REAL-TIME TELEMETRY DATA FORMATTING FOR FLIGHT TEST ANALYSIS

R. Michael O'Brien
Loral Test & Information Systems
P.O. Box 3041
Sarasota, FL 34230

KEYWORDS

Telemetry Processing System, Data Formatting, Flight Test Analysis

ABSTRACT

With today's telemetry systems, an hour-long analog test tape can be digitized in one hour or less. However, the digitized data produced by today's telemetry systems is usually not in a format that can be directly analyzed by the test engineer's analysis tools. The digitized data must be formatted before analysis can begin. The data formatting process can take from one to eight hours depending on the amount of data, the power of the system's host computer, and the complexity of the analysis software's data format. If more than one analysis package is used by the test engineer, the data has to be formatted separately for each package.

Using today's high-speed RISC processors and large memory technology, a real-time Flexible Data Formatter can be added to the Telemetry Front End to perform this formatting function. The Flexible Data Formatter (FDF) allows the telemetry user to program the front-end hardware to output the telemetry test data in a format compatible with the user's analysis software. The FDF can also output multiple data files, each in a different format for supporting multiple analysis packages. This eliminates the file formatting step, thus reducing the time to process the data from each test by a factor of two to nine.

INTRODUCTION

In a typical Telemetry Data Processing System, test data is processed in three steps as shown in Figure 1.

- o Step 1. The vehicle test data samples are digitized to computer disk or digital tape using a Telemetry Front End (TFE) system and a host computer or workstation server.
- o Step 2. The digitized data files are formatted to a file format compatible with the system analysis and report generation software by the system's host computer.
- o Step 3. The formatted data is analyzed and test reports are generated using either workstations or a general-purpose computer.

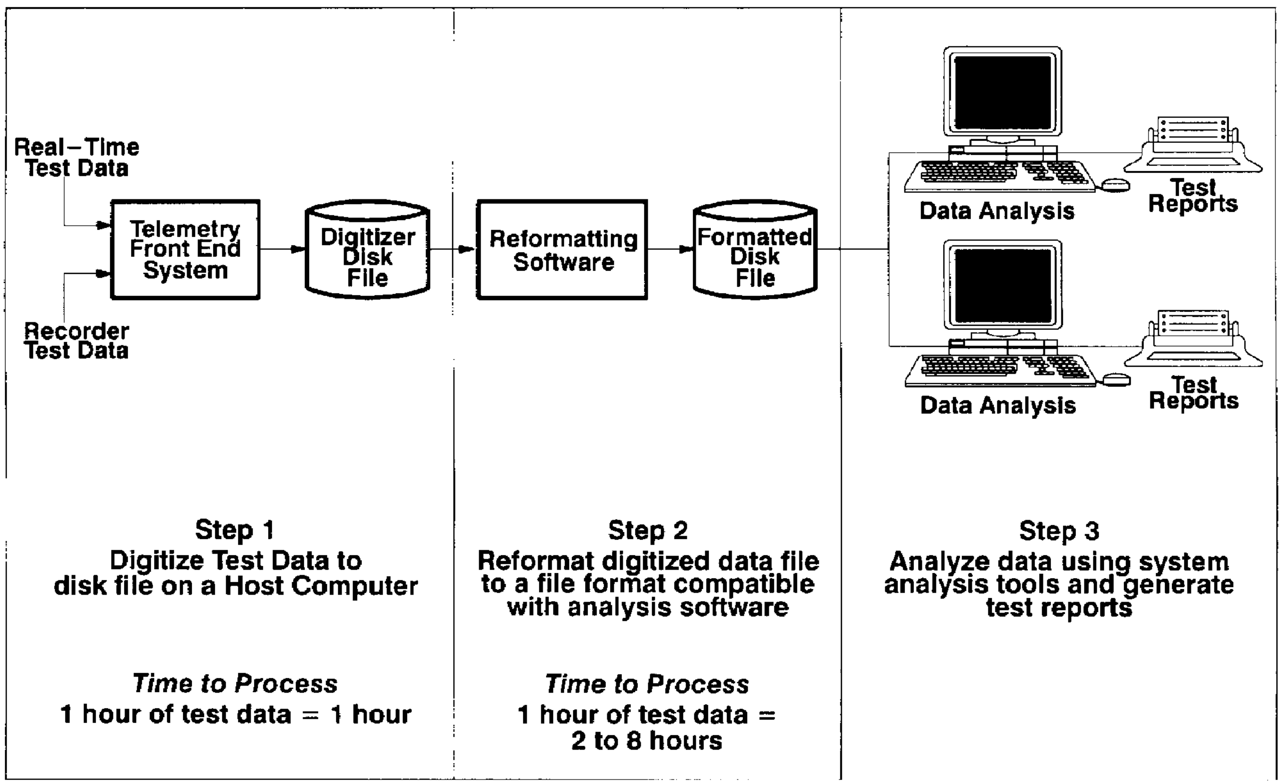


Figure 1. Typical Test Data Processing Procedure

With today's telemetry systems, an hour-long analog test tape can be digitized in one hour or less. However, the digitized data produced by today's telemetry systems is usually not in a format that can be directly analyzed by the test engineer's analysis tools. The digitized data must be formatted before analysis can begin.

The data from most flight tests is used by more than one group of analysts. One group may be analyzing the airframe, another group may be testing avionics systems, while a third group may be examining the weapons systems. Each of these groups probably uses different software analysis tools tailored to its own needs. Each software package requires the data to be in a different format. To support each analysis group, the data has to be formatted separately for each package. Considering that it takes one to eight hours to format an hour's worth of test data, this could add up to a tremendous amount of processing time.

Using today's high-speed RISC processors and large memory technology, a real-time Flexible Data Formatter can be added to the Telemetry Front End to perform this formatting function. The Flexible Data Formatter (FDF) allows the telemetry user to program the front-end hardware to output the telemetry test data in formats compatible with the user's analysis software. To support multiple analysis groups, the FDF can output multiple data files, each in a different format for supporting each group's analysis package. This eliminates the file formatting step, thus reducing the time to process the data from each test by a factor of two to nine.

FUNCTIONAL DESCRIPTION

The Flexible Data Formatter software includes the components shown in Figure 2. The FDF:

- Receives tagged data from the Telemetry Front End.
- Maps and separates the data into output streams (up to 127).
- Assembles the data in each output stream into frames, messages, arrays, etc.
- Formats each data stream into user-defined record formats.
- Accumulates the records for each stream into fixed size blocks for efficient data transfer.
- Stores the data blocks from each stream into separate files that can be distributed over a network.

DATA TO STREAM MAPPING

Each tagged data word sent to the FDF is assigned an FDF output stream number from 0 to 127. The FDF Data Router uses the stream number to route each data word to a user-specified Data Assembler. The stream number is assigned to each data word by the user when the user outputs the data word to the FDF in the Data Processing Unit (DPU) algorithm chain. The stream number 0 is reserved for system time.

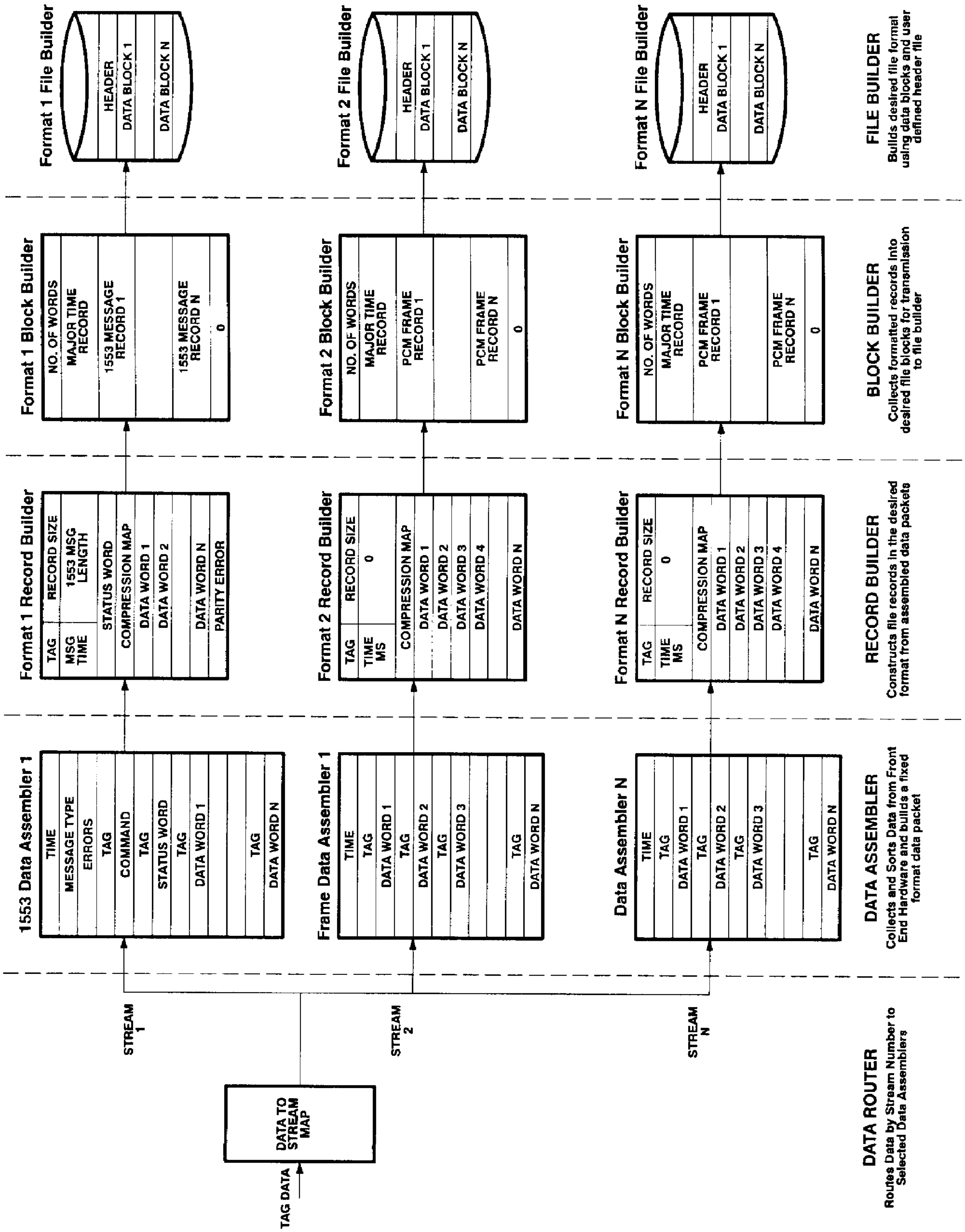


Figure 2. Flexible Data Formatter Functional Process

DATA ASSEMBLER

The FDF Data Assemblers collect and group data prior to formatting. While there are many types of record formats, there are only a few methods of grouping data such as frames, 1553 messages, or arrays. The Data Assemblers recognize the internal 8715 data formats and synchronization flags. The Data Assemblers output the assembled data in a standard, well-defined packet format used by the Record Builders. Since the Data Assemblers output the same packet format, this allows the user to define new record formats without knowing the internal workings of the Telemetry Front End.

RECORD BUILDER

The Record Builders receive assembled data packets and build data records in the format desired by the user. The user can define many different record builders and select which ones the FDF uses to build different file formats.

BLOCK BUILDER

Data is usually stored on disk, tape, or sent over a network. To optimize data transfer rates, the user may choose to create a block of completed data records before transferring the data to a file. For example, If the data is to be stored in files that are on a host computer via an FDDI network, then the Block Builder can build fixed 32 Kbyte blocks of records before transferring the data over the network. This greatly increases the data transfer rate over the FDDI network as opposed to sending one record at a time.

FILE BUILDER

The File Builder stores the blocks of data in files in accordance with the user's format. The File Builder opens and allocates disk space for the data files. The user interface includes a selection of:

1. Disk

2. Directory

3. File Name

4. File Size Allocation

In addition, the File Builder can include a user-specified Header File each time it opens a new file. The user can define up to 127 different File Builders in a single FDF system.

FDF MODULE LIBRARY

The FDF consists of a library of software modules (see Figure 3). There is only one Data Router per FDF. However, there are libraries of Data Assemblers, Record Builders, Block Builders, and File Builders. To build a file for each FDF output stream, the user defines which Data Assembler, Record Builder, Block Builder, and File Builder to use for each data output stream in the FDF Configuration Setup. The output of any Data Assembler can go to any Record Builder in the library; any Record Builder can go to any Block Builder, and so on.

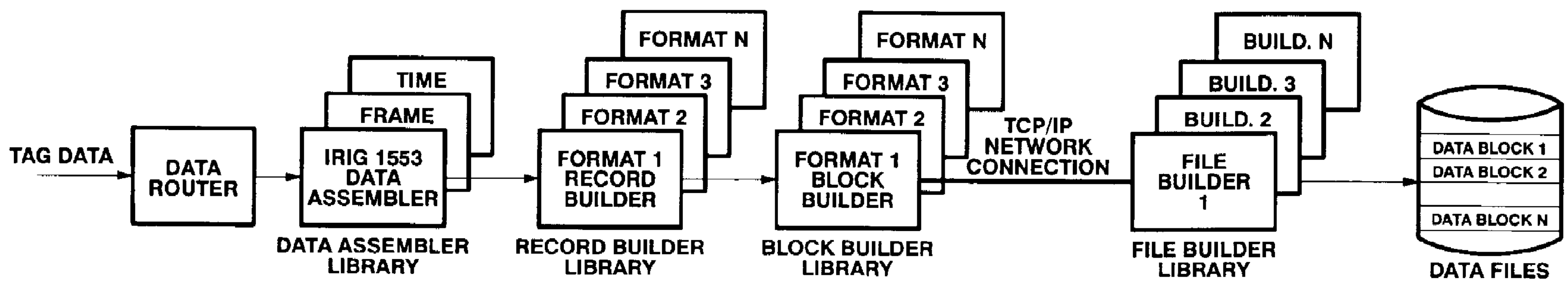
Note that the connection between the Block Builders and the File Builders is a TCP/IP network connection. This connection allows the File Builders to reside in the same hardware as the FDF or be distributed over a high-speed network (such as FDDI or ATM). The user can even mix and match File Builders in a network system. For example, the FDF can store a file in one format in the Telemetry Front End, a different file format in a host computer, and another file format in a workstation all at the same time.

All FDF module and support software is written in standard C. When a user wants to define a new file format, he/she merely writes a new Record Builder, Block Builder, or File Builder; adds it to the library; and recompiles the FDF's processing software.

The Loral OS/90 Telemetry Front End can support up to eight independent FDFs in a system.

MCCLELLAN AIR FORCE BASE FLIGHT DATA ACQUISITION AND PROCESSING FDF ARCHITECTURE

The upgraded McClellan AFB Flight Data Acquisition and Processing System (FDAPS) uses the Flexible Data Formatter to produce data files that are compatible with the Ball Systems Engineering Division's Formatting and Analysis Software Tools (BFAST) package. The FDF formats and stores F-111 flight test data onto four separate files on the system's VAX host computer. Each data type is formatted into a different record type and stored in a different file. The F-111 test data consists of MIL-STD-1553, Mark II, PCM, and PAVETACK data formats.



FLEXIBLE DATA FORMATTER CONFIGURATION SETUP TABLE

STREAM	DATA ASSEMBLER	RECORD BUILDER	BLOCK BUILDER	FILE BUILDER
TIME	TIME 1			
PCM #1	FRAME 1	RECORD BUILDER 1	BLOCK BUILDER #1	FILE #1
PCM #2	FRAME 2	RECORD BUILDER 2	BLOCK BUILDER #1	FILE #1
PCM #3	FRAME 3	RECORD BUILDER 1	BLOCK BUILDER #1	FILE #2
1553 BUS 1	IRIG 1553 #1	RECORD BUILDER 3	BLOCK BUILDER #2	FILE #2
1553 BUS 2	IRIG 1553 #2	RECORD BUILDER 3	BLOCK BUILDER #2	FILE #2
1553 BUS 3	IRIG 1553 #3	RECORD BUILDER 3	BLOCK BUILDER #3	FILE #3
PCM #4	FRAME 4	RECORD BUILDER 4	BLOCK BUILDER #4	FILE #4
ENGINE DATA	FRAME 5	RECORD BUILDER 5	BLOCK BUILDER #5	FILE #5
WEAPONS DATA	FRAME 6	RECORD BUILDER 6	BLOCK BUILDER #6	FILE #5
TSPI DATA	FRAME 7	RECORD BUILDER 7	BLOCK BUILDER #7	FILE #6

Figure 3. Flexible Data Formatter Module Library

The McClellan FDAPS Flexible Data Formatter is configured as shown in Figure 4. All of the FDF modules except the File Builders reside on a Force SPARC2CE processor mounted in an OS/90 Telemetry Front End. The SPARC2CE is configured with an 80 MHZ SPARC2 processor (twice as fast as the standard SPARC2) and 32 MBytes of memory. The processor includes an Ethernet port for setup/control and an FDDI port for data archiving to the VAX 6000.

The FDAPS system is configured for two FDF modules in the OS/90 Telemetry Front End. Each FDF can send archive data to the VAX File Builders over an FDDI network. FDF 1 has a local disk for storing the operating system and the FDF software. FDF 2 boots via the Ethernet from FDF 1's disk on startup.

The FDF produces files in the BFAST format by using special BFAST Record Builders, Block Builders, and File Builders. There is a different Record Builder for each BFAST data record type. The BFAST Block Builders accumulate records into fixed-size 32 Kbyte blocks for transfer over the FDDI network to the VAX. The BFAST File Builders reside on the FDAPS VAX 6000 and build files in the standard VMS format.

For each block stored in the data file, an entry is made into an index file which marks the time and location of that block in the data file. The index files are used by BFAST for retrieving the archived data quickly and efficiently.

The FDF is set up and controlled from the O/S90 workstation via remote login to the FDF Operator Interface Software on the 8715 CPU module. The operator is able to set up the FDF, start/stop/pause data acquisition, and specify disk file information from the O/S90 workstation. All status and error information is displayed on the status display window during operation.

CONCLUSION

Today's hardware processing, networking, and memory technology has made it possible for Telemetry Front Ends that format test data to be directly compatible with analysis software packages used by the test engineer for system evaluation. This eliminates the costly and time-consuming effort of separately formatting test data for each test group analyzing test results. Loral Test & Information System's Flexible Data Formatter is already being used in the McClellan AFB Flight Data Acquisition and Processing System to provide data directly to the Ball Systems Engineering Division's Formatting and Analysis Software Tools package.

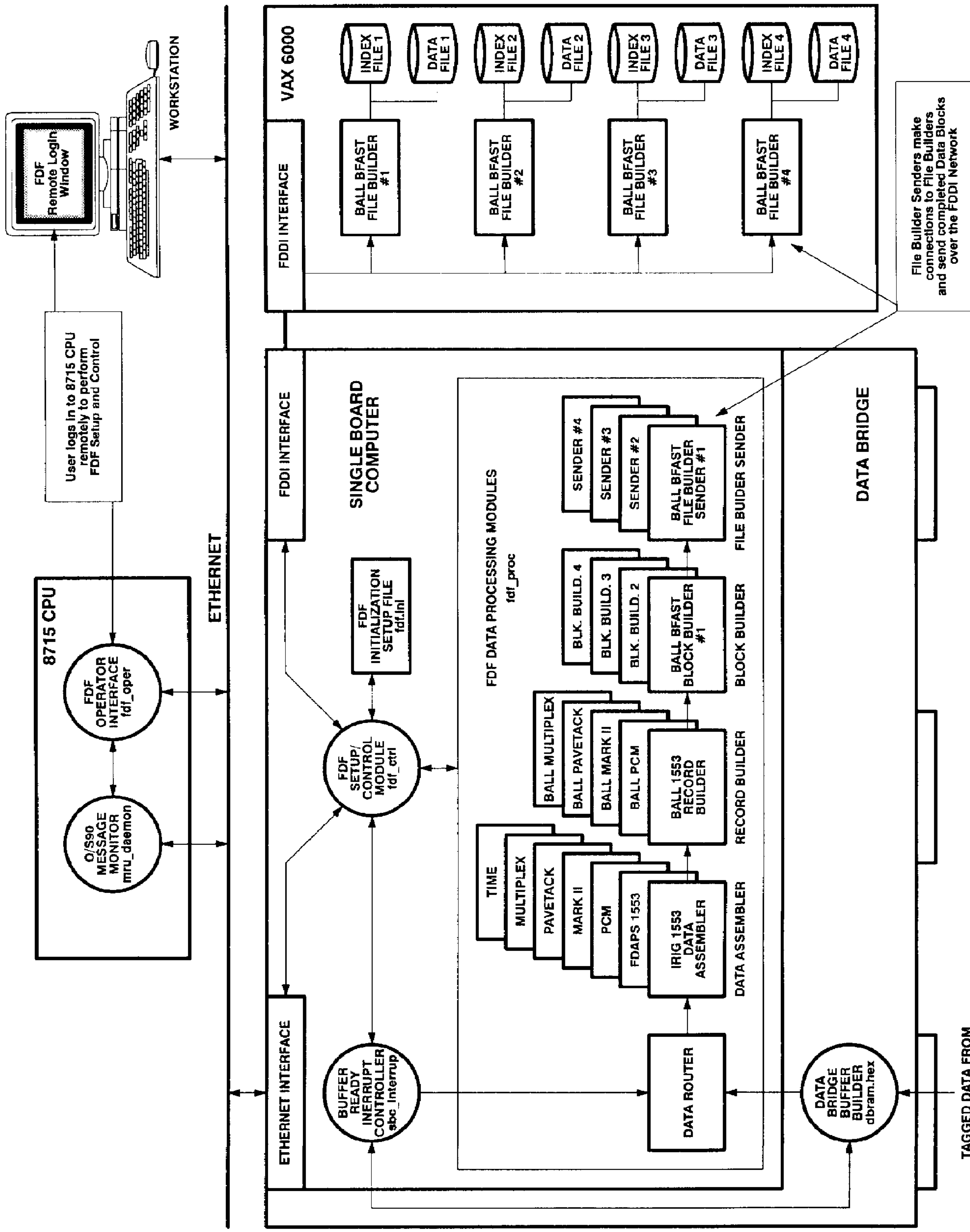


Figure 4. McClellan FDAPS FDF System Architecture