Available on CMS information server

CMS NOTE 1997/093

The Compact Muon Solenoid Experiment

CMS Note

Mailing address: CMS CERN, CH-1211 GENEVA 23, Switzerland

The Hardware Muon Trigger Track Finder Processor in CMS Prototype and Final Implementation

A. Kluge, T. Wildschek

CERN, Geneva

Institute of High Energy Physics, Vienna

Abstract

The paper discusses the prototype using FPGAs and the options for final implementation of the muon trigger track finder processor. Integration problems for both the prototype and the final implementation are dicussed. Several final implementation studies are introduced. The trigger is based on data from the drift tube muon chambers. The task of the processor is to identify muons and measure their transverse momentum p_t. Data of more than two hundred thousand drift cells are used to determine the location of muons and measure their transverse momentum.

The Hardware Muon Track Finder Processor in CMS - Prototype and Final Implementation

INTRODUCTION

This note describes the prototype of the muon track finder processor and possible final implementation schemes. For information about the system itsself refer to the literature. Hardware specification and method of the track finder processor is described in [1]. System architecture and a description of the algorithm are given in [2].

FPGA PROTOTYPE LAYOUT AND TEST

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 [3].

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 fulfill timing specifications of CMS.

In fig. 2 a detailed block diagram of a sector processor is shown. 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 whether target track segments 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 are forwarded to the assignment units (ϕAU, ηAU, qAU and pAU).

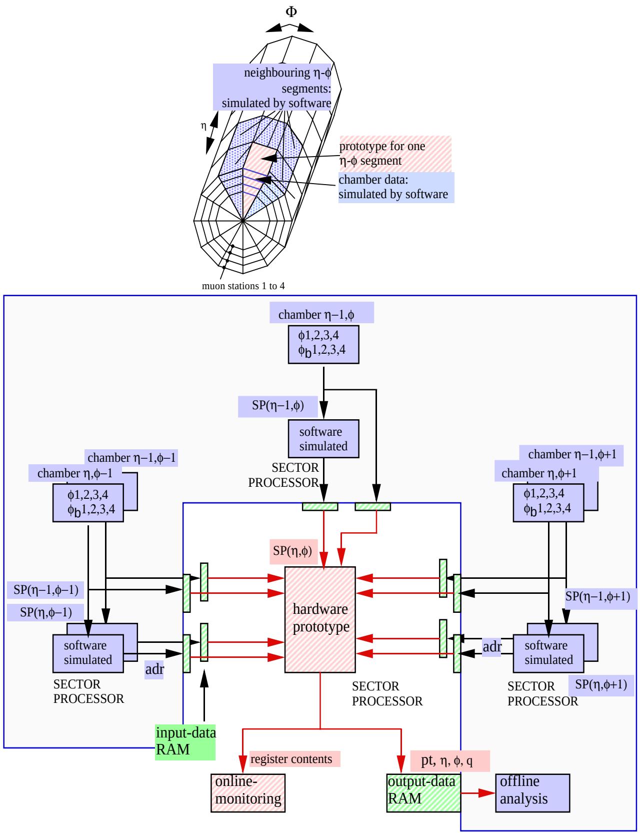

Fig. 1.: Test configuration of the FPGA-prototype.

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 is 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 amount of data provided by the drift tube trigger primitive generator.

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.

IMPLEMENTATION STRATEGY

The task of system partitioning had to be done by hand. For programming the FPGA automated translation tools were employed. The VHDL-code was translated directly to the gate array level. However, it has to be stated that this procedure was only possible for non time critical functions. Time critical functions were translated manually to the gate level.

The total execution time is 29 clock cycles. However a large fraction of the processing steps is consumed by the transmission between the integrated circuits. At least one clock cycle is lost for each transmission. The processing time without transmission time is only 20 cycles. This is the number of clock cycles for an operating frequency of 20 MHz. In order to achieve this frequency, additional pipeline steps (flip-flop stages) had to be introduced. This was not due to slow processing of the logic units but merely due to routing delays between the units. Obviously one can achieve a smaller number of processing steps by reducing the operating frequency and removing a given number of flip-flop stages.

The design of the prototype board showed clearly that FPGAs of today are not suited to the track finder requirements. Both the I/O count of the packages and the data propagation do not allow the final design of the track finder system employing today’s FPGA.

However, the functionality of the implementation of the track finder, the momentum measurement algorithm and the hardware structure mapping of the detector geometry onto the hardware level was proven to be adequate to the system requirements. Moreover it could be shown clearly that the VHDL-model can be implemented in hardware with reasonable hardware expense.

TEST CONFIGURATION

Each sector processor must communicate with its neighbours. However, only one track finder processor prototype is available. Neighbour processors are simulated logically using their VHDL-representation. In fig. 1 the test setup is shown. The hardware prototype receives test patterns from the test bench. Simulated chamber data as well as some simulated neighbour processor data are loaded at the same time.

For test pattern procurement a two-step simulation is run. Firstly detector data simulated by generating physics events [4] are processed using the VHDL model. Running the VHDL-simulation provides two more datasets. The FPGA prototype test requires all three datasets, namely input pattern data, neighbour processor data, and output data. The first two, containing data from up to 10^5 bunch crossings are stored off-line in the input partition of the test bench memory. The test bench transfers detector data and inter-processor data to the FPGA-prototype using a synchronously clocked transfer. The transfer part plus the data processing by the FPGA-prototype is what may be called the actual test at full speed. The output is stored in the output partition of the test bench memory. Comparison with the VHDL simulation data is done off-line after the actual hardware test.

The trigger test requires provision of test patterns with a data width of 1236 bits. This number includes chamber data as well as data coming from neighbouring processors (38 track segments each 22 bits; 440 bits of intermediate results from the neighbour processors). Input data passing to the processor are time multiplexed. Test patterns need to be sent in two parts. Pattern generators with a width of 618 bits are required for transmitting data using a 40 MHz clock. A flexible and versatile test system was conceived. 32-bit wide fast dual port memories [5] which can be used at frequencies up to 100 MHz on one port were employed. Multiplexer boards are fitted between the memories and the FPGA-prototype in order to be able to double the number of bits. The multiplexer boards are custom-designed for use with the fast dual port memories.

The dual port memories may conveniently be loaded via their second port which is a full VME-interface. A common clock signal allows synchronous transfer using the second port, which also triggers the output sequence of the test patterns.

Output data of the FPGA-prototype as well as some intermediate results for error detection are read out using another set of multiplexers and dual port memories. In total the prototype delivers 362 bits per processed event. After processing all test patterns the FPGA-prototype processor’s output is read to the work station via the VME-link and analysed off-line.

FEASIBILITY STUDY FOR ASIC BASED IMPLEMENTATION OF THE CMS TRACK FINDER PROCESSOR

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 [6]. Several partition schemes have been evaluated.

The general concept of employing ASICs is to increase I/O clock frequencies to 80 MHz allowing an operating frequency of 40 MHz. The first ASIC-partition scheme resembles the one used for the FPGA-prototype and is illustrated in fig. 3. The

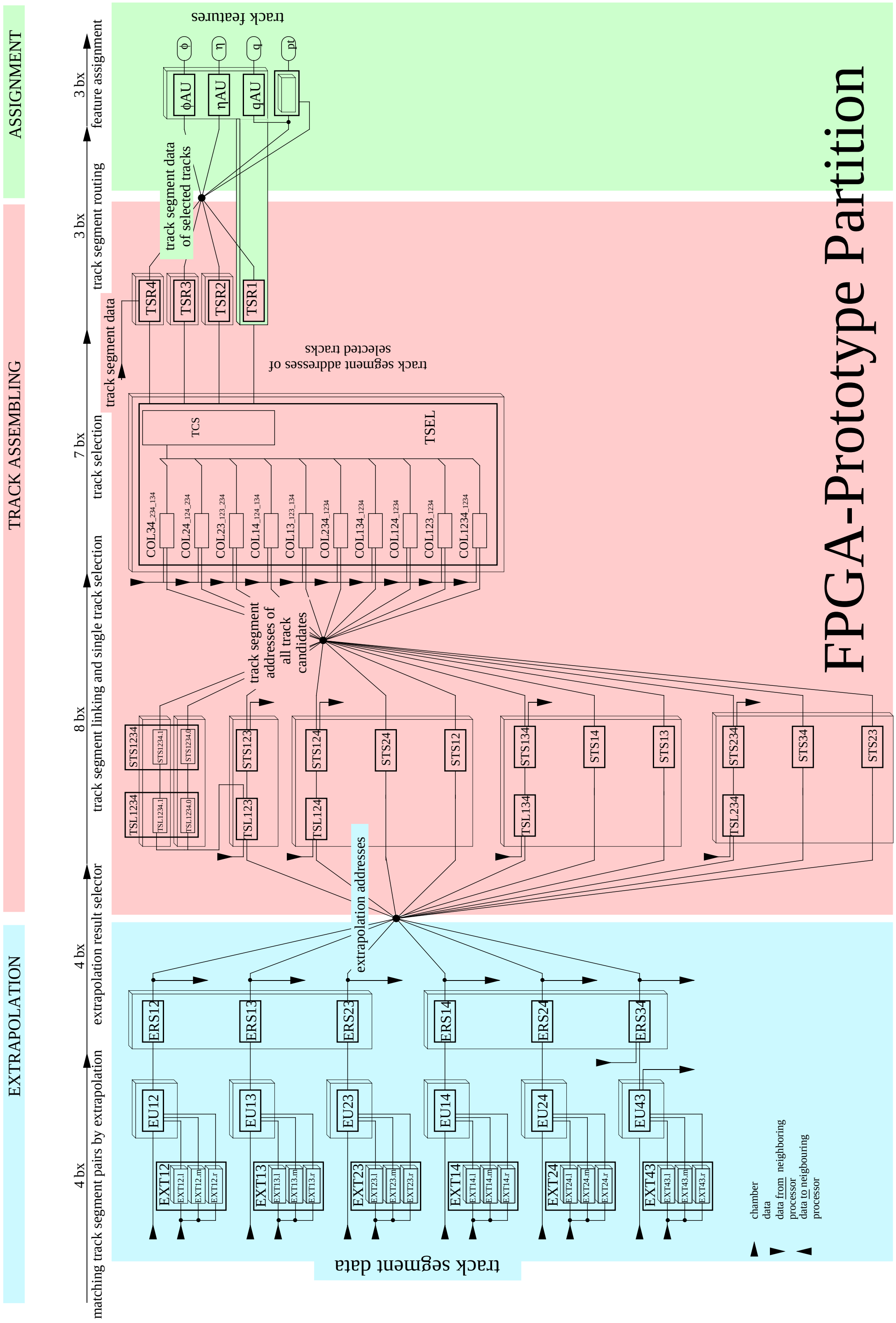

Fig. 2.: Block diagram of a sector processor. The shaded boxes represent the processing steps (extrapolation, track assembling and assignment units). The small boxes stand for a physical unit (FPGA and RAM).,

virtual ASIC implementation employed the already outdated gate array technology Motorola H4C 0.7 µm [7].

Fig. 3 shows the block diagram of the track finder processor and the system partitioning. Each grey box stands for an integrated circuit. Two ASIC types, a number of memory based lookup tables and field programmable gate arrays FPGAs are employed:

Table 1 illustrates the properties of the employed units.

(1) The input block IN prepares the data to be input into the ASIC (time multiplexing). Standard line drivers are used.
(2) The extrapolation value calculation EXT is performed using SRAM-based lookup tables.
(3) The extrapolation-ASICs EU-ASIC, containing the extrapolation units EU and the extrapolation result selectors ERS for a station pair, perform the extrapolation comparisons and give out the two addresses of matched target track segments.
(4) The track assembler-ASIC TA-ASIC, containing all track segment linker TSL, single track selectors STS, cancellation units COL and track class selector TCS, assembles matched track segment pairs and outputs the relative address of the track segments of the found tracks.
The extrapolation from station four to three works in the opposite direction with respect to all other extrapolations. The extrapolation selector ERS34 must find the two track segments in station four TS4 which are the target of the extrapolation from station three. Extrapolation results from the neighboring segments must be combined in the processor and evaluated in an extrapolation result selector ERS34. In order to reduce inter-chip transmission ERS34 is located within the track assembler-ASIC TA-ASIC.
(5) Track segment routing TSR employs field programmable gate arrays.
(6) All assignment units, except the p_t-assignment unit PAU, fit into field programmable gate arrays.
(7) The p_t-assignment unit PAU is realized using a SRAM based lookup table.
(8) IN-EU-neigh., OUT-EU-neigh., IN-TA-neigh., OUT-TA-neigh. stand for the I/O units for the data transfer between sector processors of different detector segments.

The timing specification for the entire system including the wheel sorter requires the execution to be completed within 21 bunch crossings (bx = 25 ns). The wheel sorter needs up to seven bx. As a consequence 14 bx are left for the track finder processor itself. Table 1 shows that the ASIC-based system can accomplish this requirement.

As can be seen in table 1 the gate count does not pose a problem to the integrated circuits. However, the I/O pin count is high in some units. As already mentioned earlier, the problem may be solved by multiplexing input pins with an integer multiple of the chip's operating frequency. If all data are input into and output from the integrated circuit within one bunch crossing, one cycle latency is added by the I/O process. The necessary multiplexing factor m_{IO} may be derived from the fraction of the number of input bits n_{input} and available I/O pin count p_{IO}. The resulting I/O bit rate b_{IO} is the product of multiplexing factor $m_{I/O}$ and the operating frequency f_{op} (equations 1. and 2).

$$m_{IO} = \frac{n_{input}}{p_{IO}} \qquad \text{(Eqn. 1.)}$$

$$b_{IO} = m_{IO} \cdot f_{op} \qquad \text{(Eqn. 2.)}$$

The input synchronization and buffer stages IN have to prepare the processor's input data for insertion into the ASIC. Compensation for the phase shift between the input channels must also be done. Although a first compensation step is already performed by the transmission unit from the detector to the control room the input stage of the track finder processor must be able to shift the phase of the incoming signals up to 0.5 bx (12.5 ns) relative to each other. Input data is arriving at the control room at a certain time after the according bunch crossing. The full dataset arrives in parallel format at a rate of 40 MHz. Input transfer to the ASICs is done time multiplexed. A total latency of 1 bx (25 ns) is foreseen for all input stages IN. If the way of transmitting the data from the detector to the track finder processor is adapted in a way that the necessary data format is provided already by the transmitting stage one half of a bunch crossing can be gained.

All extrapolation lookup tables EXT are designed using SRAMs of the size 256 x 18 bits. Employing SRAMs with an access time of less than 8.5 ns allows the design of one EXT unit using a single RAM for two source track segments. The EXT unit is responsible for providing extrapolation values of two track segments (one after the other) within one bx. Such devices are available today [8]. The access time of the extrapolation lookup tables EXT does not contribute to the total latency because EXT works in parallel with the extrapolation ASIC.

The extrapolation-ASIC EU-ASIC must have a maximum of 402 input bits (considering the most complex case of nine neighbouring detector segments). Applying a multiplexing factor m_{IO} of two gives an I/O pin count of 201 pins and an I/O bit rate b_{IO} of 80 MHz. Using a modern ASIC technology these requirements can easily be met. The execution time corresponds to three bunch crossings (75 ns).

The track assembler-ASIC TA-ASIC must have 942 input bits, again considering nine neighbouring detector segments. Applying a multiplexing factor m_{IO} of two yields an I/O pin count of 471 pins with a I/O bit rate b_{IO} of 80 MHz. By today's standard such an I/O count is technical feasible but at very high cost. One option is to increase the multiplexing factor m_{IO} to three which gives 314 I/O pins and an I/O bit rate of 120 MHz. Although an I/O bit rate of 120 MHz is fully feasible some

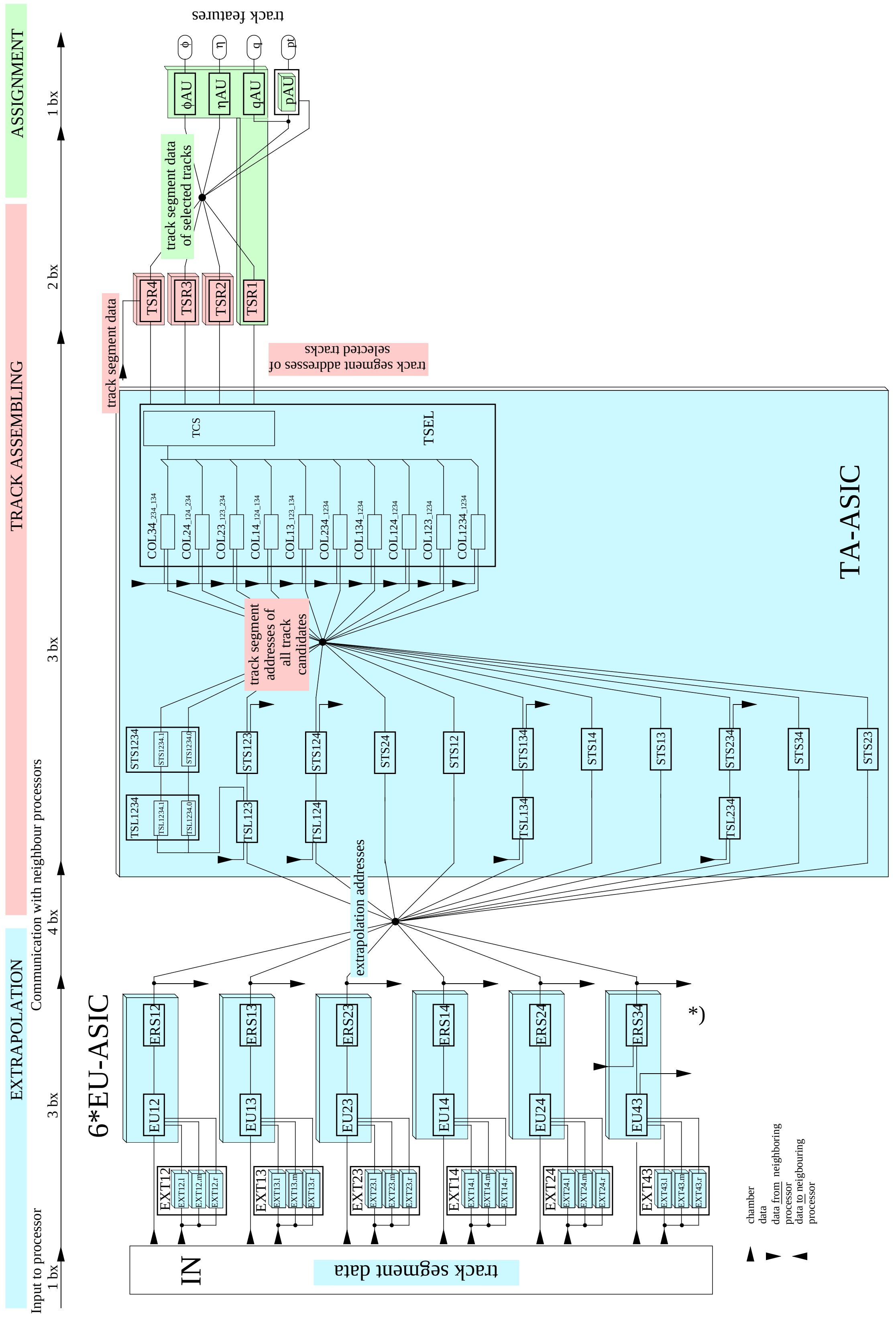

*Fig. 3.: ASIC-system partition. *)...see point (4) in text.*

unit name	type	size	I/O count for 9 neighboring detector segments	I/O count for 6 neighboring detector segments	processing time typical, PTV=1.0 25°, supply=5.0 V	processing time worst case, PTV=1.49 70°, supply=4.5 V	latency contribution in bx
IN-chamber	Latch, MUX	-	1484	1004			1
EXT	SRAM	256 x 18	-	-	8.5 ns	8.5 ns	$0^{(*)}$
EU-ASIC	ASIC	9000 gates	402	286	23.8 ns	35.5 ns	$3^{(**)}$
IN-EU-neigh.	Latch, MUX	-	700	408			1
OUT-EU-neigh.	Latch, DEMUX	-	120	88			1
IN-TA-neigh.	Latch, MUX	-	12	12			1
OUT-TA-neigh.	Latch, DEMUX	-	120	120			1
TA-ASIC	ASIC	32000 gates	942	626	23.9 ns	35.6 ns	$3^{(**)}$
TSR	FPGA	6 x 2 x 22 = 264 bit RAM	406	272			2
AU except PAU	FPGA	8000 gates	262	250			1
PAU	SRAM	128k x 8	-	-	10 ns	10 ns	
OUT-track finder	Latch	-	38	38			0(***)
total	VME board	9U	2474	1670	-	-	14

Table 1: : Properties of ASIC based system parts,
$^{(*)}$ does not add to latency, because EXT works parallel to EU-ASIC
$^{(**)}$ Latency is calculated for an I/O-frequency of 80 MHz (multiplexing factor $m_{I/O}$ = 2)
(***) Transmission and synchronisation included in wheel sorter latency.

additional complexity is added when using an odd multiple of the clock. It has to be noted that the increased transfer frequency would only apply on the board level. The transmission of data from neighbouring processors onto different boards may be performed at a lower rate.

However, there still is a solution to further reduce the necessary number of input bits (and pins) at the track assembler-ASIC input. Up to now track assembling was considered progressing only in one direction, namely from the inside to the outside of the detector. Obviously, when starting in the centre wheel particles may diverge in both z-directions, making it necessary to assemble track segments taking into account nine neighbouring detector segments. However, when performing the track assembling from the outside towards the inside of the detector the possible z-direction of the tracks is determined and hence only six neighbouring detector segments must be taken into account; in the centre wheel only three. In addition background noise in the chambers of station four is considerably lower because punchthrough [9] of particles other than muons is smaller in this region.

The drawback of outside-to-inside algorithms for track assembly is a more complicated single track selection STS. Anyway, the decision in which direction the tracks will be assembled must be postponed until more detailed simulation results of the cathode strip chambers in the forward region will be available. It might as well be possible that the requirements regarding the forward and overlap region between drift tubes DT and cathode strip chambers CSC completely rule out one track assembling direction because of chamber geometry. Anyway, the track assembling-ASIC needs only 626 input bits for six neighbouring detector segments. A multiplexing factor m_{io} of two is sufficient to get done with 313 input pins and an I/O bit rate of 80 MHz. The process is terminated within three bunch crossings (75 ns) in any case.

The track segment router TSR may be realized employing field programmable gate arrays FPGA. The necessary I/O bit count of 406 and 272 respectively cannot be implemented using FPGAs with a clock frequency of 40 MHz. However, the task of a track segment router can be distributed to two parallel units, each having only one half of the inputs. The outputs may be combined employing tri-state buffers introducing almost no additional latency. A track segment router latency of two bx (50 ns) may be achieved.

The assignment unit AU, except the p_t-assignment unit PAU, is designed using two sets of two parallel FPGAs. Data I/O (up to 262 bits) can be split up between the FPGAs.

The p_t-assignment unit PAU is designed by a SRAM of the size 128k times 8 bits. Using a RAM with an access time of 8.5 ns allows to use only one RAM to assign the momentum to two tracks within one bx. Within one bx the assignment process is terminated.

The total latency including input and output buffers plus synchronization yields 14 bunch crossings or 350 ns. Adding seven bunch crossings for the wheel sorter and synchronisation yields 21 bunch crossings for the entire ASIC-built track finder system. This corresponds exactly to the maximum value. Employing ASICs for the track segment routers TSR and assignment units AU instead of FPGAs further reduces the processing time. Removing all FPGAs from the design and replacing them by ASICs is a possibility to reduce the I/O bit rate for the track assembler-ASIC. It then becomes possible to sacrifice a bunch crossing to read in data into the chip at a lower frequency (e.g. at 80 MHz). However, this would require two track assembling-ASIC to work in parallel.

The gate count for the two suggested ASICs (9000, 32000; see table 1) is very low and does not pose any implementation problem. This paves the way for envisaging an implementation of all functions of the two ASICs (EU and TA) within one mode-switchable ASIC. A mode pin must be reserved to invoke the corresponding function.

One sector processor needs at least six EU-ASICs and one TA-ASIC. Consequently the barrel track finder system, consisting of 60 sector processors, employs 360 EU-chips and 60 TA-chips. Using a mode-switchable 420 equal ASICs would be sufficient.

It should be noted that this estimation is based on a worst case study using today’s technology. It was shown that even now the requirements can be fully met with no restrictions to the functionality of the track finder processor.

ALERNATIVE ASIC IMPLEMENTATION SCHEMES

Up to now only pure pipelined structures have been evaluated. However, in a pipelined design all input data must be inserted into the processing units within one cycle in order not to create dead time. The ‘Round Robin’ architecture employs a number of parallel running processors (see fig. 4). Each processor executes the entire algorithm for one single event. After each occurring event the data is sent to another processor which happens to be idle. When a processor finished its task it puts out the calculated data and is free to obtain the next data set. The number of processors necessary equals the execution time for one event divided by the bunch crossing period. A mixture between pipelined design and ‘round robin’ architecture, a combined architecture, allows to use more than one cycle for input data insertation. In addition the number of processing units can be smaller than with pure ‘round robin’ operation.

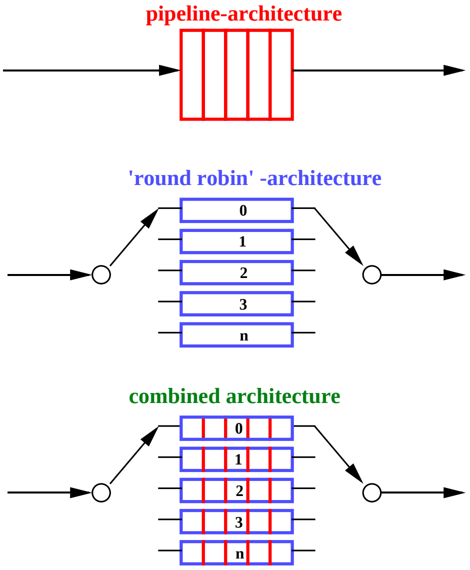

Fig. 4.: Three hardware architectures.

In the following two alternatives to the base line system partitioning are introduced. The presented schemes are supposed to guide appropriate ASIC technology selection at the time of final implementation.

A radical alternative to the baseline system partitioning is to implement all processing units from extrapolation EU, extrapolation value calculation EXT and extrapolation result selection ERS over track assembling TA and track segment routing TSR in one single integrated circuit. Two variants of this possibility are being introduced.

Variant 1:

The chip incorporates the equivalent of one track assembler chip, six extrapolation chips, the track segment router and assignment units.

The estimated gate count for one out of three track segment routers TSR is 1000 gates. The assignment units AU, except the p_t-assignment units, employs about 4000 gates. The total gate count is 93000 gates (see table 7-2). The extrapolation value calculation EXT employs SRAMs. If all processing parts are contained within an ASIC the total execution time is very fast, since no interchip transmission adds to the latency. Thus one can afford to employ only one RAM for one source track segment instead of three. Remember that in order to cope with the different origins of coordinates in the neighbouring detector segments three lookup tables are employed (see chapter 5.2.1.). The additional calculation time for subtracting a constant value from the track segment coordinates is below 7 ns. Employing RAMs as fast as to deliver extrapolation values for two extrapolations within one bunch crossing requires to implement only six RAMs of the size 256 x 18. This totals up to a memory bit count of about 28 kbit. A standard cell technology is more suited to implement a RAM within an ASIC. Using the standard cell technology ES2 0.7µm, which is already outdated [10], the estimated required die size for this amount of RAM equals 15 mm^2. Using the size of a single gate in the ES2 0.7µm technology of $4 \cdot 10^{-4}$ mm^2 yields an necessary active die area of 54 mm^2 including the RAM, which is a reasonable number.

However, the amount of input bits into the system is very large, because all track segment data must be made available to the circuit. The total number of bits yields 1692 (6 neighbours) and 2496 (9 neighbours) respectively. Assuming a multiplexing factor m_{IO} of two and an I/O pin count of 300 requires at least three bx (or four for nine neighbours) to read all data into the chip. Thus three (or four chips) must be used in parallel.

The total execution time for the chip is 57.3 ns or three bx. The delay of I/O multiplexing of chamber data (0.5 bx), EU-neighbour data (1bx), TA-neighbour data (1bx) and output data (0.5 bx) adds up to three bunch crossings. The data input process into the ASIC takes up to four bunch crossings. The p_t-assignment units requires a RAM of the size 128k x 8 (1 Mb). It cannot be contained by the ASIC. It will add another bunch crossing latency. For driving the output one bunch crossing is reserved. All this totals up to twelve bunch crossings.

Variant 2:

Up to now it was always considered to execute all extrapolations in parallel before starting the track assembling. Another implementation method combining the architectural structure of the extrapolation units and the track assembling units can be envisaged. For each track class a track searcher is employed. Beginning from the inner part of the detector extrapolations to the next chamber are being conducted and the address of up to two successful extrapolation targets are given out (block A in fig. 5). Multiplexers MUX route the track segment data of the extrapolation targets to the subsequent processing stage, block B. Using the target track segments as new source track segments the extrapolations to the next stations are being performed in block B, resulting in the addresses of the newly appended target track segments. The next step extrapolating from station three to four is not possible, as explained in chapter 4.3. Thus extrapolations from station four to three must be performed parallel to the extrapolations between stations one and two. For each track segment in station three the addresses of matching track segments in stations four must be provided. Block C then routes the corresponding track segment address and track segment data of station four to its outputs.

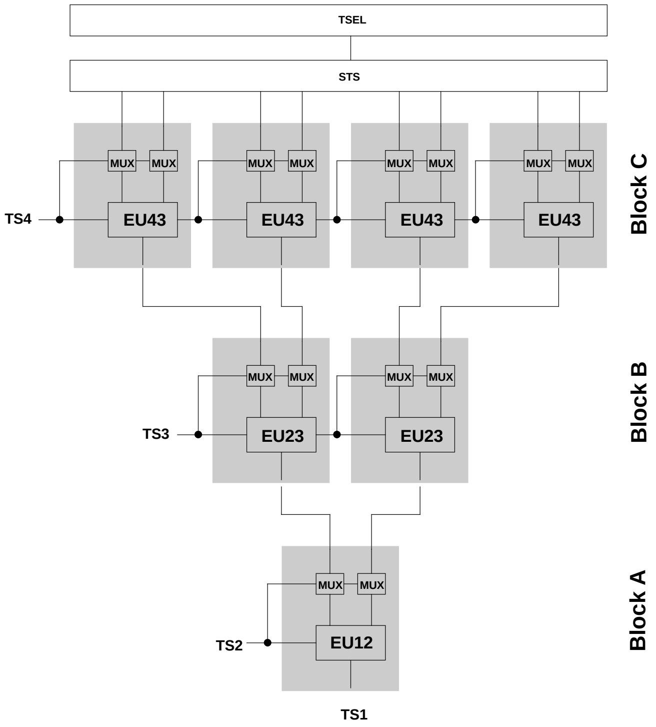

Fig. 5.: Block diagram of variant 2.

After all extrapolations have been performed the single track selection STS is applied and data is forwarded to the track selector TSEL, track segment router TSR and assignment units AU as in the previous explained architecture.

The advantage of this scheme is that all data necessary for track assembling in a given detector segment are produced within the given track finder processor and thus are available without any interconnections between several boards. This is a major advantage since interconnections cause additional processing delay and consume immense physical space. The only interconnections necessary between processors are those for the cancel out units (total 132 bits I/O).

Possible system partition for variant two

Two possibilities are open: One puts all necessary logic within one chip, another splits the system up into several small units with similar design features.

The first method is certainly favourable with respect to system complexity and execution time. However, the amount of input bits into the system is large. The total number of bits yields 1136 (6 neighbours) and 1616 (9 neighbours) respectively. Assuming a multiplexing factor m_{IO} of two and an I/O count of 300 requires at least two bunch crossings (or three for nine neighbours) to read all data into the chip. Two (or three) chips must be used in parallel.

As it is not clear at the process begin what track segment the extrapolation has to start from the extrapolation value calculation EXT cannot be dispersed from the chip. However, in this scheme a total of 38 extrapolations are necessary. Employing RAMs as fast as to deliver extrapolation values for two extrapolations within one bunch crossing requires to implement 19 RAMs of the size 256 x 18. This totals up to a memory bit count of about 90 kbit. Using the standard cell technology ES2 0.7µm [10] the estimated required die size for this amount of RAM equals 50 mm^2.

In addition the chip incorporates the equivalent of one track assembler chip and 19 extrapolations chips. This results in a gate count of 200000 gates. Using the size of a single gate in the ES2 0.7µm technology of $4 \cdot 10^{-4}$ mm^2 yields an necessary active die area of 135 mm^2 including the memory. Although this area is technically feasible by now die sizes of such dimensions should be avoided. Applying a more sophisticated technology as 0.5 or 0.35 mm^2 in the future will decrease the necessary active gate area.

The track segment router TSR and assignment units (except the p_t-assignment units PAU) can be incorporated by the integrated circuit as well, saving overall latency compared to the base line partitioning scheme. A rough execution delay estimation for this scheme yields 76.8 ns or 4 bunch crossings. The delay of I/O multiplexing of chamber data (0.5 bx), TA-neighbour data (1bx) and output data (0.5 bx) adds 2 bx. The data input process itself takes up to three bx. For assigning p_t using a SRAM and driving the outputs one bx is being reserved each. This totals up to eleven bx.

The other possibility to partition this system is to split the system up into several small units with similar design features. All blocks contain extrapolation units EU, multiplexers MUX, single track selector STS and track selector TSR. Using various modes, units are switched on or off.

Compared to the single chip design the I/O pin count of the chips is reduced.

The drawback, however, is the necessity of frequent transmissions between ASICs and the resulting increased delay. For each inter-ASIC transmission at least one half of a bunch crossing must be accounted for. Compared to the previous partitioning scheme, employing only one ASIC, additional four inter-ASIC transmissions must be performed. This totals up to 13 bx.

ALTERNATIVE IMPLEMENTATION TECHNOLOGIES

Industry [11] 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.

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.

CONCLUSION AND FURTHER PERSPECTIVES

In [6] it is shown that no previous implemented system is suitable to be applied in the track finder processor environment. Conventional methods applied in hardware triggers fail for the track finder. Especially the most common approach, the pattern comparison, must be ruled out because of the large hardware extent. Instead, an algorithm based track finding method, the track segment method or extrapolation method, is introduced. It is shown that the algorithm copes with the track finder specifications. The algorithm can be implemented with a minimum of hardware. Using VHDL simulation the algorithms and its hardware representation were optimized. Simulation shows clearly that the simplicity of the design concept, namely reducing data flow in subsequent steps (by extrapolation, track assembly and property assignment) and selecting the highest ranking track candidate after each reduction step without sacrificing measurement accuracy, proves to be an efficient method. The hardware of a FPGA-prototype is described. The algorithm is implemented in hardware with a small effort. The prototype clearly shows the proper functionality of the implemented system. Instead of storing a high number of track patterns (pattern comparison method) the problem is brought back to the algorithm level. Using simple logic modules, such as multiplexers, comparators, subtractors and logic gates, proves to be an important key point for the success of the design. However, it is pointed out that the number of bits to be processed in parallel poses a challenge to the hardware implementation. It is shown that the algorithm can be placed easily into an ASIC based hardware system that meets the given requirements. Processing speed and simplicity of the design are kept.

The design of the track finder processor, as it is introduced in this work, cannot at all be regarded as terminated. Although both the simulation and the prototype already delivered sophisticated results the track finder design presented here represents only a first step towards final implementation. The work suggests an algorithm and an implementation method. However, as the surrounding environment of the processor will evolve more and more both the specifications and the implementation of the track finder processor will have to be refined accordingly.

REFERENCES

[1] A. Kluge, T. Wildschek, The Hardware Muon Track Finding Processor in CMS - Specification and Method, CMS Note 1997/091

[2] A. Kluge, T. Wildschek, The Hardware Muon Track Finder Processor in CMS - System and Algorithm, CMS Note 1997/092

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

[4] T. Wildschek, An algorithm for Track Finding in the CMS Muon Trigger, Dissertation, Technische Universität Wien, 1997.

[5] S. Cittolin et al., Dual Port Memories in LHC Experiments, CMS-RD12, Technical Note 95-04, March 24, 1995.

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

[7] Motorola Semiconductor, H4C Series Design Reference Guide, Customer Defined Arrays 0.7 Micron $L_{eff,}$, Motorola INC, 1993.

[8] Cypress Semiconductor, Cypress Data Book Cy7C178, 32k x 18 Synchronous Cache RAM, 1995.

[9] C. Albajar, Measurement of Hadronic Shower Punchthrough in Magnetic Field, CERN-PPE95-61, May 9 1995, Submitted to Zeitschrift für Physik.

[10] European Silicon Structures, ES2 ECPD07 Library Databook.

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