A New Design of the CLAS Segment Collector

Brian Eng, Tanest Chinwanawich, Chris Cuevas, and Amrit Yegneswaran
Physics Division, Thomas Jefferson National Accelerator Facility, Newport News, Va 23606
June 11, 2004

This paper details the design and development of a new segment collector card (SCC) for the readout electronics of the CLAS drift chambers.

Each of the thirty-three SCCs accepts 96 hit-bits, whose hit pattern indicates track segments that can be used to generate the Level 2 Trigger. Currently, selectable polar angular ranges are: [8^o, 45^o], [8^o, 90^o], or the entire range [8^o, 142^o].

The new design permits any subset of the 96 hit-bits, ($\Delta\theta$ of each hit-bit is ~1.4^o), to be enabled – presenting 2^{96} options. Novel features of this design are the PICmicro microcontroller (PIC) [1], the Xilinx FPGA [2], and the PCB design.

The PIC, responsible for communications between a PC and all the SCCs and for storing the hit-bit mask and Wire Chamber Test Stand (WCTS) gate in EEPROM, is programmed with Custom Computer Services, Inc.'s C compiler with Windows IDE (PCW), which compiles PICC, similar to C but with additional PIC-only functions.

The PIC code, composed of the functional parts: saving and loading values to/from EEPROM, PIC to computer communication via a serial port, reading/writing characters, writing bits to an entire port (multiple-bit writes), analog-to-digital conversion (A/D), provides, with an area defined as an eight-bit portion of the 96-bit mask, the functions:

- view a single mask for an area: read a mask value from EEPROM and send it to serial port
- get masks for all areas: read all mask values from EEPROM and send them to serial port
- save a mask for an area: get a single value from serial port and save it to EEPROM
- mass save (save masks for all areas): get twelve values from serial port and save them to EEPROM
- get analog measurements: perform A/D and send value(s) to serial port
- write a single mask to FPGA: read a value from EEPROM and write it to port connected to FPGA
- save gate: get value from serial port and write it to EEPROM
- view gate: read a value from EEPROM and send it to serial port
- write gate: read a value from EEPROM and write it to port connected to FPGA
- revert all masks to default value of 255: save default values to EEPROM
- reset masks to defaults on next power-up
- cancel reset on power-up

The FPGA code consists of six modules: clock divider, OR, hit-bit reset, WCTS gate, WCTS trigger, and full mask creator, which perform the functions:

- Clock divider: divides by ten the 50MHz clock frequency used by the FGPA resulting in a 5MHz clock for the PIC.
- OR: the single OR of hit-bits with mask is generated with a bitwise Boolean expression. Since the final OR only has a single operand, it becomes a reduction function and produces a single bit.
- Hit-bit reset: generates a pulse after some delay when a hit-bit is detected.
- WCTS gate: generates gate when there are active hit-bits. Gate delay is based on the value received from the PIC.
- WCTS trigger: generates a 1-clock-cycle-wide pulse when there are active hit-bits.
- Full mask creator: creates the full 96-bit mask from eight-bit sections. The hit-bit mask is sent to the FPGA from the PIC in twelve eight-bit sections; this module assembles them into a single 96-bit value.

The behavioral model of the FPGA modules, which were written in Verilog and built using Xilinx's ISE 5.2i, was simulated with ModelSim.

The PCB design, Fig. 1 shows the SCC prototype, uses an Altium software suite, P-CAD 2002, main components of which are: Schematic – shows inter/intra component connections, PCB – for board layout, and Library Manager – to create needed parts that do not exist in the included libraries.

Required parts library was created by sourcing libraries: included with the program, provided by Jefferson Lab, and by parts designed from the ground up.

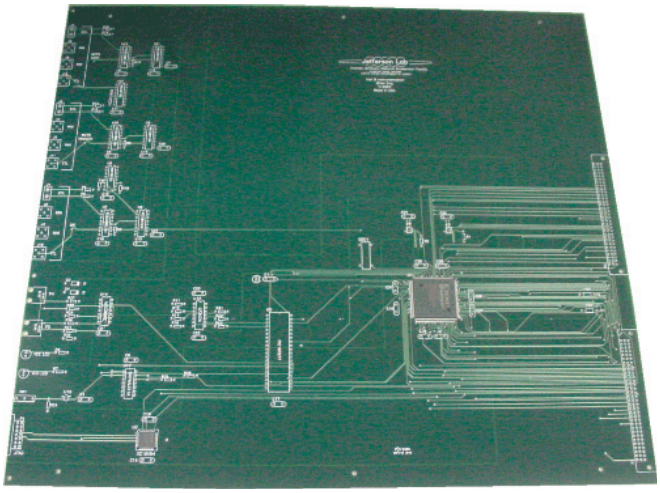

FIG. 1. Prototype SCC

Parts designed from ground up have two facets: Symbol and Pattern. Symbol used in the schematic contains the electrical connections and Pattern used by the PCB represents the physical layout – connections to the board through pads. Both facets are integrated with the Library Executive.

Once the parts library was compiled, a schematic was created from which a net-list, a list of all connections in the design, was generated and loaded into PCB. After the components were arranged on the board, the PCB file was routed using SPECCTRA.

For efficient testing of the PIC code, small sections were checked, and upon verification, were combined and tested – focus was on interfacing the modules. PIC design specifications, with one exception – the serial communications, were met.

The new SCC design requires the thirty-three boards to be networked using RS-485, enabling one host to talk to all the boards simultaneously. Due to expediency, the PIC communications tests used RS-232, which allows a PC to interface directly with the PIC. For the tests, the serial I/O of the PIC was tied to an Analog Devices ADM232A, though the final design calls for a Maxim MAX3086E.

The FPGA code meets specifications. A difference between the Verilog simulated and the implemented-in-the-FPGA code is that the FPGA code inside an initial block does not synthesize, preventing code execution from setting up variables with predefined values. The FPGA needs a few clock cycles before the values reach a known state. The 50MHz clock ensures that the startup delay is $\sim$25 ns.

The PROM [2] reset (RST) that loads new data into the FPGA, which has no internal storage, was to be initiated in three ways: via backplane, a front panel button, or on power-up. Initiating RST via backplane works; the front panel RST button, which would be useful during debugging, does not – because wiring of the button to the PROM was based on the current SCC design, which uses an older model PROM incompatible with the new PROM. While this is a board design and not an FPGA problem, it affects FPGA functionality. Power cycling the relevant SCC crate initiates RST as required.

The PCB has a few problems. Some areas do not execute properly because of design flaws: incorrect wiring, faulty placement – the backplane connectors were placed 2 mm too far into the board – caused by one of the edges on the connector not being labeled in the connector pattern, and pin-out inconsistencies between datasheet and schematic – problem was noticed for an IC. Additionally, there are a few cosmetic problems, such as rotating parts and changing wording on the silkscreen

Placement and pin-out problems require changing either the part's symbol or pattern. Cosmetic problems do not affect the functionality of the SCCs.

In conclusion, the new SCC design meets almost all design specifications. Comprehensive testing needs to be done, for instance, testing of the hit-bits and hit-bit mask. Currently, a manually loaded hit-bit mask has been tested with a single hit-bit active at a time. The next set of tests will change the mask via a program and manipulate multiple hit-bits at a time. However, all possible combinations cannot be tested since both the mask and the hit-bits have 2^{96} combinations each.

[1] Microchip datasheet for PICmicro microcontroller: http://www.microchip.com/download/lit/pline/picmicro/families/16f87x/30292c.pdf

[2] Xilinx datasheets for FPGA and PROM: http://www.xilinx.com/bvdocs/publications/ds001.pdf, http://www.xilinx.com/bvdocs/publications/ds026.pdf