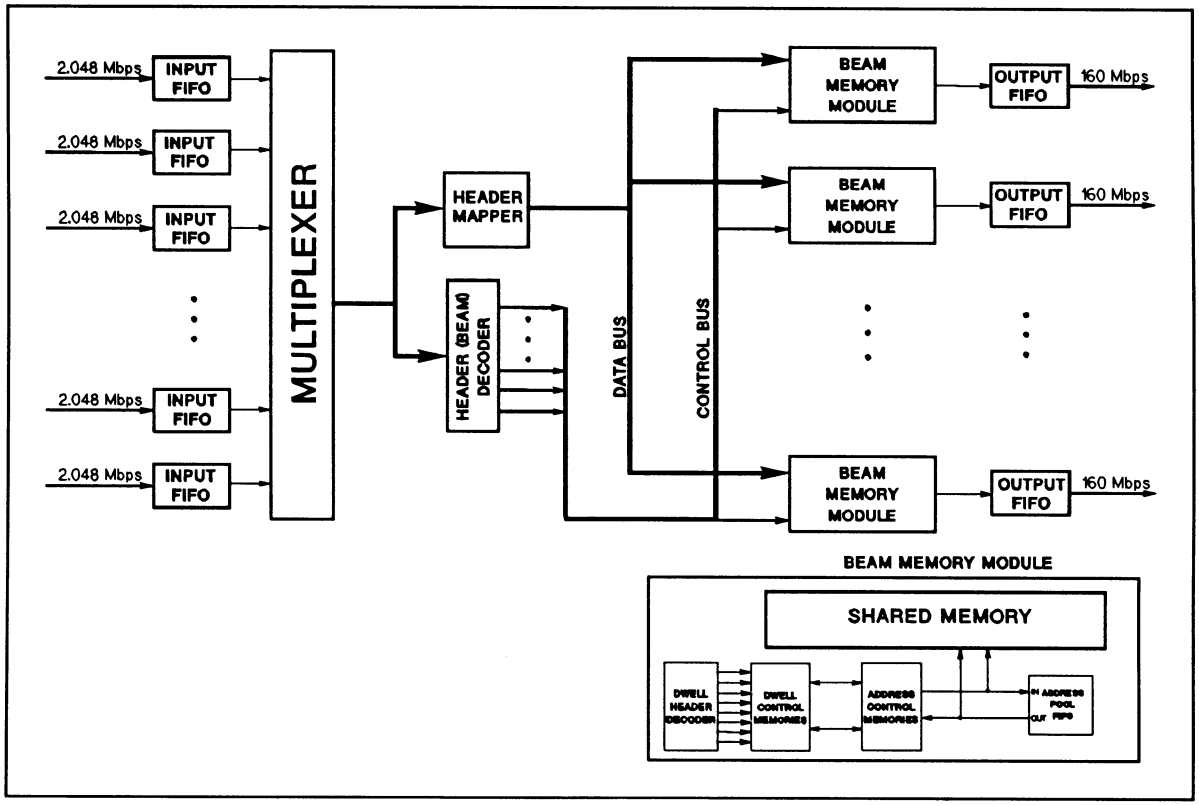

Figure 5.—Shared memory per beam packet switch architecture.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503.

1. AGENCY USE ONLY (*Leave blank*)	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	September 1993	Technical Memorandum

4. TITLE AND SUBTITLE

On-Board Packet Switch Architectures
for Communication Satellites

5. FUNDING NUMBERS

WU–506–72–21

6. AUTHOR(S)

Mary Jo Shalkhauser and Jorge A. Quintana

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

National Aeronautics and Space Administration
Lewis Research Center
Cleveland, Ohio 44135–3191

8. PERFORMING ORGANIZATION REPORT NUMBER

E–8087

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

National Aeronautics and Space Administration
Washington, D.C. 20546–0001

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

NASA TM–106328

11. SUPPLEMENTARY NOTES

Responsible person, Mary Jo Shalkhauser, (216) 433–3455.

12a. DISTRIBUTION/AVAILABILITY STATEMENT

Unclassified - Unlimited
Subject Category 17

12b. DISTRIBUTION CODE

13. ABSTRACT ***(Maximum 200 words)***

NASA Lewis Research Center is developing an on-board information switching processor for a multichannel communications signal processing satellite. The information switching processor is a flexible, high-throughput, fault tolerant, on-board baseband packet switch used to route user data among user ground terminals. Through industry study contracts and in-house investigations, several packet switching architectures have been examined for possible implementation. We studied three contention-free switching architectures in detail, namely the shared memory approach, the shared bus approach, and the shared memory per beam approach. This paper discusses these three switching architectures and examines the advantages and disadvantages of each approach.

14. SUBJECT TERMS

Packet switching; Satellite; Information switching processor; On-board processing

15. NUMBER OF PAGES

11

16. PRICE CODE

A03

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
Unclassified	Unclassified	Unclassified	