HIGH SPEED SERIAL LINK FOR UA1 MICROPROCESSOR NETWORK

S. Cittolin, CERN, B. Löfstedt, CERN, E. Zurfluh, CERN, Geneva, Switzerland

ABSTRACT

The UA1 data acquisition system consists of a set of distributed microprocessor units. An interprocessor link, independent of the CAMAC data readout, has been developed in order to have continuous remote control and run-time data handling, e.g. transmission of calibration programs/parameters, equipment test/status and histogram accumulation.

The data transmission system is designed to be used in a loop configuration equipped with transceivers for twisted pair cables (RS-422). As an economical system it is running as an ancillary serial loop-link between microprocessors like Data Acquisition Crate Controllers and systems with distributed intelligence.

The software driver consists of a loop-controller package, which may run in a Bambi Computer Language environment and a fully interrupt controlled program for all other secondary stations. A special single-character mode provides a handy link for remote debugging in a pseudo-full-duplex mode.

The format is based on the HDLC protocol without sequence numbering. The Chip MC-6854 from Motorola, Inc. enables an implementation with few components.

INTRODUCTION

The central detector image chamber of the UA1 experiment is read out by a Charge and Time Digitizer unit (CTD). Each crate is equipped with a Dead Out Processor (ROP), which controls 3-5 CTD modules, preprocesses the data and sends them onto the V-Bus to the data acquisition system.

A serial data link connects all 110 Read Out Processors together in a big loop, which is controlled by a Super Caviar Processor or ND-100 (Fig.1). This loop provides the facility to transmit various control commands to the ROP's, to set gains and offsets in the CTD's, to start diagnostic programs, to transfer special programs and to read out RAM areas.

FRAME STRUCTURE, PROTOCOL AND BAUD RATES

The frame structure complies with the international standard "High Level Data Link Procedures - Frame Structure (HDLC)" ISO 3309-1979, including Appendix A and B. The unusual application in the UA1 experiment required a special protocol to get good speed performance and full flexibility.

Hence the protocol depends only on the implemented software, it is always possible to run routines which satisfy the international standard "High Level Data Link Procedures - Element of Procedures (HDLC)" ISO 4335-1979 or other standards like SDL and ADCCP.

All transmissions are in frames and each frame conforms to the format in Fig. 2.

FLAG SEQUENCE  Frame boundaries or idling state
ADDRESS FIELD  Up to 254 secondaries in a loop
CONTROL FIELD  Type of frame
STATUS FIELD  Status information or dummy
PARAMETER FIELD  4 bytes for addresses or command parameters
DATA FIELD  data to be transmitted
DUDDY BYTE  dummy byte
FCS  16-bit error checking sequence
FLAG SEQUENCE  end of frame
The application of a Bit Oriented Protocol (BOP: no sync-characters) results in a small control overhead in the transmission of long sequences. In order to speed up the system and to reduce the number of interrupts in the secondary stations, the 16-bit transfer mode was chosen.

The data transmission rate depends on the implemented options and the processor cycle time. The following rates are jumper selectable.

- 93.75 kbit/s (few options)
- 46.875 kbit/s
- 23.4375 kbit/s (+1%)

Compatible hard-and software is already implemented in the following systems:

- SUPER CAVIAR (converter card)
- Read Out Processor BOP type 296
- General Purpose Microprocessor GPwC
- CAMAC Processor CREP (soon)
- EXORCISE : MEX6854 + interface

The software driver allows to set-up multiprocessor systems with remote control in a transparent mode.

**HARDWARE**

The Hardware is based on the MOTOROLA MC-6854 Advanced Data Link Controller (ADLC), which performs the complex central processor/data communication link. The interface includes a digital PLL clock and receiver/driver which satisfies the RS-422 standard.

Figure 3 shows the functional scheme of the hardware. The signal line is terminated by the resistor R and is connected to the differential input of a RS-422 receiver. RxD is the data input and TxD is the data output of the ADLC. The output line is driven by a RS-422 transmitter.

The ADLC chip needs a special sequence to go synchronous on loop and the electronic switch "I" (controlled by LOC) closes the datapath in an offloop condition. The absolute speed and phase of the datastream is determined by the loop controlling unit, where the transmitter clock TxC and the datasyncronous receiver clock RxC, is split by the electronic switch "2" (controlled by LCD). In all secondary stations RxC and TxC must have the same phase. A digital phase lock loop circuit (PLL) with a free running oscillator (+1%) deviates from the incoming data a local clock with the appropriate phase.

The optional power-off relay keeps the loop closed and therefore operational even some secondary stations are powered off.

The serial signal can be transmitted through telephone type line over distances exceeding 1 km. The same interface can be programmed to run in full duplex mode, as loop controller or as secondary station on a loop.

The coding scheme is set in such a way that there is a signal transition at least after 6 bit-time, independent of idling state, sending only "0" or only "1". The following hardware implemented methods are used:
- Flag Idle (line is never quiet)
- Zero Insertion/Deletion within the frame
- Non-Return-to-Zero-Inverted (NRZI) coding.

**Commands and Functions**

The basic software driver for the loop controller is a set of subroutines. If the program is running in a Bambi environment the commands are executed as Bambi calls, running the appropriate subroutines.

The following Bambi user routines, each with their own set of parameters are implemented:

- LINIT Initialize station as loop controller
- LTX Send data
- LTXA Send data with acknowledge
- LRX Fetch data from a secondary
- LSTAT Acknowledge last Tx and send status
- LRAM Select RAM map
- LCM Execute predefined commands
- LCLR Clear error register
- LMODE Run interpreter (monitor) on a secondary
- LLOAD Load a secondary program into the primary
- LMP Dump a primary program into the secondary
- LRUN Run a Bambi program
- LSTOP Stop a Bambi program
- LRESET Reset Bambi
- LMP Set program counter

Commands which apply to a secondary station:

- LSINIT Initialize station as a secondary
- LSADR Show own address

Once a secondary station is initialized, it runs fully under the control of the ADLC interrupts.

**Software**

a) ADLC and line test routines

A power-up test of the ADLC chip can be made with an internal closed loop. Other test routines provide signals for testing the modules or the full network. (50-180 bytes of 6800 code).

b) Loop controller software

The init routine sets the four control registers of the ADLC and idle flags are sent through the loop. The Bambi interface routines reformat the input parameters, load the control word and calculate the data areas, which eventually go into the frame header.

The routine TXMIT sends the frame, whose length depends on the parameters set through the network. The receiver input of the primary station checks the incoming data, which went through the entire loop, for transmission errors and gives therefore a good indication of the loop performance.
If data or a status has to be fetched the addressed secondary station knows to go
active on loop whenever getting polled by the loop controlling station. The poll sequence
corresponds to a mark idling state sent out by the primary station. A fast synchronisation
PLL is required, because during the polling sequence all stations preceeding the addressed
one (uploop stations) will loose their synchronisation. The fetched data are stored in a
buffer and various checks are executed, e.g. transmission errors and buffer overflows;
a special routine checks if there are more than one secondary station with the same address
on a loop.

In multiprocessor networks (Fig. 4), where also secondary stations have an incorporated
monitor, the CHAR INTERPRETER ROUTINE can set up through the serial line from the primary
station a link to the monitor of a secondary station. This fast single character transfer
mode is seen by a terminal like a full duplex line. This mode is possible with a permanent
"send data" and "fetch data" either with a valid or a dummy character. With a terminal
connected to a primary station all processors in the loop can be served in their usual way.
(main routines 450 to 500 bytes, BAMBI interface routines 450-500 by bytes and interpreter
routines 150-200 bytes of 6800 code).

c) Secondary software

The ADLC chip has an internal First-In First-Out buffer (FIFO) for 3+1 words, and
during the 10-bit transfer mode an interrupt occurs for every two words. In systems which
may run as secondary stations the ADLC interrupt has a high priority and the interrupt
mask is set only for short periods. The two ADLC status registers indicate the interrupt
source. There are separate interrupt bits and service routines for frame bounderies, address
present, frame valid, ready to read data, ready to send data and few more for error conditions.

In each secondary station a RAM window can be defined, into which data from the serial
link can be written. During data reception a continuous check prevents overwriting of
protected areas, but in any case it is possible to read data of the entire processor memory.
A status and error register keeps track of all operations executed by the ADLC. (750-800
bytes of 6800 code).

In the interpreter mode the program remains in the interrupt service routine, and
special BAMBI interface routines are required to convert the full duplex asynchronous
terminal code into an ADLC compatible format (300-350 bytes of 6800 code).

FINAL REMARKS

In the development of the UAI data acquisition system it turned out to be extremely
useful to have a link to the Read Out Processors (ROP), independent of the main data path
(V-Bus). The overall implementation charge of this HIGH SPEED SERIAL LINK is small: less
than 10 chips and about 1.8 kbytes of 6800 code to run ADLC with all options and additional
0.8 kbytes for the BAMBI interface routines.
REFERENCES

7) "EIA Standard PS-422", April 1975.

* * *

Fig. 1 High speed serial link

Fig. 2 HDLC frame structure
Fig. 3  ADLC interface (Block diagram)

Fig. 4  Multi-processor application