

DSP Prototype for Muon and Pixel Trigger

July 5, 2001

Objectives:

- + Test network connections to several DSP ports. Understand DSP software overhead for each connection. Understand available and achievable bandwidth.
- + Provide a platform to develop and exercise DSP kernel code and start hardware interface driver testing.
- + Provide a platform to develop and test trigger supervisor and monitor (P/MTSM) protocol and messaging. Includes initialization of the DSPs, fault detection and messaging and some hardware event histogramming.
- + Provide a platform to develop and test L1 trigger message protocol.
- + Provide a platform to test segment data flow into DSP.
- + Provide a platform to ground truth simulations of DSP algorithms.
- Provide a platform to run trigger DSP algorithms.

Hardware Features:

- Positions for four DSPs.
 - Mezzanine cards to allow test of different DSPs, especially fixed and floating-point comparison.
 - Additional external program memory per DSP. Boot ROM per DSP.
- FPGA Input and output buffers matched to Pixel Test Adapter card. Input and output buffers connected to DSP DMA ports. Buffer manager in FPGA.
- DSP Host Port connection to ArcNet for PTSM network. Includes embedded microprocessor for message processing, DSP initialization, hardware monitoring and histogram processing.
- DSP buffered serial port (BSP) connected second ArcNet for results network.
- JTAG port(s) for debugging and initial startup of prototype. Connection between PTSM and JTAG to allow in-circuit reprogramming.
- Many DSP general-purpose I/O pins and interrupt pins connect to FPGAs for exploring other control techniques.
- Operator LEDS.

On-the-Prototype-Board Software Tasks:

1. Quantify what data rates are possible on the various links.
 - Data path into and out of the DSPs.
 - Result path out of DSP to ArcNet. (Have the hardware capability to bounce ArcNet packets to baseline the ArcNet hardware so we can separate that time from the DSP overhead.)
 - PTSM link message rates in and out (baseline as B above)
2. Begin to develop message-handling software. Test network interface rates.
 - A simple 16 bit command message to PTSM processor.

- A data message containing header, quantity of data and trailer/checksum.
- ArcNet specific messages for FPGA things like JTAG port control.
- 3. Develop a basic kernel for the DSP test and debug hardware specific code.
 - FLASH memory routines.
 - BSP routines.
 - (HPI) message routines that respond to an interrupt and dispatch based on data stored in a known location by external hardware using the HPI.
 - Initialization of prototype board hardware and firmware loading.
- 4. Add kernel support for application software. Define interconnect between application software and messaging system.

Host computer Tasks:

1. Send and receive simple messages over the ArcNet port. Should these be fully handshaking, i.e. do we require a response from every message? Send data messages to the Prototype and require (?) acknowledgement of a successful transfer.
2. Support downloading of firmware for all prototype devices.
3. Read data file of simulated segment data and load it into the PTA memory. Control the PTA to move the data from the PTA memory into the Prototype board input buffer.
4. Read data from the PTA memory that arrives from the Prototype board output data port.
5. Create a test environment that supports long tests with the possibility of varying the input data, checking for valid responses and collection of some histogramming information.

PTSM Messages Needed:

- 1) Reset/Initialize
 - Application reset/initialize.
 - Hardware initialize (at the ArcNet micro level, at the DSP level and ?).
 - Micro and DSP Firmware download. ArcNet to micro. (Does the DSP need to know?).
 - DSP application initialize, PTSM to DSP.
- 2) Start/Stop application. What messages should come from the application beyond the crossing grade? Possible: every crossing processed produces a PTSM message with the crossing number and a status word that is the beginning of histogramming. Should the kernel also send regular (heartbeat) messages?
- 3) Application mode commands. A set of single word commands that can select different modes in the application.
- 4) Debugging mode commands. A second small set of single word commands to vary the volume of diagnostic messages.
- 5) Histogramming mode commands. Another set that adjusts histogram volume and type of data.

- 6) Error mode commands. For diagnosing problems. Do we dump error information out the ArcNet or through the data output path?

Greg' List of ArcNet Control functions:

H8S/Compact Flash Commands:

ResetH8S()	Hitachi uC reset itself.
GetMbStatus()	Get motherboard status, voltages, temperatures, which DSPs installed, ...
GetDspStatus()	Get status of DSP mezzanine cards.
WriteCfFile()	Write file from host PC to CF card via ArcNet
DeleteCfFile()	Delete file on Cf card from ArcNet.
BootFPGAs()	H8S boots all FPGAs.

DSP Commands:

ResetDSP()	Reset.
HpiBootDSP()	Boot a DSP mezzanine card via HPI port.
JtagBootDSP()	Boot a DSP mezzanine card via the JTAG port.
SetDspBootMode()	Set the bootmode of a DSP.

Global L1 Messages:

Result output contains crossing number and grade.

Suggestion: 32 bits of crossing, 16 bits of result. 0 is not a possible result. Suggestion: A set bit 15 (– numbers) indicates that the DSP recommends a trigger. Low order bits can subdivide the result and provide more detail.

Notes:

The candidate DSP family has DMA, external memory interface, Buffered Serial Ports, Host port and GP I/O ports.

DSP BIOS mostly provided by TI but need to understand how well the canned drivers work and how much application layer sits on top.

P/MTSM is a key factor in initialization, control and fault detection and reaction?

PTA has PCI interface, 4Mbyte memory, and 32-bit parallel port capable of 100 Mbytes per second transfers.

PTSM Microprocessor can “see into DSPs” through host port and track bunch-crossing numbers processed.

Wednesday discussion points:

-- Most of the pixel trigger architecture is data push. The pixel detector pushes data out to the pixel processors, which push into the segment processors. The segment processors push data into the switch, which does the “event building”. The switch pushes events into the DSP farms. The DSPs push results into the GL1 network and the L1 buffers.

--- The HPI can be used to do bunch crossing accounting unobtrusively. The PTSM connection to the DSP farms needs lots of memory so it can do initialization of the DSPs and hold tables of processing parameters.

-- The switch has memory enough to allow it to collect complete crossings worth of data before send it to the DSP farms. This will simplify the DSP input buffer because it will deal with a whole crossing at a time. The crossings need not be in order.