VLSI HIGH SPEED PACKET PROCESSOR

Gerald J. Grebowsky and Carol T. Dominy
Data Systems Technology Division, Code 520
mission Operations and Data Systems Directorate
NASA Goddard Space Flight Center
Greenbelt, Maryland 20771

ABSTRACT

The Goddard Space Flight Center Mission Operations and Data Systems Directorate has developed a Packet Processor card utilizing semi-custom very large scale integration (VLSI) devices, microprocessors, and programmable gate arrays to support the implementation of multi-channel telemetry data capture systems. This card will receive synchronized error corrected telemetry transfer frames and output annotated application packets derived from this data. An adaptable format capability is provided by the programmability of three microprocessors while the throughput capability of the Packet Processor is achieved by a data pipeline consisting of two separate RAM systems controlled by specially designed semi-custom VLSI logic.

Key words : packet, VLSI, microprocessor, gate array.

BACKGROUND

The Mission Operations and Data Systems Directorate of the Goddard Space Flight Center has developed a multi-channel Data Capture Front End (DCFE), also referred to as Enhanced Front End Hardware (EFEH). A DCFE channel will receive serial data in NASCOM block format, synchronize and extract telemetry transfer frames, correct errors in data, and reassemble application packets for processing in a *Level Zero Processor*. A Packet Processor utilizing semi-custom very large scale integration (VLSI) devices, microprocessors, and programmable gate arrays has been designed to support the implementation of this system. Specific requirements for assembling packetized source data were derived from the *Goddard Space Data Packetization Standard (Standard 3.3)*, a subset of the *Consultative Committee for Space Data Systems (CCSDS) Packet Telemetry Recommendation*. This standard packet format includes the contiguous insertion of variable length source packets into telemetry transfer frames from up to 8 multiplexed virtual channels. The locations of packets within the frame data fields are derived from a first header pointer in the frame header and packet lengths found in subsequent packet headers. The general requirement emphasized during the design of this capability was the adaptability to extract source data from various telemetry data formats, especially packetized data, at throughput rates up to 20 Mbps.

GENERAL DESCRIPTION

The Packet Processor design is outlined in Figure 1. The design was based on minimizing the amount of data flow through the microprocessors by pipelining the full data path and providing a means for extracting only data needed for assembling application packets. The throughput capability of the Packet Processor is achieved by a data pipeline consisting of two separate RAM systems controlled by specially designed semi-custom VLSI logic. The first, the Tribuffer, triple buffers input telemetry transfer frames (up to 4kbytes/frame) to allow a full frame period for extraction of frame and packet header data. The second and larger Reassembly RAM stores frame data separated by virtual channel (128kbytes/ channel) and provides the capability to reassemble packets by selective readout of packet pieces.

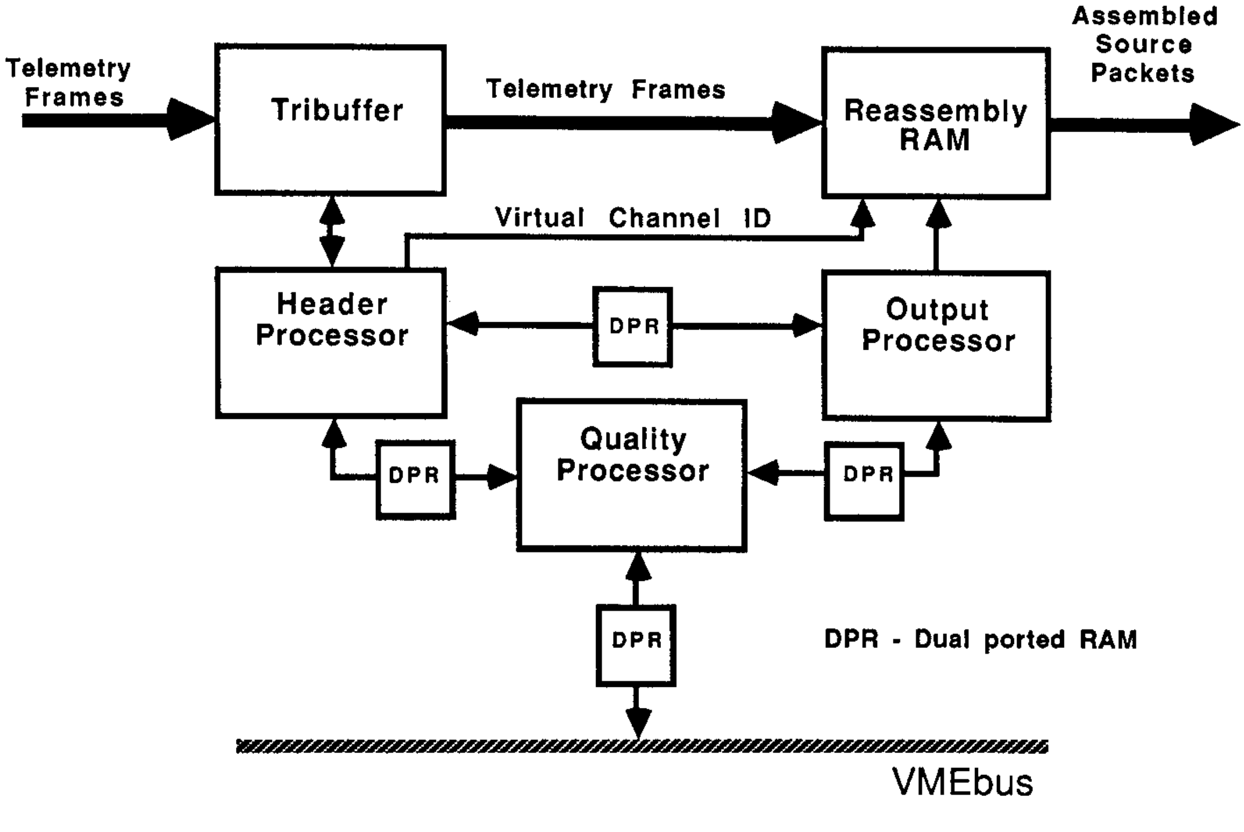

Figure 1. Packet Processor

An adaptable format capability is provided by the programmability of the three microprocessors (MC68020's) labeled Header Processor, Quality Processor, and Output Processor. The Header Processor has interface connections to the Tribuffer Controller (VLSI) and a virtual channel selector control to the Reassembly RAM Controller (VLSI). The Output Processor has interface connections to the Reassembly RAM Controller. The Quality Processor does not interface with the data pipeline directly, but does communicate with the Header and Output Processors as well as with a Channel Controller (VMEbus) via dual ported RAM. In addition the memory spaces of all three processors are accessible for read and write by the Channel Controller for loading software and/or session set up parameters.

PIPELINE DESCRIPTION

Tribuffer

The VLSI Tribuffer Controller (Figure 2.) provides the following capabilities:

- automatic cyclic multiplexing of 3 RAMS based on completion of input, output, and interval reading.
- interrupt identifies start of new cycle.
- 8 - bit input and output.
- 10mhz clock provides input and output transfer rates of 80 mbits/sec.
- programmable frame length for data input.
- automatic input of data from an input FIFO.
- automatic detection of short or long input frames.
- programmable start and end addresses for data output.
- automatic output of frame data.
- programmable addresses for reading selected data intervals from a stored frame.
- programmable address increment for deinterleaved reading of selected data intervals.
- selectable 8 or 16 bit data path for output of selected data intervals.

The Header Processor controls all the programmable features of the Tribuffer Controller. With the exception of header (data interval) reading, all options are set once as part of an acquisition session startup routine. The internal logic of the Tribuffer Controller consists of three address counters (programmable start and end addresses), multiplexer interfaces to 3 RAMs and 3 I/O ports, and the interconnecting control logic. This implementation was chosen to provide fast address generation for RAM read/write. The real throughput rate of the Packet Processor is dependent on the format of the data frames and the number of header interval accesses required per frame. The input FIFO provides buffering of input data to allow longer than one frame period for extraction of header data intervals from an occasional busy frame assuming the pipeline transfer rate will provide the catch up

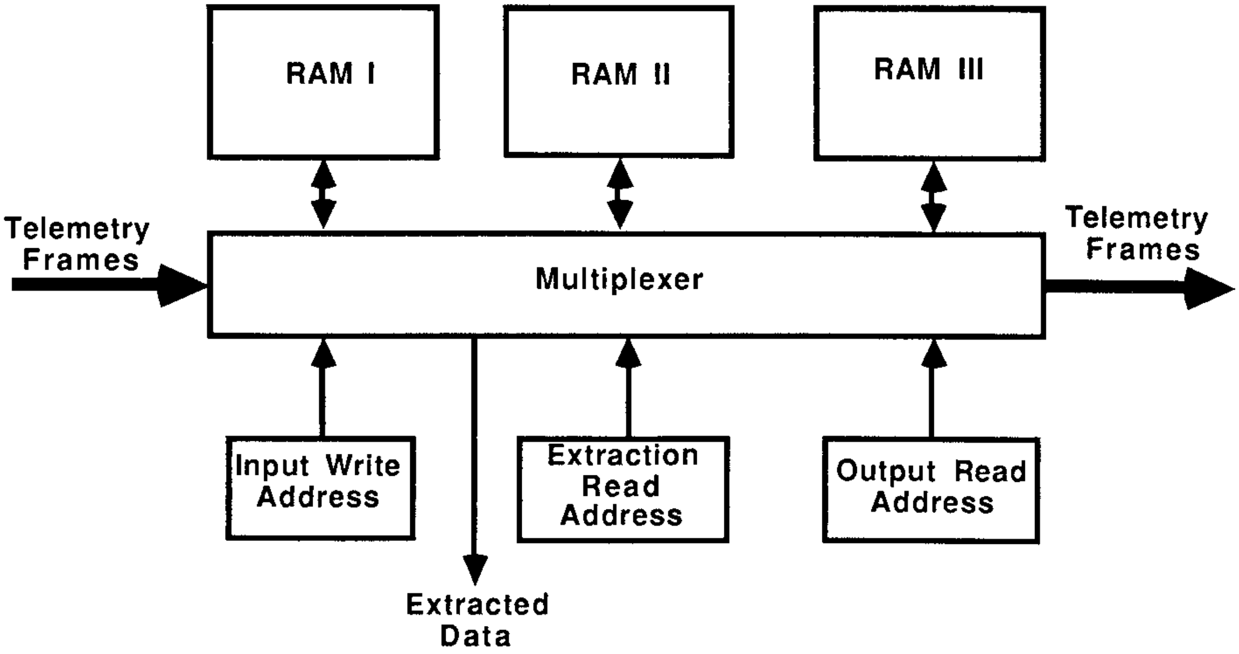

Figure 2. Tribuffer

capability on the next cycle. This capability attempts to alleviate the problem of limiting throughput by an occasional (e.g., 1 out of 3) frame requiring more than a frame period to extract header data.

Reassembly RAM

The Reassembly RAM is partitioned for 8 virtual channels (128 kbytes/channel). The VLSI Reassembly RAM Controller (Figure 3.) provides the following capabilities :

- 8-bit input and output paths.
- 16-bit RAM data paths.
- automatically stores frame data from Tribuffer Controller into contiguous RAM for a virtual channel
- latches out starting address for each input frame.
- output is instruction driven :
  reads intervals of data from RAM to produce reassembled packets.
  inserts annotation or fill data.
- check sum test on instruction set.

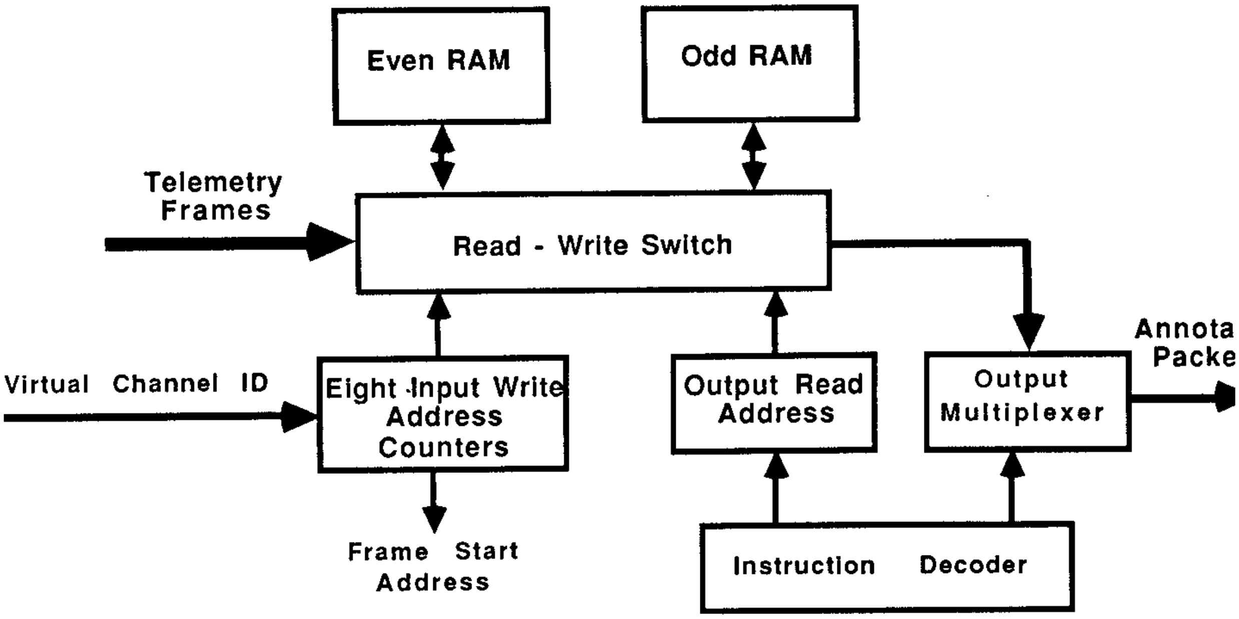

Figure 3. Reassembly RAM

The internal logic of the RAM Controller includes 8 cyclic write address counters which generate addresses for the 8 virtual channel RAM partitions and a read counter with programmable start and end addresses. An 8-16 bit conversion of input data and a 16-8 conversion of output data was incorporated into the RAM Controller to allow use of slower RAM chips for cost reduction. For the same reason, the RAM is also partitioned by even and odd address and read/write is ping-ponged between even and odd to provide simultaneous read and write without dual ported RAM. The virtual channel ID is latched to the RAM Controller by the Header Processor for selection of the appropriate channel partition and associated write address counter. The transfer of a frame of data from the Tribuffer is handled automatically by direct handshake logic signals initiated by the cycling of the Tribuffer. The starting address for a frame input is latched out to the Output Processor. The virtual channel, start address, and end address for reading packet pieces out are provided by instructions from the Output Processor. An output multiplexer allows the insertion of annotation or fill data into the assembled output packet.

PROCESSOR INTERFACES

Dual ported RAM interfaces are used for communication between each pair of processors. Communications with a Channel Controller is also provided through dual ported RAM interfaced to the Quality Processor. The Header Processor passes header data to the Quality Processor, and passes packet piece locations within a frame to the Output Processor. The Output Processor maintains a directory of packet piece locations using the intra-frame locations provided by the Header Processor and a frame starting address from the RAM Controller. The Quality Processor performs analysis of header data to define the reassembly of source packets and generates quality and accounting information. A list of pieces and annotation for packet reassembly are passed from the Quality Processor to the Output Processor and quality and accounting data is passed to a Channel Controller (DPR via VMEbus). The Output Processor generates instruction codes for the VLSI RAM Controller by converting the piece list from the Quality Processor into instruction addresses using the piece location directory. Any fill or annotation defined by the Quality Processor is passed with an appropriate instruction code to the RAM Controller for multiplexing to the output with the reassembled source packet.

In the prototype Packet Processor, the physical logic interfaces of the microprocessors to the RAMs and special logic are implemented with RAM programmable gate arrays. This interface provides all necessary control signals to the custom logic via a memory mapped logic implementation. In the final version of the Packet Processor this interface logic will be replaced by semi-custom VLSI chips.

SUMMARY

As implemented the data pipeline has a throughput capability of 80 mbits/sec. The overall rate for the present Packet Processor configuration and specified data format is software limited to about 20mbits/sec. This rate can be increased by the substituting faster components as the state-of-the-art advances. Additional improvements may be achieved as the data format definitions become less general and narrow the requirements for adaptability.