

Test System for Demonstrator Chips

K. Einsweiler

Lawrence Berkeley National Laboratory

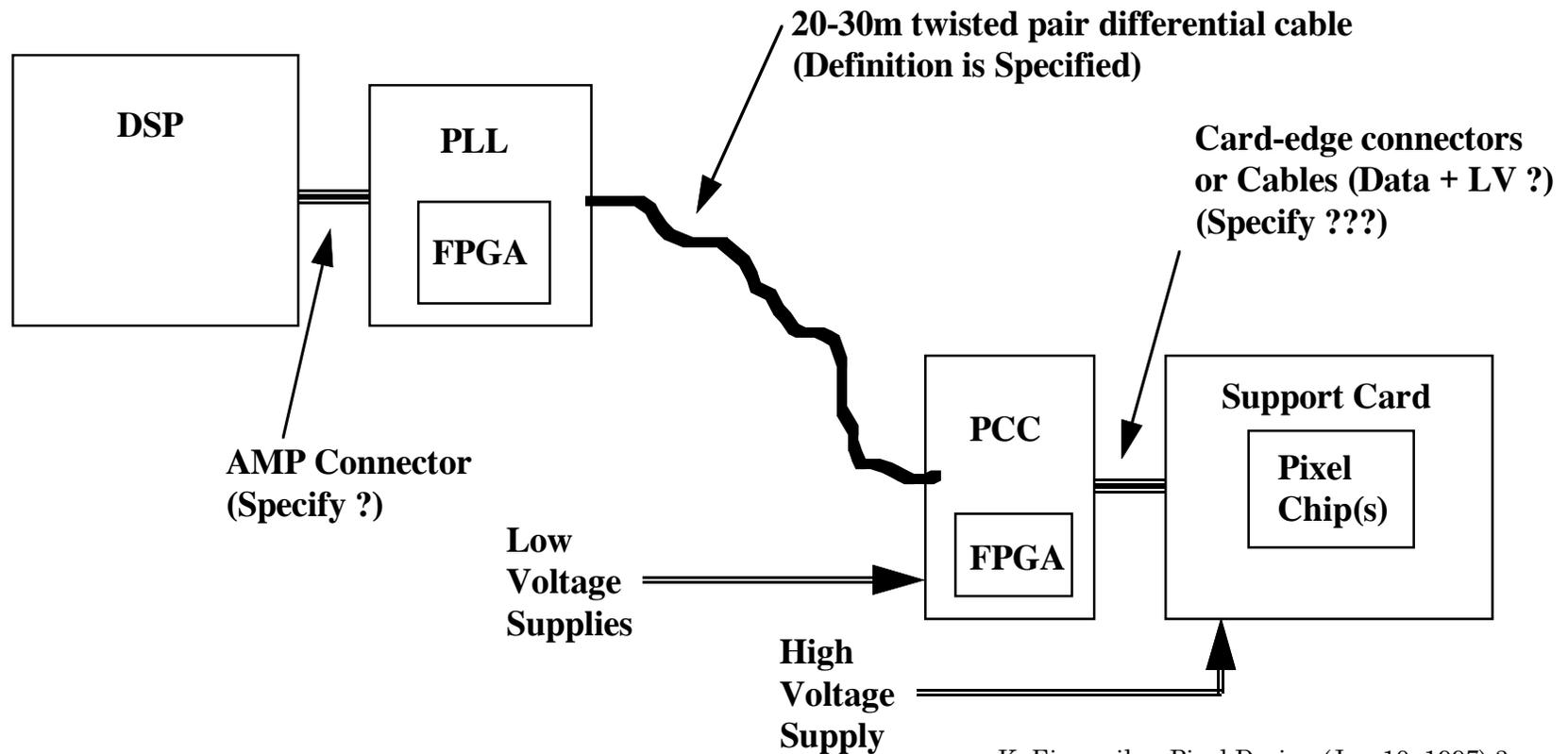
- **Propose overall design for next-generation test system, including partitioning of functions, and solicit comments an input...**

Goals and Requirements

- **Revised test system to support testing of Demonstrator chipset (MCC, FE-A, FE-B) in following modes:**
 - Wafer probing of front-end chips
 - Single-chip testing of front-end chips separately, and hybridized with single-chip detectors
 - Module testing of groups of front-end chips hybridized to module tiles, and connected via an MCC, operating either in transparent mode, or in MCC mode
- **Provide reasonable level of hardware/software standardization across this range of tests in the lab, and also support operation in testbeam environment.**
- **Improve on functional partitioning from present test boards, and move in the direction of a final system design.**
- **Support “common” test environment for two front-end chips, and share the work in a manner that exercises the pixel collaboration. Everyone provides input to overall design, but institutions take responsibilities below:**
 - Tentatively:
 - » LBL/Wisconsin do “DSP end” (PLL and DSP code)
 - » Bonn/Siegen do “chip end” (PCC and Support Card)

Block Diagram

- Major functions concentrated in two new boards:
 - Pixel Low Level Card, mounts on front of DSP as now
 - Pixel Control Card, mounts near pixel system under test
- Additional board to “personalize” and support pixel system under test:
 - Support Card with only passive components (decoupling, etc.), plus possibly a packaged MCC for module testing without a “module hybrid”.



Functional Partition

- **Support Card:**

- Mechanical support for system under test:
 - » For wafer probing, map probe card into PCC (probe card doesn't contain buffering, so support board must attach directly)
 - » For single chip testing, provide connectors for chip-on-board testing
 - » For module, provide wire-bondable bussing to packaged MCC
- Provide local decoupling for relevant input pins of pixel chip(s)
- Provide bias voltage connection (connector, series resistor, decoupling)
- Possibly buffer digital outputs from pixel system for terminated transmission
 - » This proved extremely useful in the past, allowing physical separation of the support board from the PCC board, but it does require implementing active components on the support board...

- **Pixel Control Card:**

- Provide low voltage generation/regulation/monitoring under external control
 - » Ability to accurately control voltage, including setting to zero, and measure current is essential
- Provide “voltage” DACs with current measurement
 - » FE-B: only provide VCAL DAC plus optional external chopper circuit, or input for direct connection of external pulser
- Provide differential drivers/receivers for digital signals from PLL
 - » Include optional means of varying input/output signal levels to check margins ? Probably only for production testing...
- Provide MON pin buffering from front-end chips, with output via terminated Lemo/coax ?
- Possibly provide Clock regeneration, plus local external clock input
- Possibly provide fast-signal resynchronization
- Possibly provide optional, local Strobe/Reset generation
 - » FE-A uses Strobe for diagnostic operation and may want direct control ?
- Possibly provide ”daughterboard” connection to optical link instead of copper
 - » Issue: Encode clock on serial lines (DORIC: biphasic mark encoding) ?
- Possibly provide data framing/formatting to mimic MCC format

- **Pixel Low Level Card:**

- Provide DSP interface to serial command processing in pixel chip(s)
- Provide data decoding (serial→ parallel conversion and frame decoding) for return serial data stream
 - » Possibly also include detailed data unpacking, but this removes flexibility of looking at “raw data”...
- Provide Strobe/Reset signal generation
 - » Simple DSP control of lines is adequate ?
- Provide resynchronization of return data using “return” CKR clock
 - » Over long cable, CK and DCI will be synchronized at PCC, however return data (DTO) will NOT be synchronized at PLL, therefore PCC provides a return clock which should be synchronized with DTO.
- Provide generation of “SSI” serial slow control protocol for PCC
 - » Allows option of adding standard slow-control DACs and ADCs on PCC with transparent control
- Provide data formatting to cope with MCC in either transparent mode or MCC mode:
 - » FE design: event data is formatted/transmitted at “hit” level with single-bit header, configuration data is returned without any formatting.
 - » MCC design: event data is “compressed” (hit-level format re-organized by FE chip using event-level format) and “framed” since it is now variable length. Configuration data has only a single header bit.

Functions in FPGAs

- **Role of FPGA in PCC not yet well-defined, but it is minimal (mainly provides flexibility):**
 - Would allow synchronization of signals and possible format conversions
 - Could be useful for debugging, controlling diagnostic registers in MCC/FE
- **Try to incorporate time-critical operations in PLL FPGA, so that DSP would see a “high-level interface”, consisting mainly of registers (write VHDL not C):**
 - Serial interface registers:
 - » Address + Command (29 bits)
 - » Data count (16 bits)
 - » Data Output (32 bits)
 - » Command/Status for transmission (16 bits)
 - Data output registers:
 - » Cope with up to four data formats (event data from FE, event data from MCC, configuration data from FE, configuration data from FE)
 - » For FE event data, single register storing hit information (25 bits)
 - » For MCC event data, decode into pieces using “Sync” bit (8-bit L1, 8-bit Flag/FE#, 21-bit hit, 21-bit trailer) and store in registers
 - Synchronization of return data with return CKR (small 1-bit FIFO ?)
 - Slow control interface registers

Standard Interconnections

- **Interconnect Cable:**

- Basic MCC connection:
 - » CK (clock), DCI (serial data input), DTO (serial data output)
- Additional return clock for synchronization of DTO and CK and PLL
 - » CKR (return clock)
- Control signals:
 - » STRI (strobe input), RSI (reset input)
- Transparent MCC connection:
 - » TM (transparent mode control)
 - » CCKT (serial input clock), LDT (serial input load), LV1T (L1 trigger), SYNCT (synchronization)
- Slow Control connection (SSI type):
 - » SCK (serial clock), SLD (serial load), SDI (serial data input), SDO (serial data output)
- Possible connections for buffered MON pins (or leave them as separate coax from PCC ?)

- **Simplest configuration has 15 signal pairs, transmitted on shielded twisted pair using ATT LVDS drivers ?**
- **Try to standardize also on PCC/Support Card connection, using standard cable as above plus a LV supply cable ?**

Additional Issues

- **One major question is FPGA choice:**
 - Bonn uses Xilinx chips, programmed using simple PC-based package where “program” is defined via schematic entry
 - » Major advantage is cost and simplicity, and ease of finding parts
 - LBL uses Orca chips, programmed using VHDL and Synopsis under Mentor.
 - » Orca is more powerful chip and runs faster (much easier to code at 40 MHz). Could in principle also generate Xilinx code if necessary.
 - » VHDL is much higher-level and more device-independent coding method
 - » Clearly higher performance, but environment is very complex and expensive
 - Would propose that LBL use Orca for PLL end (significant functionality required at 40 MHz) and Siegen use a Xilinx for PCC end
- **Another issue is “non-DSP” interface support (direct to PC via ?)**
 - Some users would like ability to test without VME crate and DSP...
- **The concept is mature enough to start engineering, we need to get started ASAP !**