The Track Finder of the CMS First Level Muon Trigger

Alexander Kluge (Alexander.Kluge@cern.ch), Torsten Wildschek (Torsten.Wildschek@cern.ch)

PPE Division

CERN

The track finder receives trigger primitives from drift tubes in the barrel and from cathode strip chambers in the endcap muon system. It assembles those trigger primitives to tracks, assigns transverse momentum, direction and quality to tracks and transmits these data to the global muon trigger. We present the track finder algorithm, hardware implementation and the status of the FPGA prototype.

INPUT

The track finder receives input from two sources:

- DTBX (drift tubes with bunch crossing identification) in the barrel, and
- CSCs (Cathode Strip Chambers) in the endcap.

Either system outputs trigger primitives representing a track segment, which consists of a position, crossing angle and a quality tag.

Trigger primitive generation in the DTBX is described in [1], and the logical interface between chamber trigger logic and track finder in [7].

The CSCs in the endcap have 6 layers of radial strips and tangential wires as sensing elements. Front-end strip and wire cards find track stubs (called LCTs), using a stored sets of valid patterns [3].

The LCTs are passed on the motherboard that performs matching of strip and wire LCTs and selection. The selected LCTs are sent to the portcard that combines several of the endcap chambers (which cover 10 or 20^o in Φ) to a logical chamber with the barrel segmentation of 30^o in Φ [4].

In the endcap the type of data and the logical segmentation of the detector are still under discussion. There will probably be up to three track segments per chamber in either of the two projections, if no segmentation along the radial direction is used.

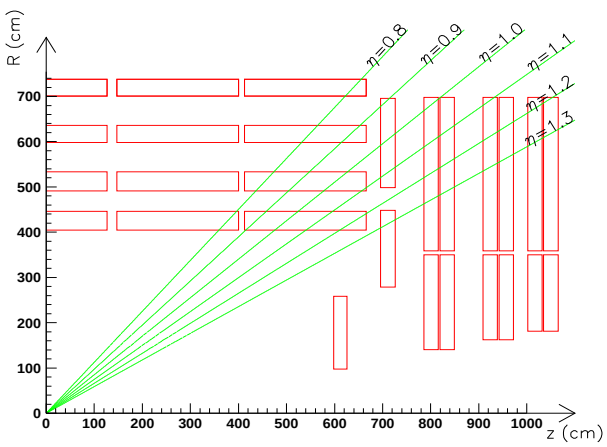
TASK OF THE TRACK FINDER

The task of the track finder is to

- link track segments to tracks,
- assign transverse momentum, direction values (η,Φ), and quality,
- suppress ghosts,
- transfer selected tracks to the global muon trigger.

ALGORITHMS

The algorithm adopted for the track finding processor in the barrel region is described in detail in [8]. The approach to the endcap region is to employ the same algorithms as much as possible. Particularly tricky is the overlap region, where a track crosses both the barrel and the endcap muon system (see following figure).



HARDWARE IMPLEMENTATION

An FPGA-hardware prototype of one sector processor of the barrel track finder has been realized [5]. The goals of the realisation of the FPGA-prototype were:

- to demonstrate that the VHDL-model of the processor can be implemented in hardware with reasonable expense;
- to show that the designed and simulated algorithm also works implemented in hardware and
- to show that the general design concept is feasible.

However, as for economical reasons the XILINX 4000 technology was employed (and not an ASIC) our design aims were not to build a prototype capable of fulfilling timing specifications of the CMS first level trigger [6].

For the hardware implementation of a processor with a large number of input and output (I/O) signals the careful partitioning of processing logic blocks into physical units is essential. FPGAs with an I/O pin count of not more than 192 pins were employed. Considering this restriction we used each I/O pin to insert or extract two data bits to or from the each physical unit within one clock cycle. As the used technology does not allow a synchronously working design with a clock frequency in excess of 50 MHz the internal clock frequency was designed to be 20 MHz. As a consequence the I/O clock frequency yields 40 MHz. It is obvious that this is no option for a final implementation. However, as mentioned earlier, the FPGA-prototype was not designed to fulfil timing specifications of CMS.

In fig. 1 a detailed block diagram of a sector processor is shown. For a description of the processor's functionality refer to [5,7,8]. The three basic steps; extrapolation, track assembling and (p_t-)assignment can be seen and are processed by different physical units. Each cube represents an FPGA or an SRAM based lookup table.

- EXT (extrapolator) block stands for SRAM-based lookup tables. They provide the extrapolation values for each single start track segment of an extrapolation.
- Blocks EU (extrapolation unit) and ERS (extrapolation result selector) conduct the comparison if target track segment have been found fulfilling the extrapolation criteria and output the data in compressed format.
- Blocks TSL (track segment linker), STS (single track selector), TSEL (track selector) and TSR (track segment router) form the track assembling part of the processor. There the track segment pairs are assembled to a complete track and the track segment data is forwarded to the assignment units (ϕAU, ηAU, qAU and pAU).

In all the FPGA processor employs 19 FPGAs and 19 lookup tables. A total of 240000 FGPA gates (only 60 to 70% are used) or 10000 cellular logic blocks are available. 10000 component pins are on the printed circuit board. 1236 input bits are brought onto the board each clock cycle and 362 output bits are given out each cycle. As the board is operated in time multiplexed mode only half the number of I/O pins is necessary, i.e. 618 input pins and 181 output pins. The FPGA-processor evaluates each event within 29 cycles. The printed circuit board is designed in 9U VME standard. However, due to the large number of I/O bits the board is about 15 cm longer than the standard VME board.

As the FPGA-prototype demonstrates, the logic implementation of the designed algorithms does not pose a problem for implementation. The number of necessary logic gates is comparably low. However, the I/O count of the physical units is very high. In the prototype design only a reduced number of bits are processed. For the final implementation a bit number of some 2500 has to be processed each cycle in each sector processor. Even worse discussions are ongoing to increase the number of data provided by the drift tube trigger primitive generator[1][2].

There are several options to come by the problem. One is transmitting a group of track segments sequentially with 80 MHz. Another is transmitting in a bit serial format using an optical link receiver directly on the board. However, this problem is not solved yet.

A study employing application specific gate arrays (ASIC) has been conducted to prove the feasibility of an implementation of the track finder processor system with today's technology fulfilling all specifications given by CMS. Detailed results can be seen in [5].

The general concept of employing ASIC is to increase I/O clock frequencies to 80 MHz allowing an operating frequency of 40 MHz. Depending on the partition scheme data are processed in a pure pipelined scheme or a mixed architecture. In the latter case several cycles are used to read data into a physical unit in order to process the entire data set in the subsequent cycles. Certainly a number of parallel units have to be used to cope with data arriving each cycle. The scheme allows to reduce the necessary number of I/O pins for the processing unit on the board.

The ASIC implementation study reveals that timing constraints of the track finder processor can be met by an ASIC based system. In the study the virtual ASIC-implementation employed the already outdated technologies Motorola H4C 0.7 µm and ES2 0.7 µm. The ASIC based implementation can process the algorithm within eleven to 14 cycles (depending on partition

schemes). The CMS first level trigger latency plan [6] reserves 14 cycles for the track finder processor (not including the final sorting for the global muon trigger).

However, the developments of field programmable gate arrays FPGAs should be carefully watched. As only a low number of physical units will be necessary for the final implementation economically seen the use of FPGAs is the first choice (in the barrel are only 60 sector processors). Of course the reprogrammability of the FPGAs would open tremendous design opportunities.

Industry [9] announces FPGAs for the end of 1998 containing 320 kGates, 14212 registers, 185 kbit RAM, and four input lookup tables with 1.7 ns delay in packages with an I/O count of up to 608 pins. The propagation delay from a register to the output is supposed to be lower than 6 ns. This would enable a data transmission between FPGAs of faster than 80 MHz.

As a conclusion one can say that the track finder processor could be implemented today. However, in order to find the most efficient solution it is vital to search for the most appropriate implementation technology.

REFERENCES

[1] M. de Giorgi et al., Design and Simulation of the Trigger Electronics for the CMS Barrel Muon Chambers, CMS TN/95-01

[2] M. De Giorgi et al., Design and Simulations of the Trigger Electronics for the CMS Barrel Muon Chambers, Proceedings of the First Workshop on Electronics for LHC Experiments, CERN/LHCC/95-56, p. 222-227.

[3] J. Hauser, Baseline Design for the CSC-based Endcap Muon Trigger, CMS TN/95-013

[4] P. Padley, private communication and WWW site "http://bonner-ntserver.rice.edu/motherboard/"

[5] A. Kluge, The Hardware Track Finder Processor in CMS at CERN, Dissertation at the Technical University of Vienna, July 1997

[6] A. Kluge, W. Smith, CMS Level 1 Trigger Latency, CMS TN/96-33, CERN, 8 March 1996

[7] A. Kluge, T. Wildschek, Track Finding Processor in the DTBX-Based CMS Barrel Muon Trigger, Proceedings of the First Workshop on Electronics for LHC Experiments, p. 228, CERN/LHCC/95-56

[8] A. Kluge, T. Wildschek, Track Finding Processor in the DTBX Based CMS Barrel Muon Trigger, Proceedings of the Second Workshop on Electronics for LHC Experiments, p. 319, CERN/LHCC/96-39

[9] Lucent Technologies, Optimized Reconfigurable Cell Array (ORCA) OR3Cxxx/ OR3Txxx Series Field Programmable Gate Arrays, Advance Product Brief, October 1996

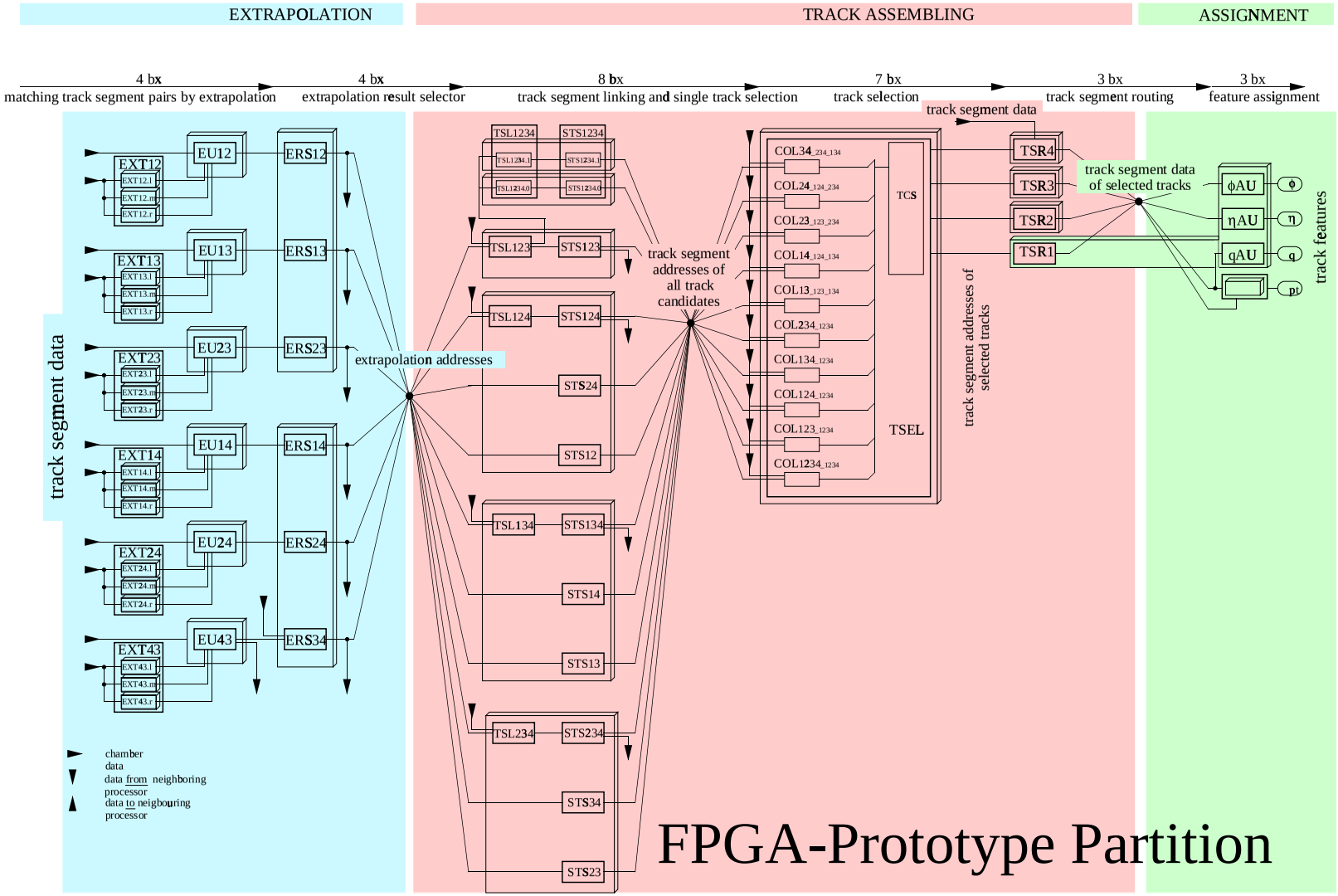


Fig. 1.: Block diagram of a sector processor. Each cube stands for a physical unit (FPGA or SRAM). Each rectangle stands for a logical block.