Fermi National Accelerator Laboratory

FERMILAB-Conf-99/179

SUMAC: A Monitor and Control Tree for Multi-FPGA Systems

Mingshen Gao
For the BTeV Collaboration

Fermi National Accelerator Laboratory
P.O. Box 500, Batavia, Illinois 60510

July 1999

Published Proceedings of the *11th IEEE NPSS Real Time Conference*,
Santa Fe, New Mexico, June 14-18, 1999

Operated by Universities Research Association Inc. under Contract No. DE-AC02-76CH03000 with the United States Department of Energy

SUMAC: A Monitor and Control Tree for Multi-FPGA Systems

Mingshen Gao
FermiLab, P.O.Box 500, Batavia, IL 60510
For Btev collaboration

The BTeV pixel trigger is a data acquisition system capable of finding tracks and vertices in real time in the BTeV pixel detector array. The trigger uses some 3000 processing elements (DSPs) arranged in three processing levels to handle a raw data rate of nearly 100 Gigabytes per second and bring the trigger rate down to 10 KHz.

The trigger system has more than 6000 programmable elements, including Field Programmable Logic Arrays (FPGAs), microprocessors (DSPs, interface to the monitor and control tree through FPGAs), and others. **Sumac (Serial Utility Monitor and Control tree)** is used for configuring and monitoring of these devices. Its primary function is the downloading of FPGA bit streams, microprocessor programs, chip configurations, and test data. In addition, remote cpus and other devices can send messages and status back to the host. The Sumac system is capable of handling several thousand remote devices from a single host PC. Because it stores configuration data in local flash eeproms, it will be capable of achieving a complete system reboot in less than 1 second.

The Sumac system is a tree hierarchy connected via high-speed serial links. Typically each board in the system will have a control node which accepts a single upstream serial link and fans out to as many as 32 downstream links. The downstream links can connect to FPGAs or to other control nodes for further fanout.

Short links that connect devices on the same board or backplane are implemented as 20 MHz synchronous serial links. Longer links that connect host to backplane are implemented as 12.5 MHz asynchronous serial links.

The serial link sends and receives 33-bit words. Figure 3 shows the three basic word formats. Data words contain user-defined data. Typically, these get placed into a local buffer until a message word is received that tells the software how to deal with them.

Message words contain a 31-bit payload that is defined by the software. Message words cause an interrupt to the CPU. Each message word that is sent must be acknowledged before a new message word can be sent.

Control/Status words are low-level hardware words. Typically, control words are sent downstream and affect the hardware state of the receiver. Status words are typically sent upstream and are used to indicate the current hardware status. They can be sent periodically or under CPU control.

Bit 29 of the Control/Status word is reserved to indicate a message acknowledge. When this bit is received, it clears the BUSY bit, and tells the CPU that it can send another message.

In addition to the 33-bit serial format, each control node can transmit a raw bit stream that can be used to configure the receiving device. Figure 5 shows the flow for configuring an FPGA.

Control nodes can keep data files locally so that the host does not have to re-load them on each reboot. The control node accepts commands to manipulate and download files. It also interprets a simple scripting language so that it can configure an entire board with only a single command from an upstream node.

Fig 1. Tree Structure

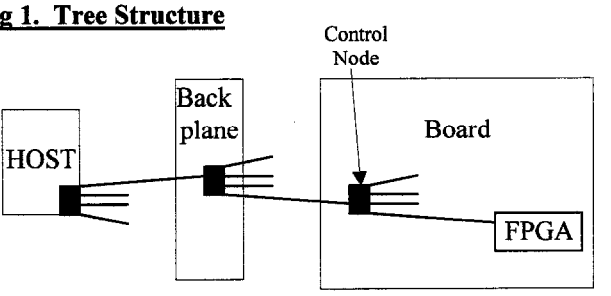


Fig 2. Control Node

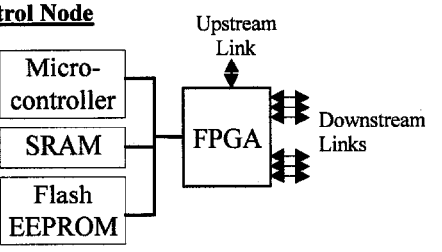


Fig 3. Word Formats

0	32 data bits	Data words get placed in a 2KB buffer.
11	31 Message	Message words cause a CPU interrupt and are interpreted by software.
10	31 Control/Status	Control/Status words affect/reflect hardware status.

Fig 4. Typical message exchange

Sender: Check that BUSY bit for destination link is clear.
Send body of message as data words.
Send message-word to tell the receiver what to do.

Receiver: Get interrupted and read message word.
Save a pointer to the sender's data buffer.
Set up a new buffer for the sender's next message.
Send an ACK to the sender.
Deal with the sender's data.

Fig 5. Bitstream download

Sender: Enable all downstream links that will receive the bit stream. Disable the others.
Send a break-high signal on the links
Send a break-low.
Send a raw configuration stream to the FPGA.

Receiver: A break signal will reset the FPGA and put it in download mode. The bitstrem will set its new configuration.

The control node also accepts commands from downstream nodes to pass messages upstream. This allows remote devices to send occasional text messages to the host to indicate error conditions or status changes. This feature will be accessible to user-level programs running in remote CPUs as printf-like system calls.

Serial Link Encoding

Sumac links that run between boxes use RS-485 differential signaling at 12.5MHz. A serial encoding scheme is needed that embeds a clock reference into the data, and maintains a reasonable DC bias. For Sumac, the 33-bit word is broken into eight "nibbles". The first nibble is 5 bits; the other seven are 4 bits each. The nibbles are then encoded using 5B/6B or 4B/5B NRZI encoding.

NRZI (Non-Return to Zero Inverted) encoding uses a transition to indicate a 1, and the absence of a transition to indicate a 0. The 5B/6B encoding inserts an extra bit into each nibble to insure that there is a transition (1) at least once every 4 bits. The table shows how each combination is encoded. In addition to the 32 data encodings, there are also codings for a FILL word which is sent whenever there is no data to send, and a SYNC word which is sent at link startup or when an error is detected. Any code not in the table is considered an error.

For 4B/5B encoding, a subset of the 5B/6B is used. Only the lowest 5 encoded bits are used to represent a 4-bit nibble.

Encoding Table, 5B to 6B

Nibble	Binary	Encoded
0	00000	010010
1	00001	011010
2	00010	010110
3	00011	011110
4	00100	001001
5	00101	001011
6	00110	001101
7	00111	001111
8	01000	010001
9	01001	010011
A	01010	010101
B	01011	010111
C	01100	011001
D	01101	011011
E	01110	011101
F	01111	011111
10	10000	110010
11	10001	111010
12	10010	110110
13	10011	111110
14	10100	101001
15	10101	101011
16	10110	101101
17	10111	101111
18	11000	110001
19	11001	110011
1A	11010	110101
1B	11011	110111
1C	11100	111001
1D	11101	111011
1E	11110	111101
1F	11111	111111
SYNC	Sync	100000
FILL	Fill	101110

Special states

Break

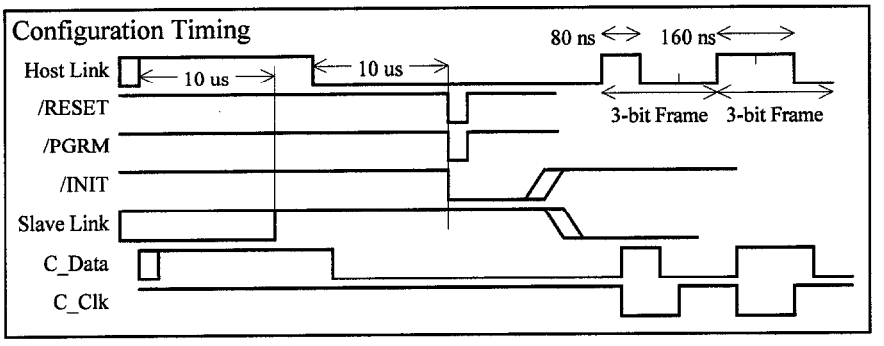
If the Host holds the serial link high for 10 microseconds, the Slave enters Break-High state. In this state, the Slave's control processor is allowed to operate independently and attempt to boot without the host. The Slave responds to a Break-High by driving its own serial link high. This state is also entered if the control-link cable is disconnected.

Reset

If the Host holds the serial link low for 10 microseconds (Break-Low), hardware reset is asserted on the slave node. This will reset the FPGA and Re-start the Slave's control processor. The Slave responds by driving its serial link low.

Config

If the Host sends a Break-High followed by a Break-Low, then the Slave is reset, and its FPGA is cleared and prepared to accept a new bitstream configuration. Each bit of configuration data is sent to the FPGA as a 3-bit frame on the serial link. The frame consists of a high start bit followed by a data bit, followed by a low stop bit. For each frame, the FPGA configuration clock (C_CLK) is cycled once, and a bit is written into the FPGA on the rising edge.

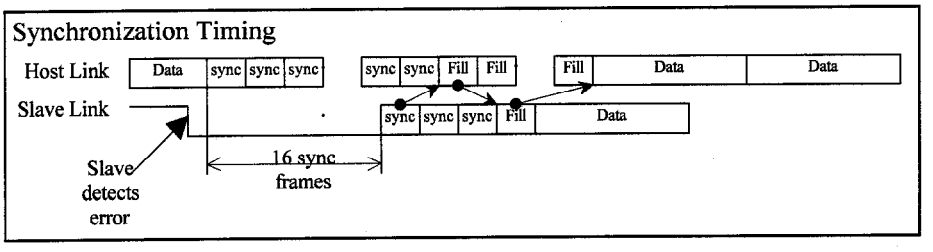


Decoding Map, 6B to 5B

Upper 3 Bits \ Lower 3 bits	000	001	010	011	100	101	110	111
000	-	-	-	-	-	-	-	-
001	-	4	-	5	-	6	-	7
010	-	8	0	9	-	A	1	B
011	-	C	2	D	-	E	3	F
100	Sync	-	-	-	-	-	-	-
101	-	14	-	15	-	16	Fill	17
110	-	18	10	19	-	1A	11	1B
111	-	1C	12	1D	-	1E	13	1F

Sync

If the slave detects an error of any kind, it drives its output link low, and enters the Sync state. If the Host detects this or any other error, it enters the sync state and begins sending Sync words. When the slave has detected 16 contiguous sync words, it begins sending sync words on its output link. When the host detects this, it begins sending Fill words. When the Slave detects Fill words, then it begins sending Fill words. At this point, the links are synchronized, and data words can be sent.



Decoding equations 6B to 5Q

Q0= B0 ? B1 : B3
Q1= B2
Q2= B0 * B3
Q3= B0 * B4
Q4= B5

Encoding Equations 5Q to 6B

B0= Q3 + Q2
B1= /Q3 * /Q2 + Q0
B2= Q1
B3= Q2 + /Q3 * Q0
B4= /Q2 + Q3
B5= Q4