CMS TN/95-028
6 March, 1995

The Sorting Processor Project

G.De Robertis and A.Ranieri
Sezione INFN di Bari, Italy

I.M.Kudla
Institute of Experimental Physics,Warsaw University, Poland

G.Wrochna
on leave of absence from Institute of Experimental Physics,Warsaw University, Poland

Abstract

The general scheme for a Sorting Processor like tree structure, is presented here together with some simulation results made considering a CMOS 0.7μ technology implementation of such device. This scheme will be implemented as part of the electronics foreseen for the trigger muon system of the CMS experiment.

1. Introduction

The CMS muon trigger is required to identify muons, measure their transverse momentum p_t, and determine the bunch crossing from which they originated. This is achieved with four muon stations in the return yoke of the CMS magnet. Each station will have one or two planes of RPC's as dedicated μ trigger detector [1][2][3]. RPC's have been chosen for their excellent time resolution, to ensure unambiguous bunch crossing assignment. A muon traversing RPC's will give usually at least four points hit pattern used to make the p_t cut.

Due to the energy loss and the multiple scattering a muons of a given p_t and pseudorapidity may produce different hit patterns. A set of those predefined hit patterns are loaded into a Pattern Comparator Trigger (PACT) processor [4].

The PACT will compare the predefined hit pattern with the observed pattern, delivering the p_t code of the highest momentum track inside of a detector segment ($\Delta\eta$=0.11 and $\Delta\varphi$=2.5), to the sector (ring) processor. The Ring processor, by using a like tree Sorting processors structure, selects 4 highest p_t muons code, in a ring of 0.11 rapidity unit and transmits the data to the global muon trigger system where, the trigger informations coming from the others parts of the global CMS muon detector, are combined.
The aim of this note, is to describe the internal structure of the Sorting processor, how it works and finally the structure, developed as a Sorting Processors Network, of the Ring Processor, which is the final goal of our project.

2. The Sorting Processor Structure

The Sorting Processor, is a specialized VLSI circuit, composed by the following parts:

- a 6 bits two words comparator;
- a chain of 4 multiplexers, 2 for data and 2 for addresses values;
- a 2 fold 4 inputs Sorters;
- an 8 inputs Merger.

These components are connected together to finally form, an 8 Inputs Sorting Network (the Sorting Processor), composed by 2 Sorters plus 1 Merger. The general structure of the Sorting Processor chip is shown in the Fig.1. In this figure both the Sorters and the Merger, are more complex elements the structure of which, are explained later.

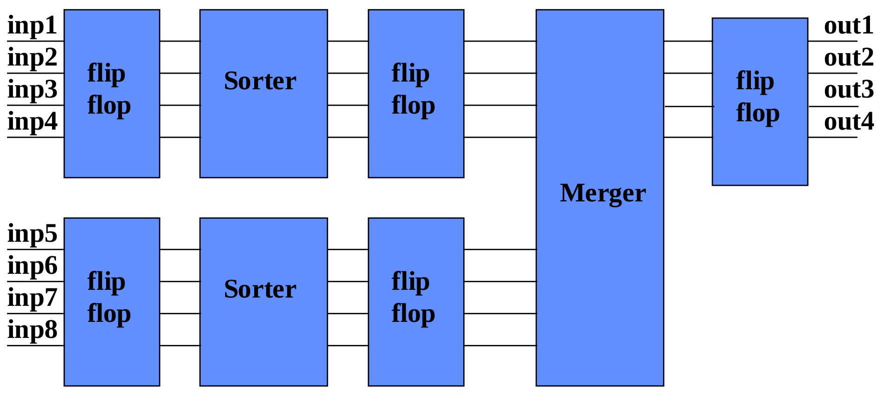

Fig.1
8 inputs Sorting Processor scheme

3. The Comparator

The basic element to built a sorter is the comparator, the faster is the comparator the faster is the sorting process.

Our comparator is formed by a pure combinatorial network, in which two 6 bits input words are compared producing, as a result, a signal *A_GT_B* used by the subsequent component of the structure, a 4 multiplexers chain, to select the greatest, between the two input words.

Such process is performed by the comparator in the worst case, in 3.88 ns.

The output of the comparator, is used afterward, to drive a 4 multiplexers chain which distributes the input words on the Sorting Processor's outputs, in such a way to have on the first 6 bits output the greatest word, in terms of absolute value, between the two input data words (the so called exchange mechanism). At the same time, the Sorter gives in output, also the addresses related to the correspondent data values.

The proposed solution, allows to propagate, along the network, the data and the associated address with no need for additional circuit and with no increase of the time propagation.

4. The 4 Inputs Sorter

Following the literature [5], we have decided to use, as basic structure of our Sorting Processor, an architecture composed by a 2 fold 4 inputs Sorter, capable each to give, as output, the 4 input data values, in a decreasing order, plus one 8 inputs Merger using, for this last one, a scheme of the "odd-even" merge type. The sorter, the structure of which is illustrated in Fig.2, behaves like a comparators network into which any comparator causes an interchange of its inputs if necessary, so that the larger number value appears on the higher horizontal line after passing the comparator. At the right of the diagram all the number outcome in a decreasing order from top to bottom.

The merger, the structure of which is illustrated in Fig.3, behaves exactly as the sorter, only that it needs of input values already ordered in a decreasing order from top to bottom, to give as result the 4 highest number values between its 8 inputs.

The letter indicated on the right of the dot in Fig.2, represents the output of the network at that point.

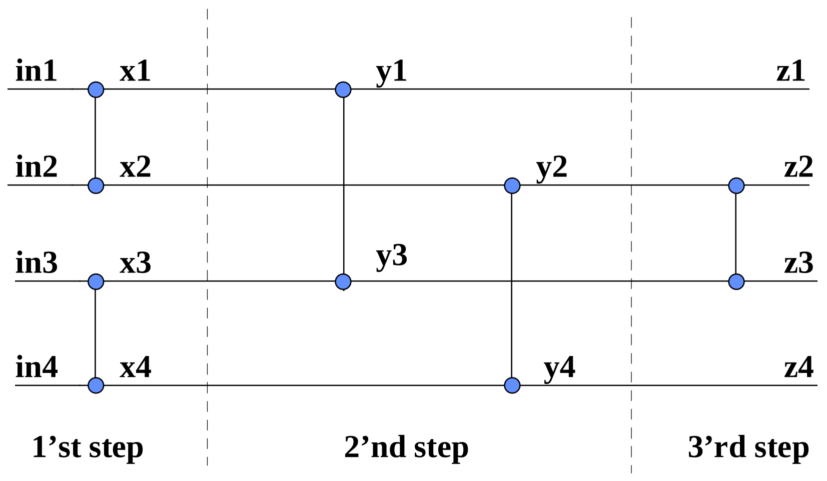

Fig.2
4 inputs sorter scheme
with 3 comparison steps

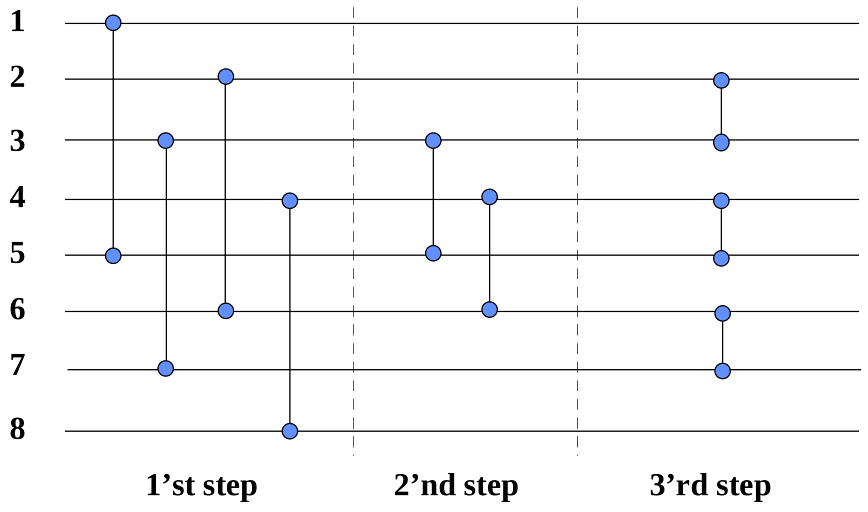

Fig.3

8 inputs merger "odd-even" type
with the comparison steps visualized

Any vertical segment indicated in the Fig.2 and 3, represents the basic element constituting the general Sorter Processor scheme. The representation of this basic element, is shown in Fig.4. In both figures, any horizontal line represents the connection between the input on the left and the output on the right of the comparators of various steps (the dots represent the connection points), through which the input and output data flow. The simulation results have shown that, after transforming the various comparison steps as explained later, the more complex structure using sorter plus merger shown in Fig.1 and proposed as final solution of our processor, is more faster respect to that which uses only sorter circuits.

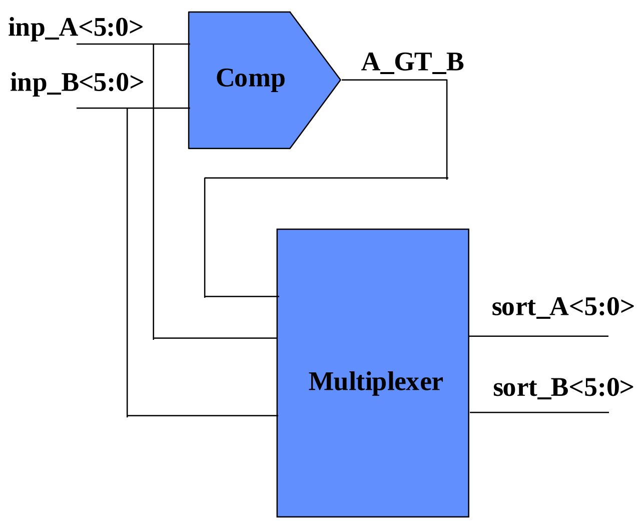

Fig.4
Sorter Processor basic element

5. More about the Sorter

The structure shown in Fig. 2, in which 3 steps of comparisons are need, can be optimized to make the Sorter’s outputs available in shorter time. In fact, if we consider the various steps into which the Sorter’s job is subdivided, we can easily transform the last step of comparison, into a simple “exchange” step into which only a simple multiplexer is enough instead of the more complex structure shown in Fig.3. This process is achieved considering the results obtained from the first step and taking into account the results of the second step. This considerations are explained in a schematic form, in the following figures and tabs, where the C letter has the meaning of the output value (one or zero) of the corresponding comparator represented in the structure shown in Fig.4, depending on its input data values.

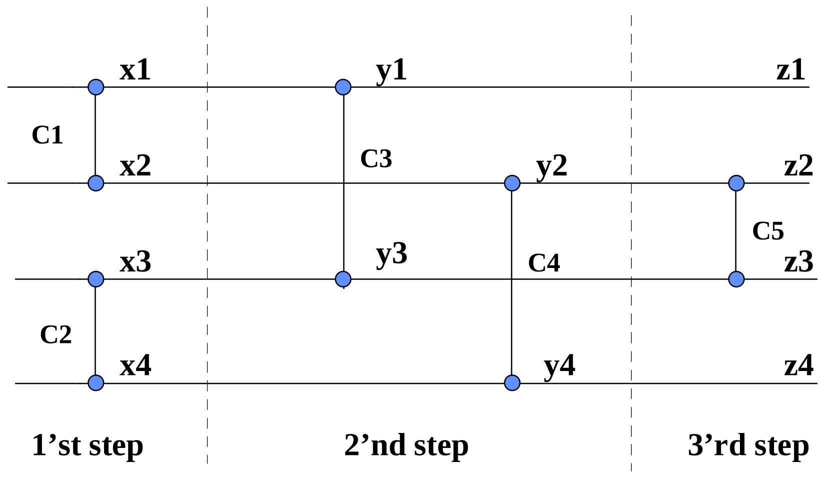

Fig.5
4 inputs sorter scheme with "in-steps" results.
Any letter represents the output of the corrisponding comparator.

$C_i = 1$	does not exchange
$C_i = 0$	does exchange

Tab.1

C_3	C_4	C_5
0	**0**	**$X_4 > X_1$**
0	**1**	**$X_2 > X_1$ (can be fixed to zero)**
1	**0**	**$X_4 > X_3$ (can be fixed to zero)**
1	**1**	**$X_2 > X_3$**

Tab.2

The Tab.1, shows the rule used by the generic comparator into the Sorter. If the result of the comparator is "0", the input data are exchanged, otherwise no action is undertaken and the input values are reported in the same order to the output. By using this rule, if we consider the Fig.5, it is possible to eliminate the last comparator C5 substituting it only by a multiplexer having as inputs, the results of the 2'nd step comparators and having as a control input, the result concerning one only of the four comparisons indicated in the Tab.2, depending by the output of the comparators C3 and C4. Moreover, the two intermediate lines, indicated in the Tab.2, are really redundant, because the corresponding comparisons, are already calculated during the 1'st step. In such a way, we can reduce the 4 inputs Sorter, having only the first two steps of comparisons and the last step reduced to a simple multiplexer stage (exchange step). This reduce furtherly the propagation time of the result of the sorting process.

(Note that in all the tables shown, the > symbol is used to indicate that the operation result depends only on the value entities X_i's present on that line of the table).

The same argument has been adopted when we consider the Merger. In fact, looking to the Fig. 6, the 8 inputs Merger, can be seen as two 4 inputs sorter one below the other, the first one composed (ignoring the first step of comparison as present in the Sorter, which is not present in the Merger) by the comparators C1, C2 and C5 and the other one composed by the comparators C3, C4 and C6.

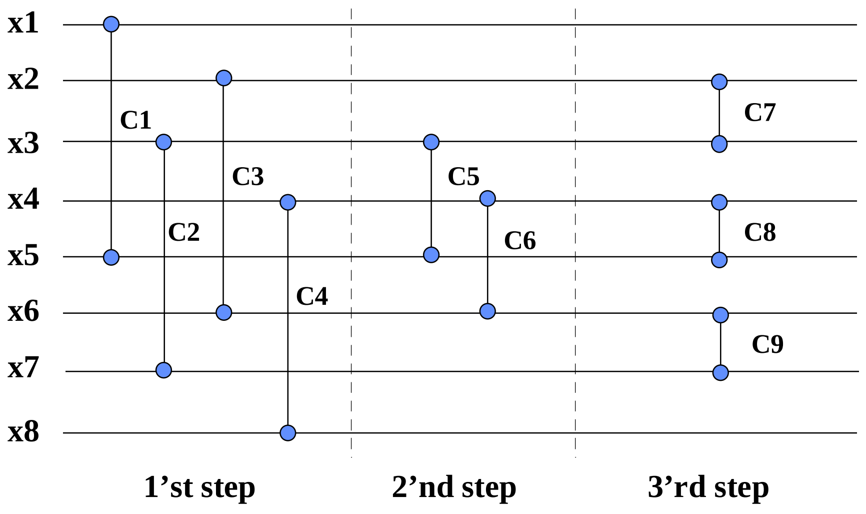

Fig.6

C_1	C_2	C_5	
0	0	$X_7 > X_1$	
0	1	$X_3 > X_1$	can be fixed to zero
1	0	$X_7 > X_5$	can be fixed to zero
1	1	$X_3 > X_5$	

Tab. 3

C_3	C_4	C_6	
0	0	$X_8 > X_2$	
0	1	$X_4 > X_2$	can be fixed to zero
1	0	$X_8 > X_6$	can be fixed to zero
1	1	$X_4 > X_6$	

Tab. 4

In the Tab. 3 and 4, are shown the 8 possible combinations necessary to calculate the C5 and C6 outputs where, excluding the intermediate comparisons already done because coming from the Sorter's outputs, the necessary comparisons to calculate the C5 and C6 outputs, are reduced to 4.

To calculate C7 and C8 outputs, we proceed in the same way as before, taking into account all the possible combinations shown in the following Tables, for the different possible values of C5 and C6 . The combinations for C9 are not taken into account, because we are interested only at the first 4 outputs of the Merger.

C_1	C_2	C_7
0	0	$X_6 > X_1$
0	1	$X_2 > X_1$ can be fixed to zero
1	0	$X_6 > X_5$ can be fixed to zero
1	1	$X_2 > X_5$

Tab. 5
Possible value of C7 for C5=0

C_2	C_3	C_7
0	0	$X_6 > X_7$ can be fixed to one
0	1	$X_2 > X_7$
1	0	$X_6 > X_3$
1	1	$X_2 > X_3$ can be fixed to one

Tab. 6
Possible value of C7 for C5=1

C_2	C_3	C_8
0	0	$X_2 > X_7$
0	1	$X_2 > X_3$ can be fixed to one
1	0	$X_6 > X_7$ can be fixed to one
1	1	$X_6 > X_3$

Tab. 7
Possible value of C8 for C5=0 and C6=0

C_3	C_4	C_8
0	0	$X_8 > X_7$ can be fixed to zero
0	1	$X_4 > X_7$
1	0	$X_8 > X_3$
1	1	$X_4 > X_3$ can be fixed to zero

Tab. 8
Possible value of C8 for C5=0 and C6=1

C_1	C_2	C_8
0	0	$X_2 > X_1$ **can be fixed to zero**
0	1	$X_6 > X_1$
1	0	$X_2 > X_5$
1	1	$X_6 > X_5$ **can be fixed to zero**

Tab. 9
Possible value of C8 for C5=1 and C6=0

C_1	C_4	C_8
0	0	$X_8 > X_1$
0	1	$X_4 > X_1$ **can be fixed to zero**
1	0	$X_8 > X_5$ **can be fixed to zero**
1	1	$X_4 > X_5$

Tab. 10
Possible value of C8 for C5=1 and C6=1

Examining in more detail all the combinations shown in the Tables from 5 to 10, we can see that some of them are superfluous from the circuit point of view, because the corresponding comparisons, are already calculated in some previous step. To explain this, we take as example the first and forth combinations shown in Tab. 6.

Here, looking at Fig. 6, we see that the two comparisons X6>X7 and X2>X3, can be excluded from the circuit and their outputs can be substituted by “one”. The reason of this, relays on the fact that these two combinations are already evaluated from the circuit and therefore we substitute them by one because, in this case, the comparator C7 must not exchange its inputs.

By looking at all the combinations, we can conclude that the Merger is constituted by 16 comparators plus 14 Mux’es, but that all the possible and useful comparisons are done only at the first step of the merge process, gaining a lot of calculation time. In fact arranging the Merger as described, we obtain only 5.41ns, as worst set-up time for the calculation of the overall 8 data input sorting process.

6. The Sorting Processor Chip

We can then conclude that our Sorting Processors, is a chip made by two 4 inputs Sorters, plus an 8 inputs Merger. It has a total of 168 pins, considering only the signals pins, so distributed:

• inputs	8 data x 6 bits = 48
	8 address x 8 bits = 64
• outputs	4 data x 6 bits = 24
	4 address x 8 bits = 32.

It will contain all the circuitry necessary to make “Boundary Scan”, for the chip remote test.

7. The Ring Processor

Our final goal is to build an electronic board capable to sort the 144 inputs coming from a “ring”, that is a piece of the muon detector subtending an angle of 360° in φ and interesting an angular region, along the beam direction, corresponding to Δη ≈ 0.11. To do this, we use a total of 39 chips, which will furnish the final result, in a total of 7 time slices. In fact if we see the result of the simulation shown in the Fig.7, the structure of the circuit equivalent to a one Ring Processor, gives the outputs, after 182.9 ns, since the application of the inputs.

In this figure the input signals coming from only 16 data sources, are shown as ‘*Ini*’, while the 4 output codes are indicated as ‘*OUTi*’. The timing interval between any input values group and its corresponding output, is just 182.9 ns as indicated in the simulation output.

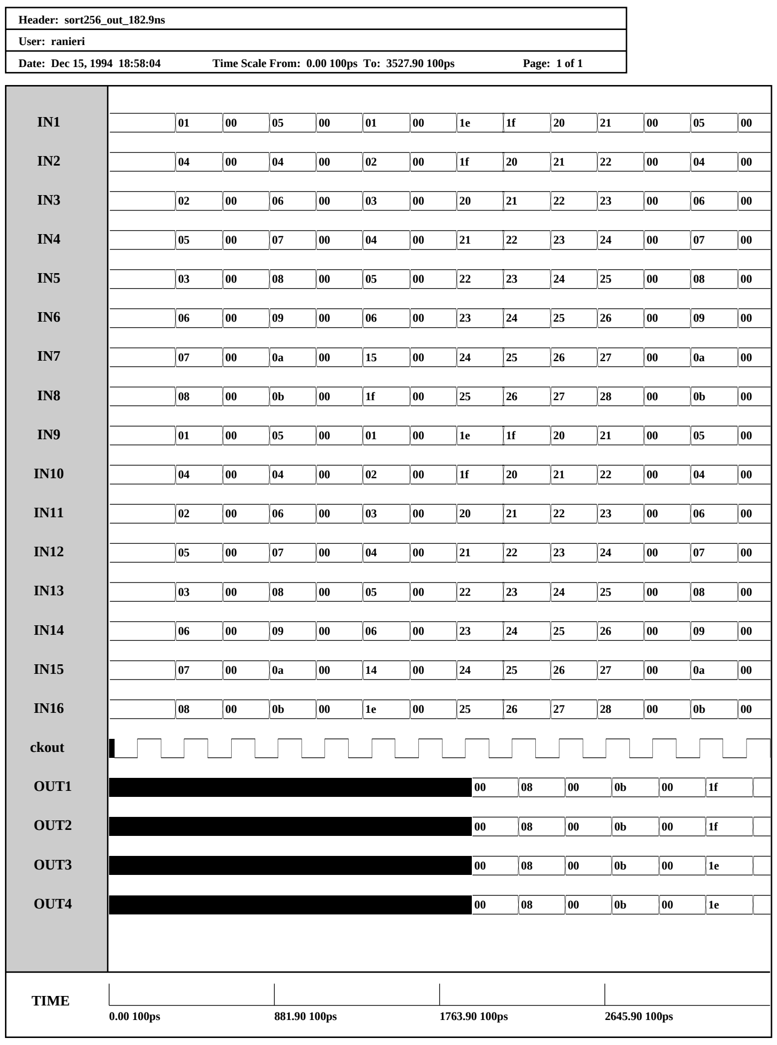

Fig.7
Ring Processor timing simulation

The structure of the Ring Processor is shown in the Fig.8, where only a part of the overall circuit is illustrated. Note that the inputs, are strobed, by using the system clock, into the circuit, in a first stage latch, just to synchronize the incoming data. For all the chips in the network different from those at the first level, the unique part involved in the calculation is the Merger part. This means that the Ring Processor, will use the Sorters only for the chips at the first level. For the others stages instead, the Sorters are excluded by using appropriate mux'es and the data will propagate through the network, directly to the Mergers. Onto one Ring processor board, we will have a total of 39 chips and due to the muon trigger segmentation of the overall apparatus, we aspect to have a total of 44x39=1716 chips.

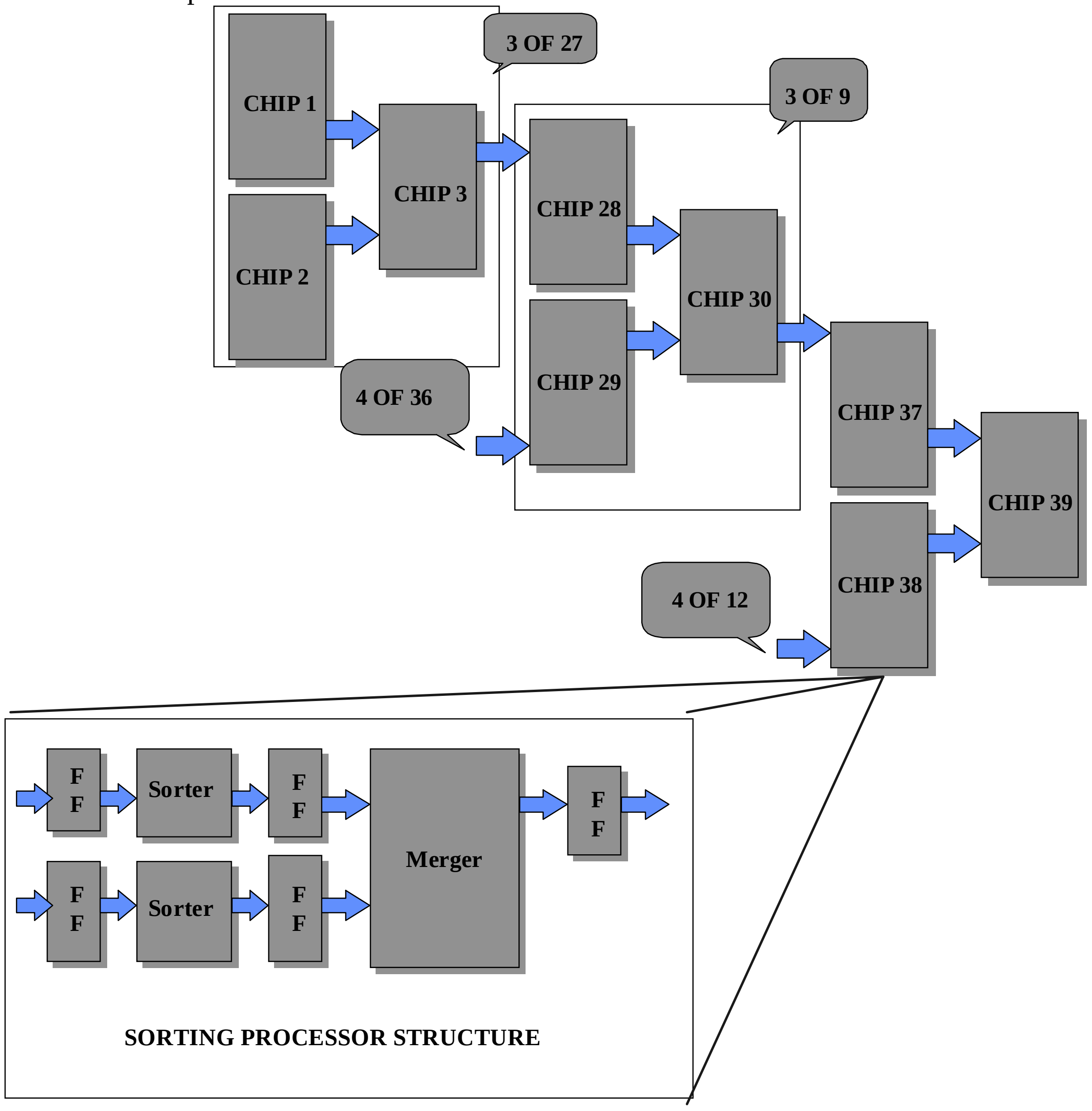

Fig.8
Ring Processor structure

References

[1] **R.Cardarelli and R.Santonico**: *Development of Resistive Plate Counter*. Nucl.Instr. and Meth. A 187 (1981) 377.
[2] **R.Cardarelli and R.Santonico**: *Progress in Resistive Plate Counter*. Nucl.Instr. and Meth. A 263 (1988) 20.
[3] **M.Gòrski**: *Warsaw RPC test results.* Talks given at RD5 Collaboration Meetings, May 27 and June 14,1993.
[4] **G.Wrochna et al.** *Pattern Comparator Trigger (PACT) for the Muon System of the CMS Experiment.* CMS-TN/94-281 (1994).
[5] **D.E.Knuth.** *The Art of Computer Programming. Vol.3 Sorting and Searching.* Addison-Wesley Publishing Company.