

The CSC Track-Finding Processor for the Level 1 Trigger of the CMS Experiment at the LHC at CERN

B. Scurlock, S. M. Wang, and D. Acosta

University of Florida, Gainesville, FL

August 18, 1999

Abstract

This project was concerned with various aspects of design and construction of the Cathode Strip Chamber (CSC) Track-Finding processor in the Level 1 electronic 'trigger' which will be used in the Compact Muon Solenoid (CMS) experiment at the Large Hadron Collider at CERN. The layout of CMS, CSC detector operation, Level 1 trigger operation, and ongoing work on the Track-Finding processor are described in this paper.

1 Introduction

The Compact Muon Solenoid (CMS) experiment is expected to become operational in 2005. It will study proton collisions at very high energies - some of the highest particle energies ever produced by man. With such high energy collisions, physicists will be able to explore such things as the origin of mass and the existence of the Higgs boson. These collisions will take place inside an enormous cylindrical detector apparatus which consists of layers of various detector types used to measure spatial and energy information from passing particles. One type of detector which will be used at CMS is the cathode-strip-chamber (CSC) which is a multi-wire proportional chamber. CSCs will be used in the end cap regions of CMS and it is these regions which have been the main concern of this project.

At the Large Hadron Collider (LHC), proton bunches will collide with each other at a very high rate - once every 25ns (40,000,000 colliding bunches per second). The total

collection of all CMS detector information which result from 1 proton bunch collision is called an event. The recording and detailed analysis of every event is an impossible task: one event will produce 1 megabyte of information - about 40 terabytes of information will be generated per second; thus, CMS must utilize a trigger which will select only 100,000 events per second for further analysis. This is the task of the Level 1 (LV1) trigger. Other triggers will then be used to further reduce this rate to 100 events per second.

Due to the high rate of collisions, detector information must either be accepted for further analyses or rejected as uninteresting in a small amount of time because of constraints on available storage space. Thus, the Level 1 trigger must make the keep or reject decision in as short a time as possible. The strategy being implemented in CMS is to pipeline the detector information through a series of memories. While the data is being pipelined, the LV1 trigger works in parallel. The trigger performs many tasks in order to make its final decision. These tasks require more time than 25ns (25ns = 1 CMS clock cycle); thus, each task is performed in a series of steps which also take multiple clock cycles. When the final task is finished, the trigger will send out an accept/reject command for corresponding pipelined data. This accept/reject decision is made about 120 proton bunch crossings later (see Figure 1) [1].

Part of this project involved the simulation and implementation of various parts of the LV1 trigger process through the use of programmable logic devices; this involved the development and testing of many algorithms which are needed to perform specific tasks within the trigger - mainly within the track finding process. This is the process which will use pattern recognition to correlate the multitude of seemingly random particle hits in the detector layers, with the particles that caused them. Each of these tasks require careful logic and timing analysis so that the overall triggering and track finding process will be accomplished within the needed time constraints. In particular, we developed sorting algorithms required to select the best quality muon tracks in the shortest amount of time.

2 CMS Geometry

The basic shape of CMS consists of a series of concentric cylinders. It weighs a total of 12,500 tons, has a total diameter of 15 meters, and length of 21.6 meters. Starting from the central axis along the z -direction is the beam line - this is the axis along which the proton bunches will collide. Surrounding the collision site will be a series of inner trackers and calorimeters which are aligned concentrically and are cylindrical in shape. All of these detectors offer fast responses for the detection of the multitude of particles expected to be seen in this region.

Outside of this first series of detectors is a superconducting coil that will generate a 4 Tesla magnetic field. Outside of this coil will be a series of muon detector layers. In the barrel portion, drift tubes and resistive plate chambers will be used, and in the end cap portions, cathode strip and resistive plate chambers (CSCs and RPCs) will be used. In between each outer detector layer will be placed iron yokes which act to contain the high magnetic field. These various layers can be seen in Figure 2.

CMS uses a cylindrical coordinate system in terms of (r, ϕ, z) . However, it is often more convenient to use (ϕ, θ, z) coordinates instead, where $\theta = \arctan(r/z)$. Along the central axis of CMS is the z coordinate. Using the z direction to define a standard rectangular system, let the y -axis point up vertically, orthogonal to z , and let x point horizontally, orthogonal to both z and y . ϕ is the angular measurement in the $y - x$ plane, and θ is the angular in the $y - z$ plane (see Figures 2 and 3).

3 CSC Operation

Each CSC contains six detector planes. Each plane is constructed of cathode strips which are used to measure the ϕ coordinate of a passing particle, and anode wires which are in the orthogonal direction used to measure the θ coordinate. The anode wires are kept at a high positive potential and the cathode strips at ground. The cathode and anodes are held in a special mixture of gas which will ionize via coulomb interaction, with the passage of charged particles. After a portion of gas is ionized, free electrons will drift to an anode wire. The collection of electrons on the anode wire will induce an opposite charge on a group of cathode strips. The differences in charge on each individual cathode strip can be used to define a curve whose peak can be interpreted as the most probable position of the particle in the ϕ position. The collection of electrons on the anode can be used to find the particle's θ position. The combined anode and cathode signal determines a two dimensional coordinate for the passing particle [2]. The z coordinate is given by the CSC detector plane. When six planes of these cathodes and anodes are combined, one can obtain a three dimensional track-stub through the whole chamber (see Figure 4).

4 Endcap Regions

The CMS endcap region will contain 4 detector layers of CSC chambers. An iron yoke will be placed in between each layer to contain the large magnetic field. CSCs are the preferred

detector type for this region because they can operate well in the inhomogeneous magnetic field contained in the endcaps and have fast response times (electrons drift to anodes in about 60ns). These chambers are constructed to be trapezoidal in shape, extending 10° or 20° in ϕ , such that an array of them can be arranged in a disk-like orientation to make 1 station. This arrangement of chambers guarantees full detector coverage in the ϕ direction [1].

4.1 Chamber Organization and CMS Electronics

Information from each CSC is sent to front end electronics which are mounted on and around the chambers. The front end electronics will locate the positions and time for detections in each CSC and construct track-stubs. These track-stubs will then be sent to the sector processor (SP) which contains the track-finding processor (see Figure 5 and Appendix A).

1 SP takes the 3 best track-stubs from a 60° section in ϕ which extends through z from 1 endcap (1 sector), and tries to find correct muon trajectories through all 4 stations. The SP will link the track-stubs into complete tracks (see Figures 6 and 7), assign a momentum to the muons responsible for the tracks, and assign a quality rating to each track. The quality is a numeric rating of how good the constructed muon track is compared to the other muon tracks. Thus, the higher the quality number, the better the track. Each SP sends out its best 3 constructed muon tracks to more logic which will sort the tracks, correlate the tracks with data from other detectors, and finally send out the best 4 muon tracks along with the final trigger decision (Appendix B) [3].

4.2 Track Finding Logic

Within 1 SP is the track finding logic. This logic receives track stubs from each station and constructs 6 possible tracks. From these tracks, 3 distinct tracks with the highest quality must be selected. Thus, the 6 full tracks are sent into a track sorting algorithm which will select the best 3 (see Figure 8). This project focused on the creation and testing of this track sorting algorithm.

5 Sorting Algorithms

In order to know which 3 tracks have the highest quality, the 6 track qualities must be placed in order so that the lowest 3 can be eliminated and the remaining 3 can be passed along.

Thus, a digital sorting algorithm must be implemented. To make things more complicated, the algorithm must first check to make sure that none of the 6 tracks are identical to any other one. If this is not done, we could end up with the highest 3 quality tracks actually being made up of 2 or more tracks that are identical (see Figure 9). Before creating a sorter to find the best 3 distinct tracks, we had to first develop a sorter to find the 3 highest quality tracks. Thus, this initial sorter does not check for distinctness in a track.

5.1 The Quality Sorter Without ID Test

A number of sorting algorithms exist which work well in programmable logic chips; however, most of these require a number of clock cycles to complete because they utilize stages of pairwise comparisons whose results are used in each following step - for example, bubble sort and odd-even merging. A large number of clock cycles is not an asset to a trigger whose job needs to be completed in the shortest time possible. Thus, we needed to create a way to sort binary numbers with the minimum number of clock cycles. The ideal number of cycles is 1; however, this is not possible given current constraints on memory sizes (with most processes involving digital logic, a compromise must be reached between resource size and time constraints). We are however, able to sort a list of N numbers in 2 clock cycles by using VHDL - a electronics design programming language used to configure hardware such as programmable chips (see Appendix C).

The VHDL sorting method is carried out in 2 steps: first it compares each possible pairing of the inputs, then takes the results from each of these comparisons and combines them to create a binary number. This binary number is then translated into a specific ordering of the inputs. To carry out these processes, the sorting method uses two basic elements. The first is the 2-input comparator, and the second is the VHDL bus translator. The 2-input comparator reads inputs a and b , if $a \geq b$, a 1-bit output will read '1'. The output will read '0' if $a < b$ (see Figure 10). The second element is the VHDL bus translator which will take the outputs from the 2-input comparator and assign an ordering to the inputs.

So, for a single list of N number of inputs a_1, a_2, \dots, a_N being sent into a N -input sorter, a_1 and a_2 , a_1 and a_3, \dots, a_1 and a_N , a_2 and a_3 , a_2 and a_3, \dots, a_2 and a_N, \dots , and in general $a_{N'}$ and a_N where $N' < N$ must all be compared. For N inputs, the number of comparisons needed in this first step is $n = \frac{N(N-1)}{2}$. The combination of these outputs will be sent into a $[\frac{N(N-1)}{2}]$ -bit bus (see Figure 11). Each possible ordering of the N inputs will have a corresponding value for the n -bit bus. It is possible to program each of the values for the n -bit bus such that the proper ordering of the inputs will be given. A n -bit bus will have 2^n

possible values; however, only $\frac{N!}{(N-3)!}$ values are needed for our purposes. Thus, only specific bit-patterns of the n -bits have to be considered when determining the order of the inputs.

The VHDL bus translator looks for specific bit patterns in the n -bit bus. These patterns are easily generalized with parametric equations which describe the values of a $N \times \frac{N(N-1)}{2}$ truth table (see Figure 12). Each pattern is programmed into VHDL in the form of: if input bus has pattern A, then input x has the greatest value, input y has the second greatest, and input z the third (see Figure 13). Rather than program each of these if-then statements manually (for sorting 6 inputs, there can be as many as 120 if-then statements), we created a C macro which will use the parametric equations to generate the VHDL code for us. This greatly reduces the amount of time spent on error checking and typing VHDL. The greatest advantage of using this macro is that it gives us the ability to create sorters for any number of inputs because the parametric equations are completely generalized for number of inputs and the number of bits per input.

5.2 Timing Analysis and Size

In order for this sorting algorithm to be of any use, it must meet the timing requirements for the LV1 trigger. Thus, it must perform its task in the shortest time possible. To see if it does this, we must make timing measurements on the Field Programmable Gate Array (FPGA) (see Appendix C). FPGAs are the programmable chips that are used to execute the Track Finding logic. The most basic way to check the number of clock cycles needed by the algorithm is to use a logic simulator. OrCAD Express, a FPGA programming software package, comes equipped with such a feature which allows one to error check the logic in the algorithm, and to see how internal chip lines change with time. This works well for the sorting algorithm because it allows us to check that the outputs are actually in order within the correct amount of clock cycles.

A more realistic timing analysis can then be done by using the delay information generated by OrCAD during the VHDL compilation process. OrCAD calculates the delay through each of the internal lines it routes in a FPGA during compilation. While these simulations offer valuable information during the process of creating the sorting algorithm, we must still conduct timing measurements within a real chip. Thus, an actual delay measurement using an oscilloscope and pulse generator must be performed. This is done by using a pulse generator to vary an input at a given frequency, and connecting the oscilloscope to both the varied input and the effected output. Most digital oscilloscopes can automatically measure the delay between two signals. So, using the oscilloscope, we measure the delay in the chip

by measuring the delay between the oscillating input and output. Another consideration for this algorithm is the amount of FPGA resources it needs. For example, if the algorithm requires more Configurable Logic Blocks (CLBs) (see appendix C) than the desired chip has, it must be modified or abandoned in favor of a more reasonable algorithm.

5.3 Results for the VHDL Sorting Algorithm

For simple tests of logic programs we use the Xilinx 4005E FPGA. This FPGA only has approximately 40 usable i/o pins; thus, only a small scale test can be completed with this chip. For a 4-input VHDL sorter with 5 bits per input, we measured a 20ns delay for the bus translation portion of the sorter, which is less than 1 CMS clock cycle. Previous tests have shown that the delay through a comparator is also less than 1 CMS clock cycle. This implies that the total delay for this algorithm is less than 2 clock cycles to complete. The delay should improve with better quality chips with higher speed grades. The amount of chip resources required for the algorithm is very small. For a 6-input sorter, which is the size being used by the CSC Track-Finding processor, only 226 of the 400 CLBs in a 4005E chip are needed.

5.4 The Modified Version of the VHDL Sorter: The Quality Sorter with ID Test

As stated above, it is not enough to find the muon tracks of greatest quality, we also need to make sure that none of the tracks are identical. Thus, cancellation logic must be added to the sorting algorithm. First, all pairwise combinations of the track qualities are compared, just as they are above; while in parallel, the IDs of each track are compared for equality. If any of the IDs are found to be equal, the sorter checks to see which of the two have the higher quality. The track with the lower quality is then disqualified from the list of track candidates. This cancellation process modifies the n -bit bus such that the disqualified tracks will not be selected as 1 of the best 3 tracks. This modified bus is then sent into the VHDL bus translator as before, where the top 3 distinct muon tracks will be selected (see Figure 14).

This modified version of the sorting algorithm requires only 1 additional clock cycle - approximately 1/2 of a clock cycle for the logic which decides if an input is to be disqualified, and 1/2 of a cycle for the logic which corrects the n -bit bus to be sent into the bus translator. Thus, a total of 3 clock cycles is needed to select the best 3 distinct muon tracks.

6 Conclusion

We succeeded in creating a sorting algorithm which will take the required 6 tracks for the SP and will select the best 3 tracks. This can be accomplished in the minimum amount of clock cycles, thus enhancing the performance of the CSC Track-Finding processor. This will of course, decrease the necessary time required to select the best 3 muon tracks per sector, which in the end will decrease the amount of resources required to pipeline the detector data which await the LV1 trigger decision.

Acknowledgments

I would like to thank the NSF for funding the summer REU program, and Kevin Ingersent and Alan Dorsey for allowing me to participate. I would also like to thank Raymond Chow and Mike Marquez for helping me with the creation and testing of this project.

A Details of CMS Electronics

Each CSC is organized into groups of 16 cathode strips and 16 anode wires. Each of these groups is projected through all 6 planes in the chamber to create 96 (16 x 6) channels to be read out through the front-end electronic board (FEB) one board per 96 channels for both cathodes and anodes. Thus, cathode strip and anode wire signals are first analyzed separately by individual FEBs. The cathode FEBs use a Comparator Network which finds the correct position of a hit along the ϕ position in a single chamber plane, and then marks the muon's time of passage through that point. This information is then sent to a Local Charge Track (LCT) processor which correlates the ϕ information for all 6 planes in the chamber. Given a valid pattern of hits through the minimum number of planes, a Local Charge Track (LCT) in ϕ is formed and its time of passage is marked and sent on to the Trigger Motherboard (TMB). A similar process is completed in the anode FEBs; however, the LCT processor creates LCTs in the θ direction. Both the anode and cathode LCT processors select up to the best 2 LCTs in ϕ and θ in each CSC.

A TMB receives data from all of the cathode and anode FEBs from the CSC and correlates the 2 ϕ and 2 θ LCTs to form the 2 most probable muon trajectories through the CSC. These trajectories are actually straight lines which can be used to construct vectors through the CSC. These vectors are referred to as muon LCTs or track-stubs. The TMB will also

associate each track-stub with the correct proton-proton bunch crossing from which the muon originated. The TMB then sends the muon LCTs to the Muon Port Card (MPC).

The MPC takes 2 track-stubs from 9 CSCs in one 60° endcap sector - that is, 18 total muon LCTs at once, and finds the best 3 track-stubs for that sector. The MPCs for the first station differ slightly from this because of the greater amount of CSCs being used. Since the first station uses only 10 degree CSCs, there are 9 total CSCs in a 30° sector; therefore the MPCs for the first station receives 18 track-stubs from only a 30° sector. The MPCs then send their 3 best track-stubs to a Sector Receiver which will translate the stubs in to ϕ, θ , and quality information which will be readable by the Sector Processor (SP) [3].

Each SP sends out its best 3 constructed muon tracks to a Muon Sorter (MS). The MS receives the best 3 muon tracks from each of 12 SPs for a total of 36 muon tracks from which the MS will select the best 4. These 4 muon tracks will be sent to a Global Muon Trigger which will analyze all of the trigger data from the CSC, DT, and RPC detector regions and the Calorimeter trigger to find the best 4 muons which are then sent to the Global Trigger where the muons are further analyzed and the final LV1 trigger decision is made [4].

B Track Construction with the SP

15 total possible track-stubs will be sent to the SP - 6 for station 1, and 3 each for stations 2-4. If all 15 available track-stubs are sent to the SP, then an enormous amount of possible tracks could exist that a muon could take. For example, if the SP received 3 track-stubs in both stations 2 and 3, that would mean that the muon could have taken one of 9 possible paths between stations 2 and 3. The number of paths increases when we consider stations 1 and 4 as well. Luckily, we know of certain things that the muon of interest cannot do when moving along a given track, and this information allows us to cancel out possible paths. For example, if in station 2, we have a track-stub that points linearly towards the collision vertex, and in station 3 a track-stub points in some direction orthogonal to the track-stub from station 2, then we can conclude that whatever caused the station 2 track-stub could not have caused the station 3 track-stub. Thus, a track cannot be constructed between those two stubs. There may however, exist another track-stub in station 3 which also points in the vertex direction, and so we may conclude that the same muon which caused the station 2 track-stub caused the station 3 stub; thus, we may construct a track between the two stations along the track-stub paths.

B.1 The Extrapolation Process

In order to find a correlation between each track-stub in each station, we must first take the stub of interest and extrapolate it to the other stations. To be consistent with a muon passing along a given path, the stubs in the other stations must have their parameters lie within an allowed range. So, the extrapolation is compared to the actual stubs which are measured in the other stations. If it is found that a stub from one station lies in an allowed range for the extrapolated stub, then a track may be linked between the two stubs - a link between 2 stubs is enough to be considered a full track. If the 15 stubs are measured (6 from station 1, and 3 from stations 2-4) the maximum allowed for one 60° endcap sector, then a total of 81 track-stub comparisons must be made. However, the extrapolation from station 1 to 4 has been prohibited in order to rule out triggering on random hits; thus, a total of 63 comparisons must be performed. The extrapolations will be completed by performing pairwise extrapolations from the stubs in station 2 to stations 1,3,and 4, and in parallel performing extrapolations from the stubs in station 3 to stations 1,2, and 4. From each of these extrapolations, track segments may be created, and a quality is assigned to these segments. Each track segment is then checked to see if it can be linked with other track segments to create longer tracks, if it can, a new track is formed by this combination and a new quality is assigned.

Since extrapolations are carried out in parallel using two different stations (stations 2 and 3), any identical tracks resulting from both of these stations must be combined to eliminate redundancy. The criterion for deciding when a track from one station is identical to the other remains flexible. When the tracks are considered identical, the track with the lower quality must be eliminated. Finally, the complete tracks with of the highest three qualities will be selected and sent out of the SP.

C FPGAs and VHDL

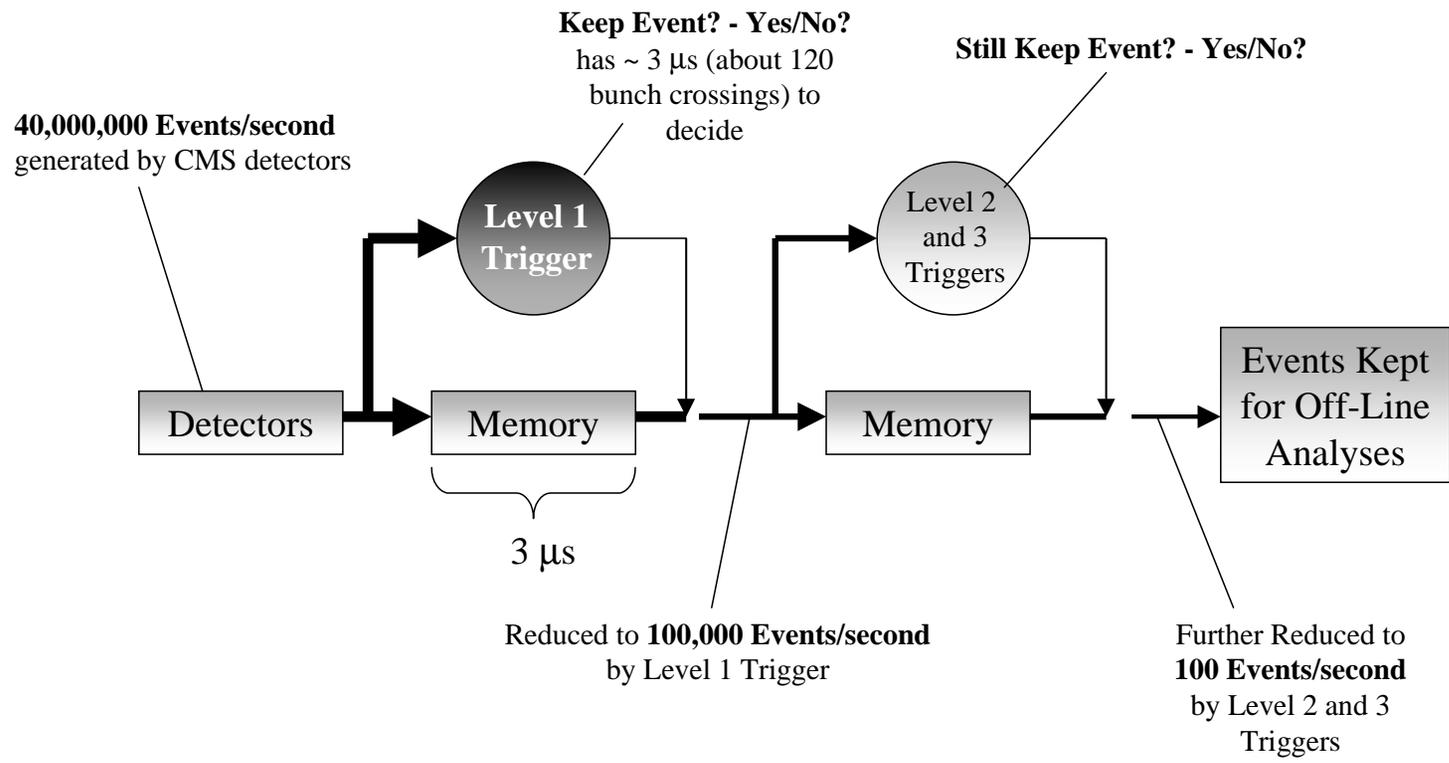
Field Programmable Gate Arrays (FPGAs) are programmable chips which contain elements that can perform Boolean logic operations such as “and” and “or” gates. They also contain flip-flops, and programmable memories for RAM and ROM operation. The basic unit of the FPGA is the configurable logic block (CLB) each of which contains these elements. It is in the CLBs that the desired logic is constructed. We test logic using the Xilinx 4000E family chips which contain around 400 CLBs. These chips come in various sizes, i/o pin numbers, speeds, etc. We program these chips using software such as OrCAD Express. OrCAD

allows us to either visually create the logic operation we would like use - for example, if we want to use the “and” operator for two data lines, we can schematically draw the lines connecting to an and- gate with the OrCAD graphical interface, or we can program the logic using (Very High Speed Integrated Circuits) Hardware Description Language (VHDL). Using this programming language, we can create things like the logical and-gate through if-then statements rather than schematically. We can also use a combination of both VHDL and schematic representations to further simplify projects. Physical i/o pins are mapped to the internal lines using VHDL or schematic, and these lines are then fed into a combination of gates and other chip elements. The combination and organization of these chip elements which produce the desired logic are constructed by the OrCAD software. Thus, OrCAD takes the schematic diagrams and VHDL code, compiles it, and turns it into a readable format to be sent into the FPGA.

References

- [1] CMS Collaboration, Muon Technical Design Report, CERN/LHCC 97-32 (1997)
- [2] Fernow, R., *Introduction to Experimental Particle Physics*, Cambridge University Press, New York, NY, 1986.
- [3] D. Acosta, et al, “The Track-Finding Processor for the Level-1 Trigger of the CMS Endcap Muon System,” CMS Internal Note 1999/000, http://www.phys.ufl.edu/~acosta/cms/sp_design.pdf
- [4] J. Kubic, “CMS Endcap Muon Trigger System Inter-Module Signals,” <ftp://hepsun0.physics.ucla.edu/pub/cms/trigger/triggerbits.ps>

Figure 1: Trigger and Data Acquisition scheme for CMS.



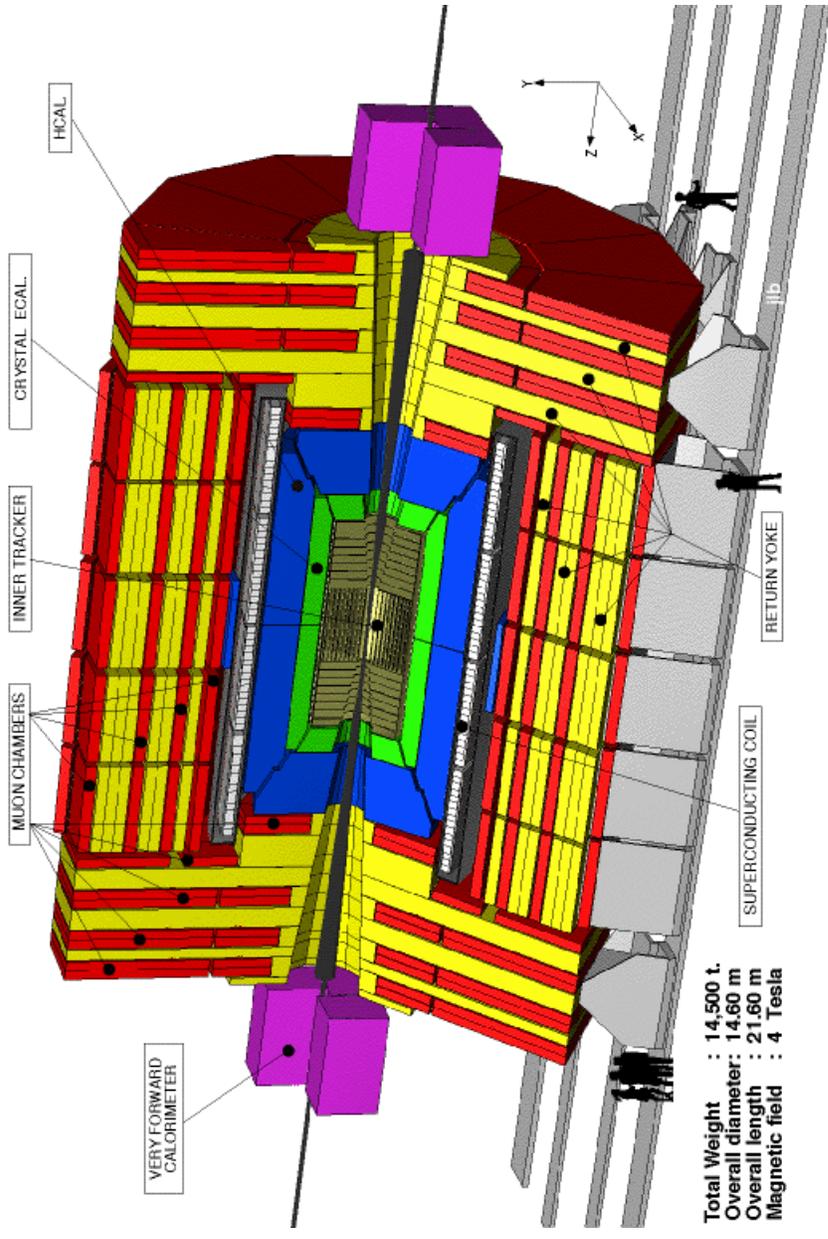


Figure 2: Various detector layers of CMS.

C.M.S.
A Compact Solenoidal Detector for L.H.C.

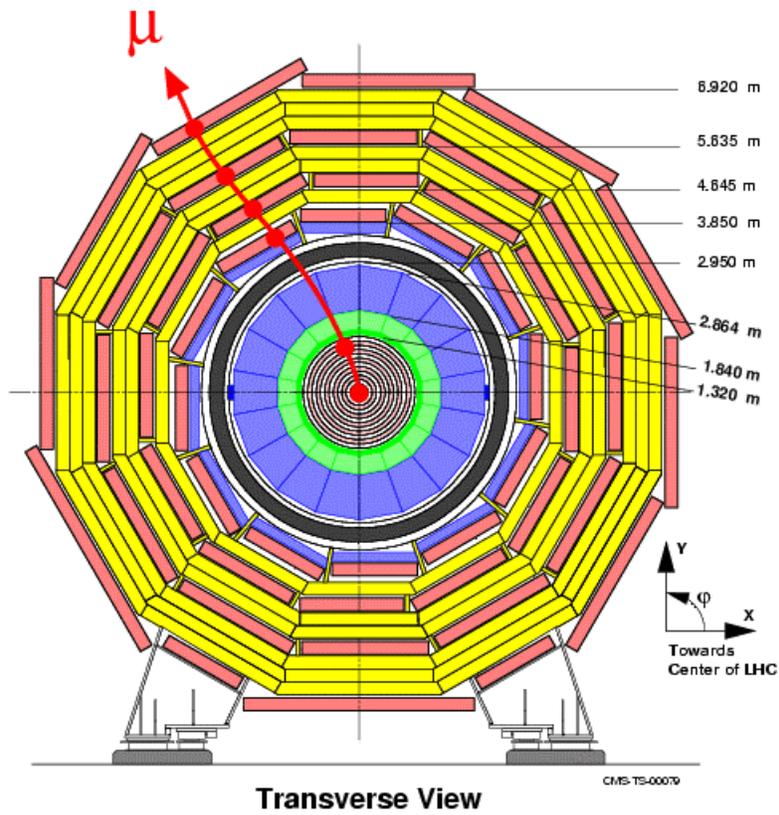


Figure 3: CMS coordinate system.

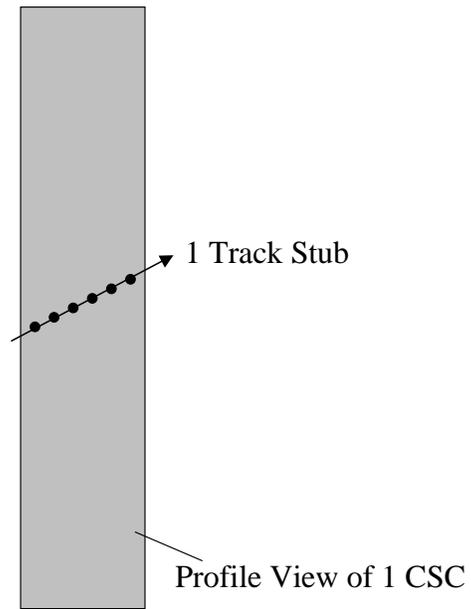


Figure 4: Profile view of 1 CSC showing 6 detector planes which can be used to create track-stubs.

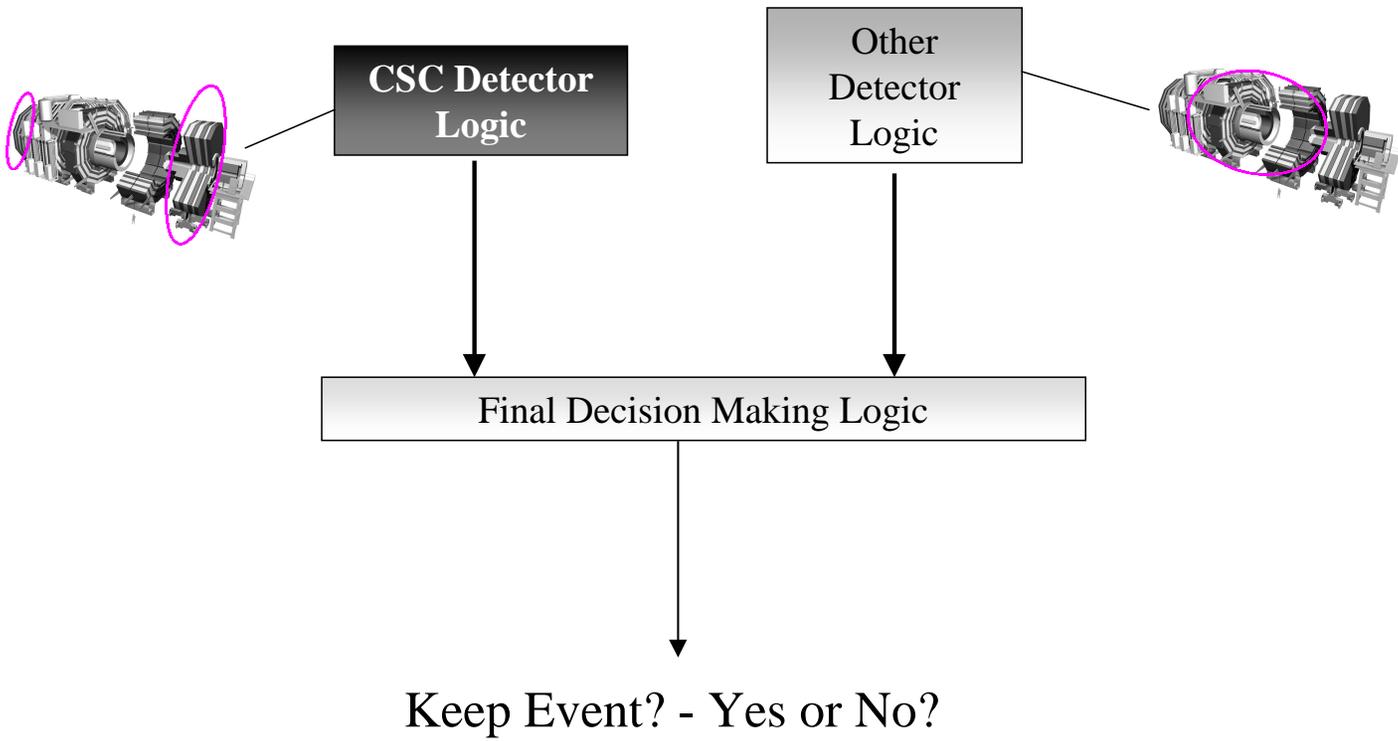


Figure 5: Detector regions used in LVL1 Trigger

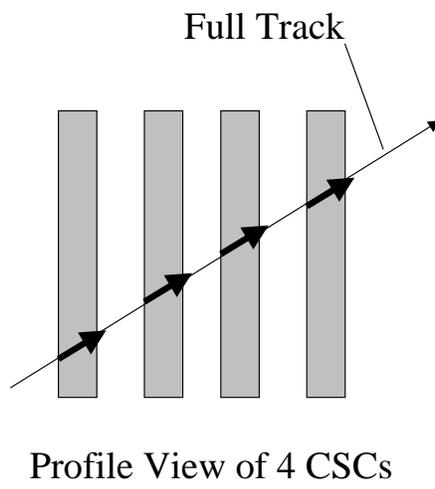


Figure 6: 4 track-stubs used to construct a full muon track.

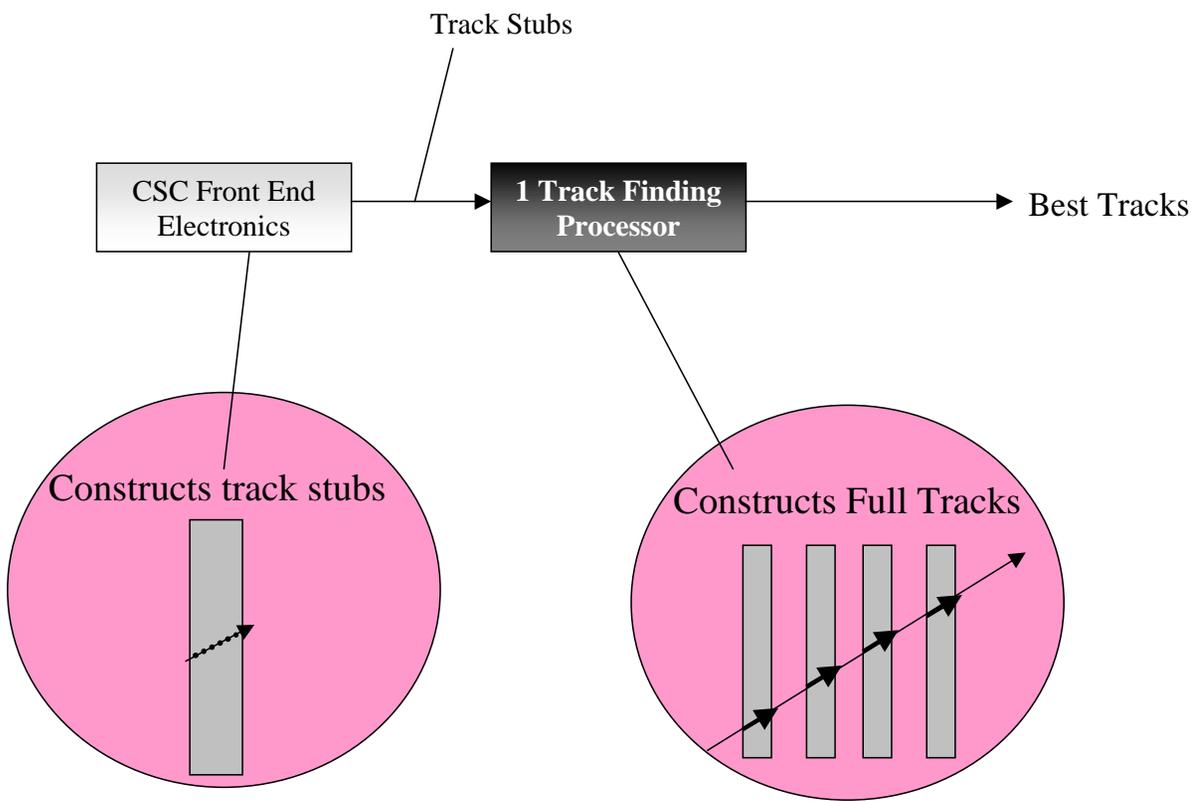
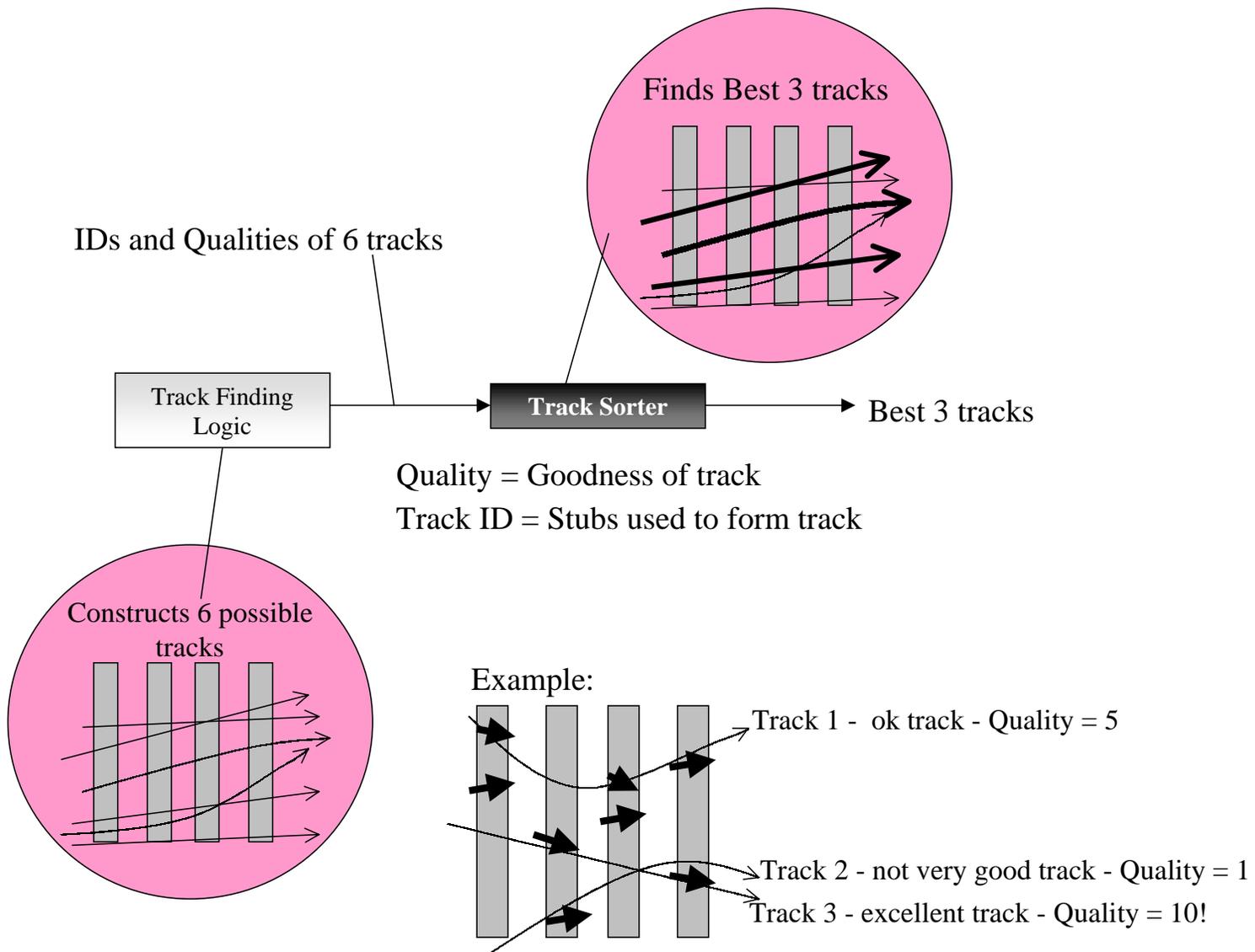


Figure 7: Details of CSC detector logic.

Figure 8: Details of Sector Processor.



Best 3 tracks = 3 distinct tracks w/Highest Quality #

Must Sort Quality #s to find highest 3 => Need to use Binary Number Sorter

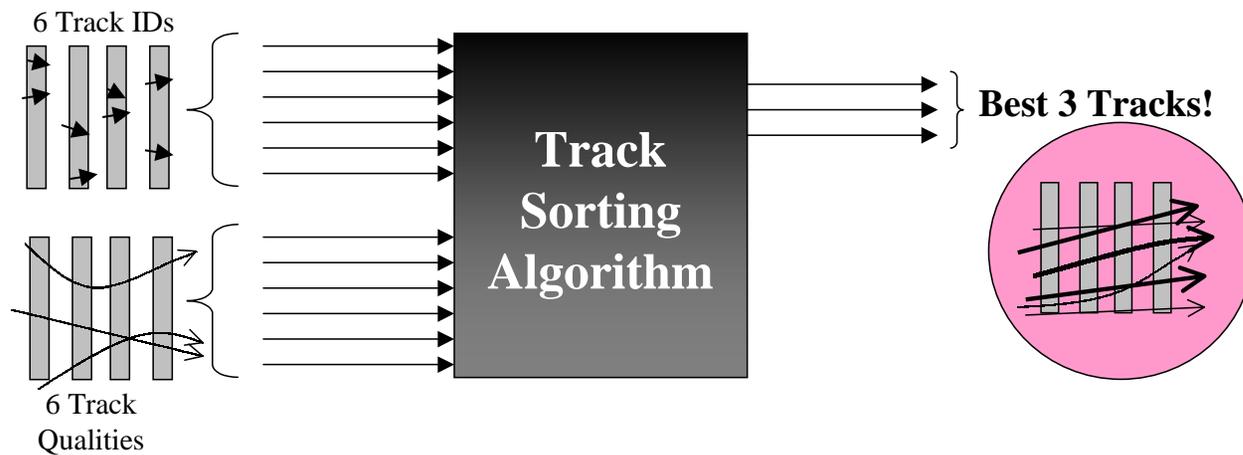


Figure 9: Details of Track Sorter.

Tasks: 1) Sort these Tracks in the Shortest Time Possible

2) Make Sure These Tracks are Distinct

Basic Number Sorter:

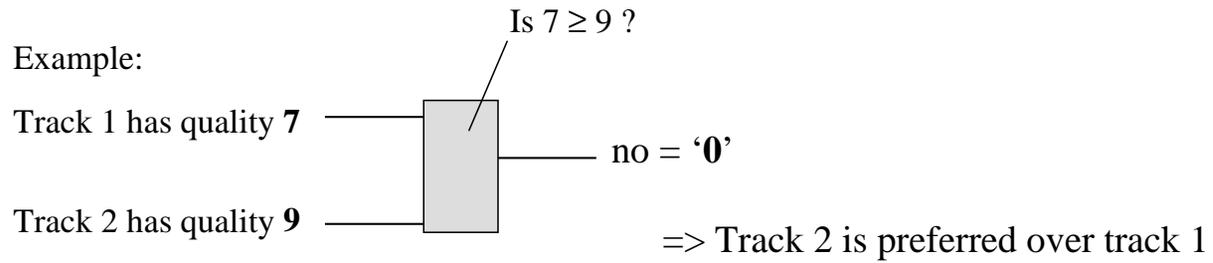
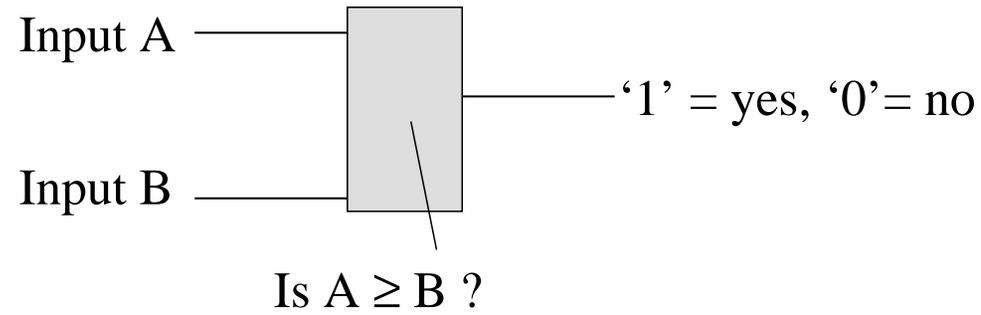


Figure 10: 2-input comparator operation.

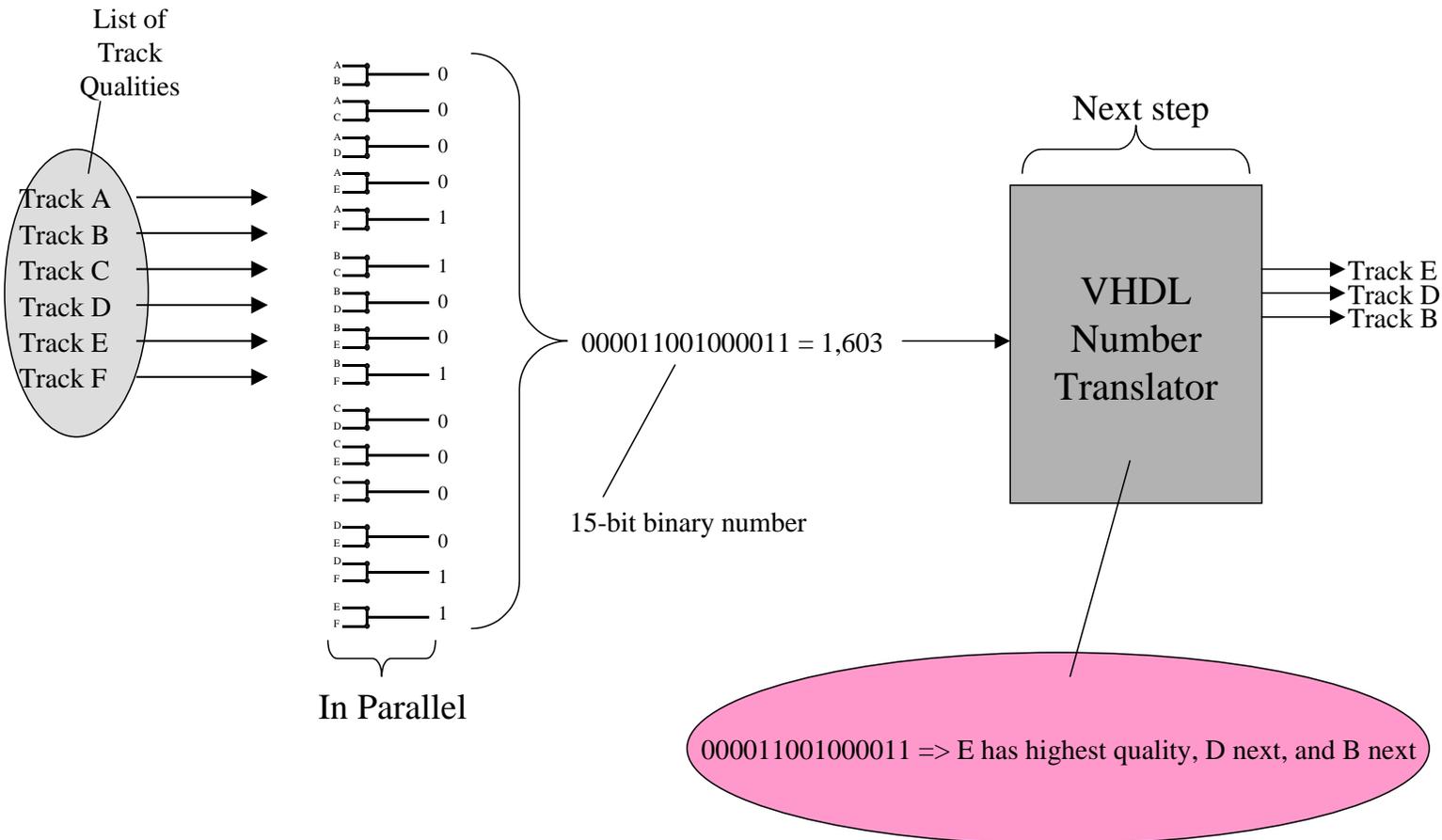
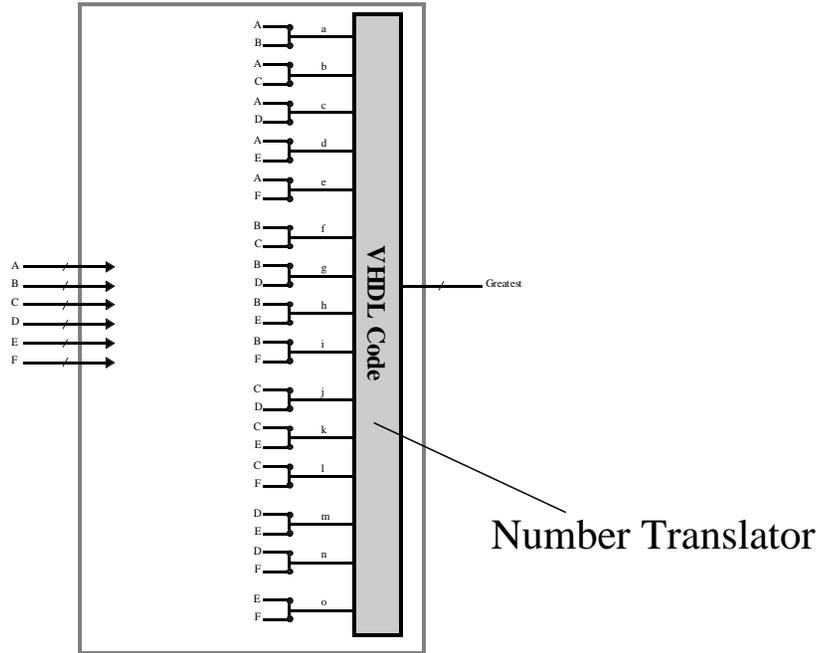


Figure 11: Using 2-input comparator to sort inputs.

Each 15-bit value corresponds to a unique ordering of the inputs



Bits that must be set for Track B to be the best

	compa	compb	compc	compd	compe	compf	compg	comph	compl	compj	compk	compL	compm	compn	compo
inputA	1	1	1	1	1	x	x	x	x	x	x	x	x	x	x
inputB	0	x	x	x	x	1	1	1	1	x	x	x	x	x	x
inputC	x	0	x	x	x	0	x	x	x	1	1	1	x	x	x
inputD	x	x	0	x	x	x	0	x	x	0	x	x	1	1	x
inputE	x	x	x	0	x	x	x	0	x	x	0	x	0	x	1
inputF	x	x	x	x	0	x	x	x	0	x	x	0	x	0	0

Figure 12: Example of truth table used to find the greatest of 6 inputs.

Figure 13: Conversion of truth table into VHDL.

	compa	compb	compc	compd	come	compf	compg	comph	compl	compj	compk	compl	compn	compo
inputA	1	1	1	1	1	x	x	x	x	x	x	x	x	x
inputB	0	x	x	x	x	1	1	1	1	x	x	x	x	x
inputC	x	0	x	x	x	0	x	x	x	1	1	1	x	x
inputD	x	x	0	x	x	x	0	x	x	0	x	x	1	1
inputE	x	x	x	0	x	x	x	0	x	x	0	x	0	x
inputF	x	x	x	x	0	x	x	x	0	x	x	0	x	0

Truth Table



```

if ((compa = '1')and(compb = '1')and(compc = '1')and(compd = '1')and(come = '1')) then
    inputA = BestTrack;

```

VHDL

Take the same basic VHDL Sorter and **add some disqualification Logic**

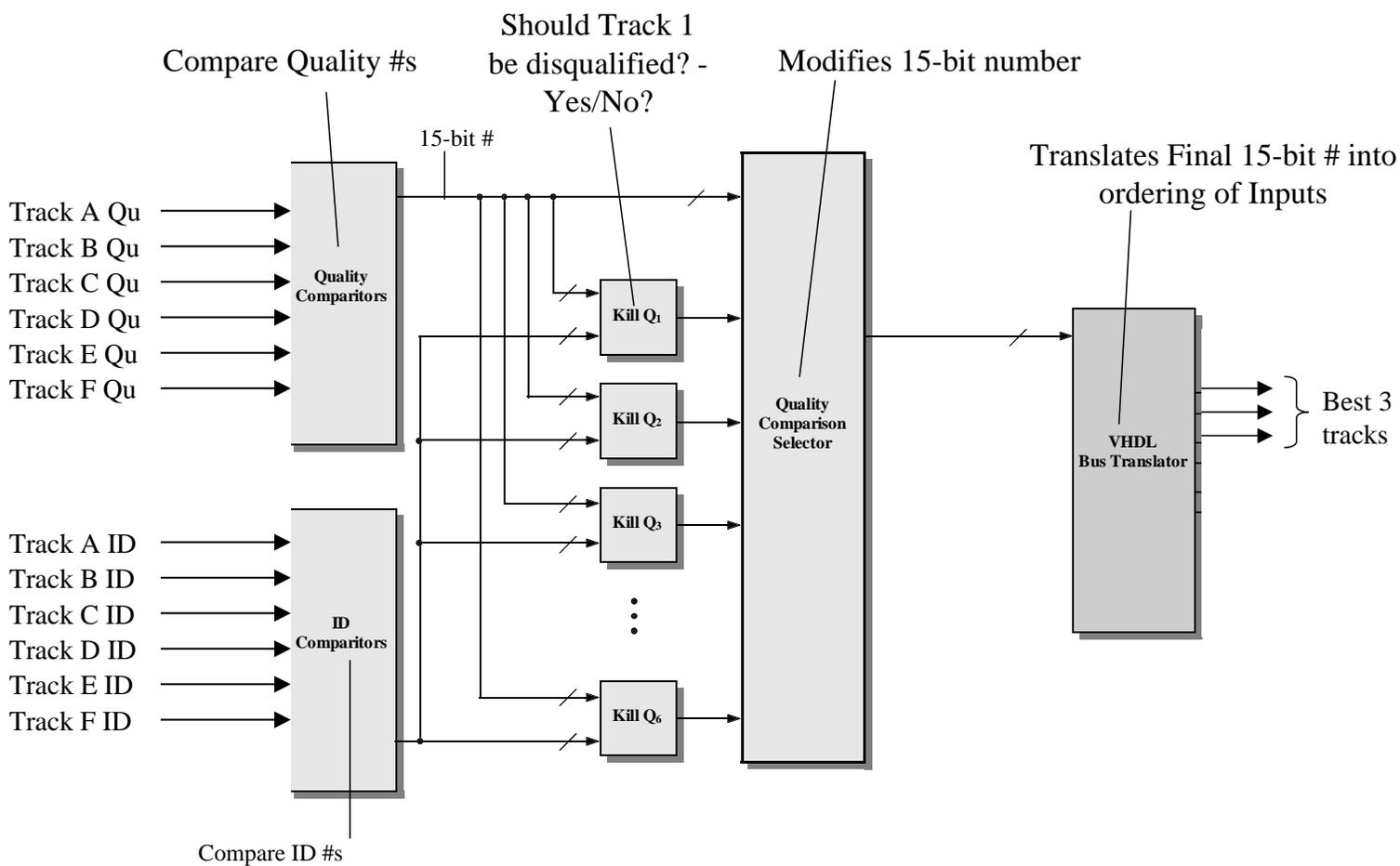


Figure 14: Track sorter with added cancellation logic.