Available on CMS information server **CMS IN 2001/xxx**

CMS Internal Note

The content of this note is intended for CMS internal use and distribution only

Draft of 19 January 2001

PAttern Comparator (PAC) ASIC version 2 Specification

Jan Królikowski[a)], Ignacy Maciej Kudla[a)]

Warsaw University

Grzegorz Wrochna[b)], Piotr Zalewskib[a)]

Soltan Institute for Nuclear Studies

Zbigniew Jaworski[a)], Wieslaw Kuzmicz[a)], Elzbieta Piwowarska[a)]

Warsaw University of Technology

Marcin Konecki [a,c)]

CERN, Geneva, Switzerland

Abstract

The detailed functional description of PAC ASIC version 2.0 processor for the RPC Muon Trigger is given.

a. Partially supported by the Polish Committee for Scientific Research under grant KBN 115/E-343/SBUP/P03/057/98 and KBN 115/E-343/SPUB/P-03/DZ 8/99.
b. Partially supported by the Polish Committee for Scientific Research under grant KBN 115/E-343/SBUP/P03/057/98 and KBN 621/E-78/SPUB/P-03/DZ 5/99.
c. On leave from Warsaw Uniersity, Warsaw, Poland.

1 Introduction

This specification describes version 2.0 of PAttern Comparator (PAC) processor ASIC, a segment processor for the RPC Muon Trigger.

General principles of PAC architecture has been described in CMS notes and conference talks [1],[2],[4],[6], and we will only briefly recall them here. The processor finds a spatial and temporal coincidence of hits in four RPC planes, the second RPC plane is used there as a reference plane. The spatial pattern of hits is programmable. Not all possible patterns of hits are available in the chip; we had to restrict both the range of available patterns i.e. the number of strips in stations 1, 3 and 4 connected to a given strip in the reference plane (called cones), and their number i.e. the maximum number of combinations of hits in a given cone to which we may assign a momentum code (to 80 patterns for each positive track going through a given strip in the reference plane, and the same number for the negative ones). The restriction in the range and the number of patterns would be a severe limitation, esspecialy for low momentum tracks. Fortunately, full spatial resolution is not required for those tracks and they can be handled with coarser granularity obtained by logical OR of several strips. In fact, the simulation have shown that we need single, double, quadruple and octuple strip logical signals in the PAC processors with 80+80 programmable track patterns for each reference strip (OR1, OR2, OR4 and OR8).

Prototype version 1.0 of PAC has been implemented in ATMEL 0.7µm technology in 1997/98 [3], and successfully tested. In this version, the formation of all logical OR signals was done outside of the PAC processor on the trigger board. The processor itself used as inputs various OR-ed signals. For technological reasons it was deemed impossible to combine all OR-ed logic in one chip, and we envisaged several PAC processor in a cascade, each of them working with a given level of OR-ed input signals. We decided that every chip should cover four reference strips. Therefore, a basic logical unit of the RPC trigger - the segment of eight strips in the reference plane and their cones would require 5 to 6 such PAC processors in a cascade (two for OR1, one for OR2, one for OR4 one for OR8 and one for low transverse momenta in the barrel).

The second version of PAC contains several improvements and modifications in comparison with the first prototype. These improvements mostly result from better technology; three metal 0.35 µm AMS technology used. The basic conceptual improvement is the dynamical cone concept, which we would like to describe below, and which will restrict both the range and the number of needed patterns.

Straight, high momentum tracks going through a given strip in the reference plane occupy the middle of a cone subtended on the reference strip. Strongly curved, low momentum tracks are on the outside of a cone. Therefore, a dynamical gradation of strip granularity within the cone, OR1 inside, then OR2 and OR4 and only some OR8 on the outside may result in patterns which have e.g. OR2 in station 1, OR1 in the reference plane, OR4 in the third and fourth stations and to which a momentum of e.g: 5 GeV/c will be assigned. By optimizing this design we may end up with cones with smaller number of connected logical strips, and smaller number of patterns. The simulation was done and carefully compared with the previous PAC design and this conjecture was proven to be true [5].

The most important implications of the dynamical cone concept and new 0.35 µm technology for the version 2.0 design are twofold:

- there is no need for a cascade of processors, since the dynamical cone comprises all needed OR-ed signals. Moreover, it comprises the (normal) high transverse momentum (hpt) function and the special, restricted really to the barrel low transverse momentum (lpt) one. This will make the trigger board simpler and it will decrease the overall cost.
- the logical OR signals may be produced inside processor, increasing the reliability of the trigger system.

2 General description

PAC version 2.0 fulfils the task of the segment processor, it allows to program the hpt and lpt muon patterns and provides the code of the highest momentum muon in the segment area. PAC is a pipelined, synchronous device - output code corresponds to a muon candidate found in input data 3 clock periods before. The structure of PAC input and output signals is shown in Fig. 1 The term “layer” means signals from a given station of the RPCs. The logical layer corresponds to different RPC stations in different parts of the CMS detector.

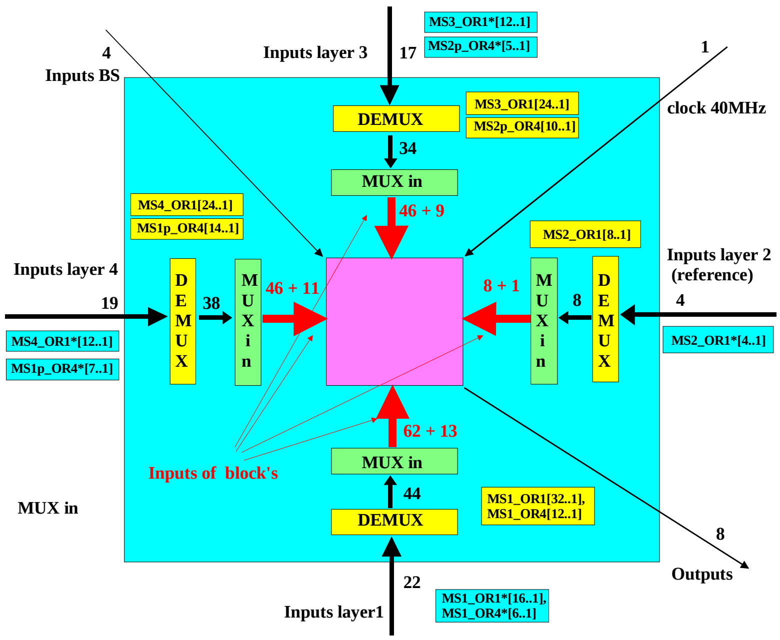

Figure 1: Input and output structure of PAC version 2.0

To keep the number of PAC input signals as small as possible input signals are passed into PAC using both edges of a clock signal, and all necessary OR2, OR3, OR4, OR8 signals are calculated inside the chip from the RPC single strip inputs. Input part is described in details in Sec. 2.1.

The code output consists of the muon code information (4 bits), muon sign bit (1 bit) and muon quality (2 bits).

Fig. 2 shows the structure of main part of PAC, which logically consists of five units: track recognition, quality selection, track grouping, code programming and code selection.

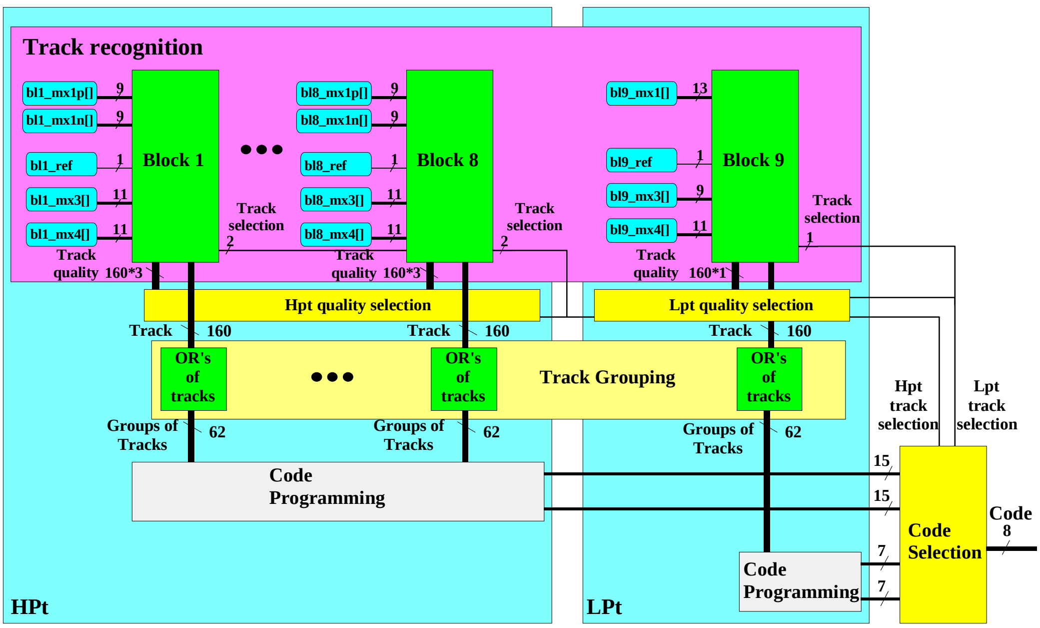

Figure 2: Structure of PAC ASIC version 2.0

PAC contains 9 track recognition blocks (see Sec. 2.2). Blocks 1-8 are used to recognize hpt muon tracks (see Sec. 2.2), each of them corresponds to one reference strip. Block 9 corresponds to OR of all 8 signals from reference layer and is used for low p_T track recognition. Its use and meaning is different in the barrel and in the endcaps. At the output the highest momentum code is presented with corresponding muon sign and quality bits. Muon tracks are recognized if they corresponds to programmed patterns (see Sec. 2.2). Up to 9*(80+80) muon patterns can be programmed into PAC. Groups of programmed patterns are described by a programmable code (see Sec. 2.4). Muon patterns and codes are programmed in the PAC by usage of a Boundary Scan (BST) circuit (see Sec. 2.6). In the code selection part of PAC (see Sec. 2.5), the momentum codes can be redefined in the Look Up Table Memories, separately for hpt codes coming from blocks (1-8) and lpt ones from block 9.

In the segment at most one candidate track - that of the highest momentum - is selected. It turn out that it is difficault to try and give out the information on the predefined pattern which matches the actual RPCs hits. We decided that substantial complication of, the already involved enough, logic needed for that task is too risky. Thus, this information is not given out; we have no means of knowing, which pattern actually fired the PAC and produced the fixed momentum code.

The selection of one muon code from the PAC is done in 3 steps:

- we select out off all possible track candidates found inside PAC those with the highest quality, separately for hpt and lpt. The determination of the candidate track quality is based on the number of the RPC hits (4 out of 4 or 3 out of 4) on their distribution (see Sec. 2.2 for details).
- from among tracks with the highest quality, we select one track with the highest momentum, again separately for hpt and lpt.
- we decide whether a hpt or lpt candidate is the better one.

2.1. Input circuits

PAC has 62 data input pins, 1 clock input pin and 5 JTAG input pins. In this section we describe data input signals, JTAG will be described in Sec. 2.6. All PAC input data are registered at both edges of clock signal. Table 1 shows the PAC pins. All demultiplexed PAC input data signals can be masked. When mask bit is zero, the corresponding internal data input is set to zero as well. Mask bits are loaded via a JTAG interface (see Sec. 2.6). Figure 3 and Figure 4, show the sequence of operations in the input part of PAC, for the signals from layer 1, 2 (reference), 3 and 4.

Table 1: PAC data input pins

PAC data input pins			Internal PAC input signals (demultiplexed data)	Usage	
Input data (multiplexed)	Positive edge data	Negative edge data		hpt	lpt
Ms1_or1*[16..1]	Ms1_or1[32..2]	Ms1_or1[31..1]	Ms1_or1[32..1]	layer1	layer4
Ms2_or1*[4..1]	Ms2_or1[8..2]	Ms2_or1[7..1]	Ms2_or1[8..1]	reference	reference
Ms3_or1*[12..1]	Ms3_or1[24..2]	Ms3_or1[23..1]	Ms3_or1[24..1]	layer3	not used
Ms4_or1*[12..1]	Ms4_or1[24..2]	Ms4_or1[23..1]	Ms4_or1[24..1]	layer4	not used
Ms1_or4*[6..1]	Ms1_or4[12..2]	Ms1_or4[11..1]	Ms1_or4[12..1]	not used	layer4
Ms1p_or4*[7..1]	Ms1p_or4[14..2]	Ms1p_or4[13..1]	Ms1p_or4[14..1]	not used	layer3
Ms2p_or4*[5..1]	Ms2p_or4[10..2]	Ms2p_or4[9..1]	Ms2p_or4[10..1]	not used	layer1

Demultiplexer registers and masking circuit are followed by a circuit, where the signals needed for 9 blocks of PAC are prepared (for details, see Tables 3-9 in the Appendix). Inputs from layer 1 multiplexers in blocks 1-8 form an asymmetric cone, all other cones are symmetric. Figures 13-19 shows signal cones for all layers.

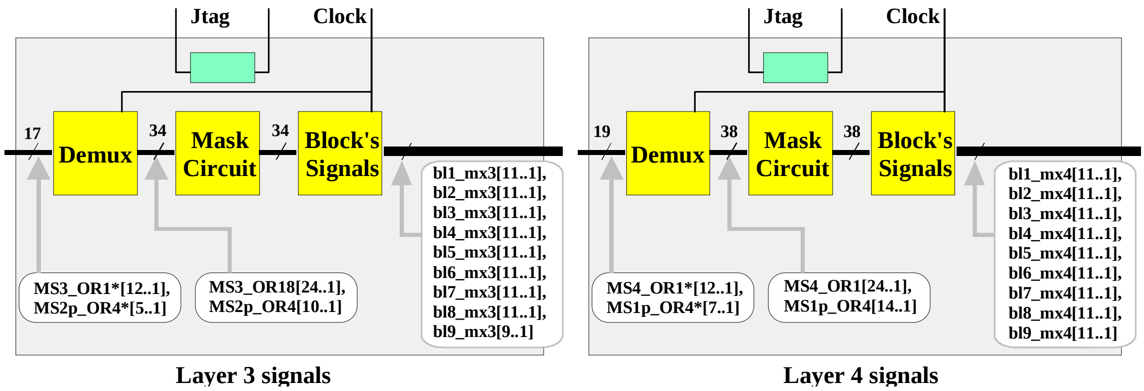

Figure 3: Input circuits for layer 1 and 2 (reference) multipexer signals

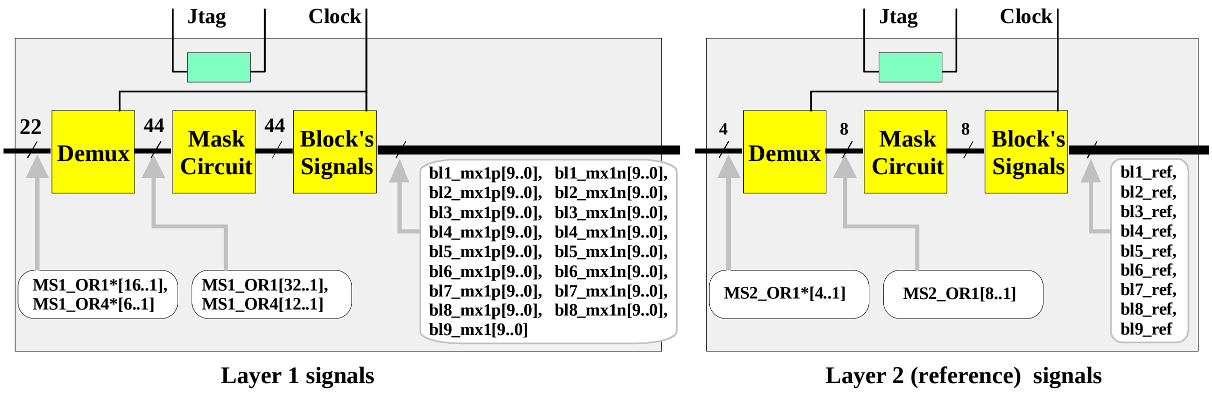

Figure 4: Input circuits for layer 3 and 4 multiplexers signals

Table 2 give sizes of multiplexer input busses:

Table 2: Multiplexer (mux) bus sizes

Layer	Block	Mux bus size
1	1-8	10
1	9	14
3	1-8	12
3	9	10
4	1-8	12
4	9	12

2.2. Track recognition circuits

This is the main part of PAC. It allows to program and recognize up to 1440 muon tracks (pre-defined patterns). It is divided into 9 track recognition blocks (Figure 5 and 6). Each track recognition block can recognize up to 160 muon tracks. Blocks 1-8 are used to recognize hpt muon tracks. Blocks 1-8 corresponds to a single reference strip each, block 9 corresponds to OR of all 8 signals from reference layer and allows to recognize strongly bent, low pt muon tracks.

Each block allows for simultanous, parallel recognition of 80 positive and 80 negative tracks. Inputs signals for a block are prepared in input circuits. Each block has its exclusive reference signal (from the second RPC layer). The predefined Muon pattern is a set of 3 numbers corresponding to the selection signals for multiplexers mx1 (layer 1), mx3 (layer 2) and mx4 (layer 4). These numbers are written to the 4 bit mux selection registers, which

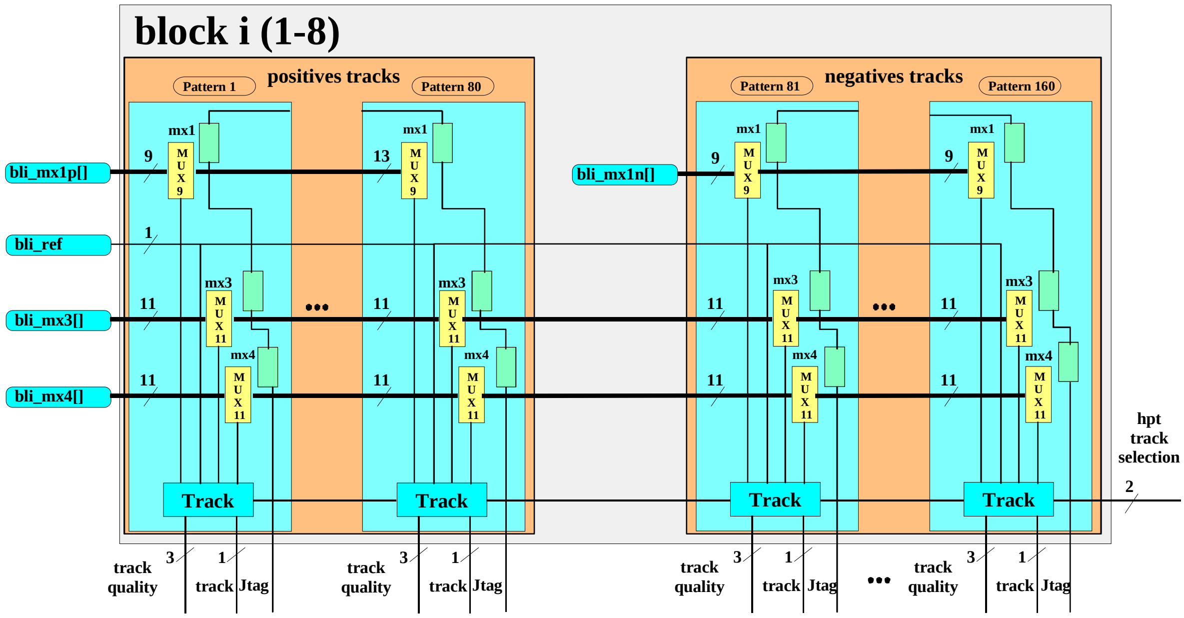

Figure 5: Track recognition blocks 1-8

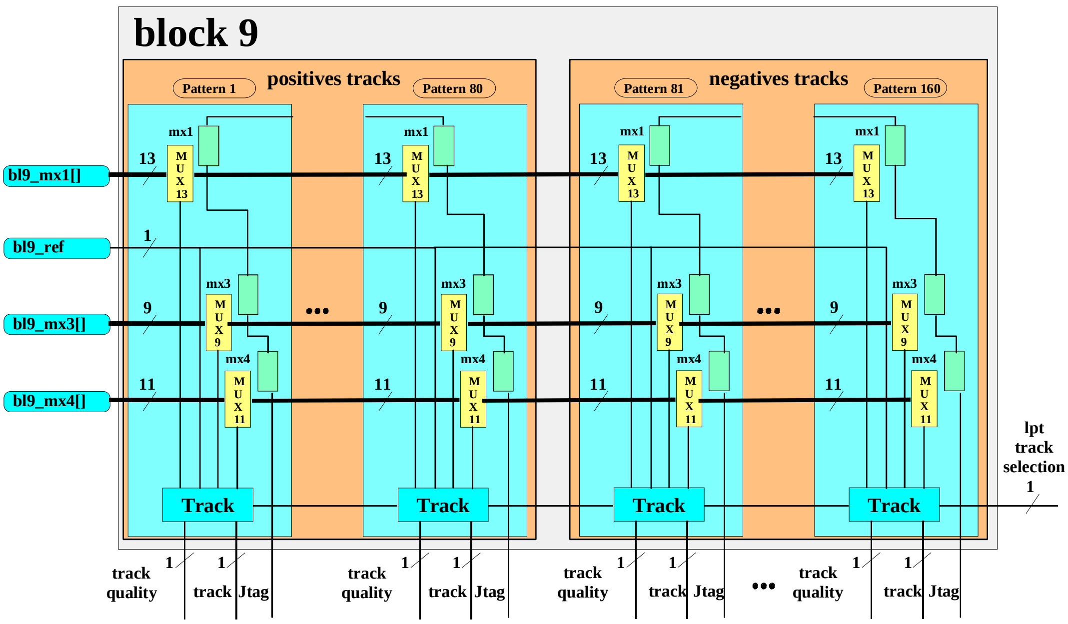

Figure 6: Track recognition block9

are accessible via JTAG. The input signal to a multiplexer (hit) - corresponding to the unique pre-programmed selection for this multiplexer is transmitted to the track circuit. Track circuit for one pre-defined pattern is shown in Figure 7. Its inputs are mx1, mx3, mx4 and reference layer strip signal. There are 1440 track circuits (pre-defined patterns in PAC ASIC. Track circuits for blocks 1-8 are different from that for block 9. Each track circuit provides track quality information, separately for blocks 1-8 and 9. Track quality circuit charactrizes the track candidate according to the following priority list (for blocks 1-8):

- coincidence of 4 signals from 4 layers (quality = 3);
- coincidences of 3 signals from layers 1,2,3 or from layers 1,2,4 (quality = 2);
- coincidence of 3 signals from layers 2,3,4 (quality = 1);
- coincidence of 3 signals from layers 1,3,4 (quality = 0).

For block 9 equivalent priority list is as following:

- coincidence of 4 signals from 4 layers (quality = 1);
- coincidences of 3 signals from layers 1,2,3,4 (quality = 0).

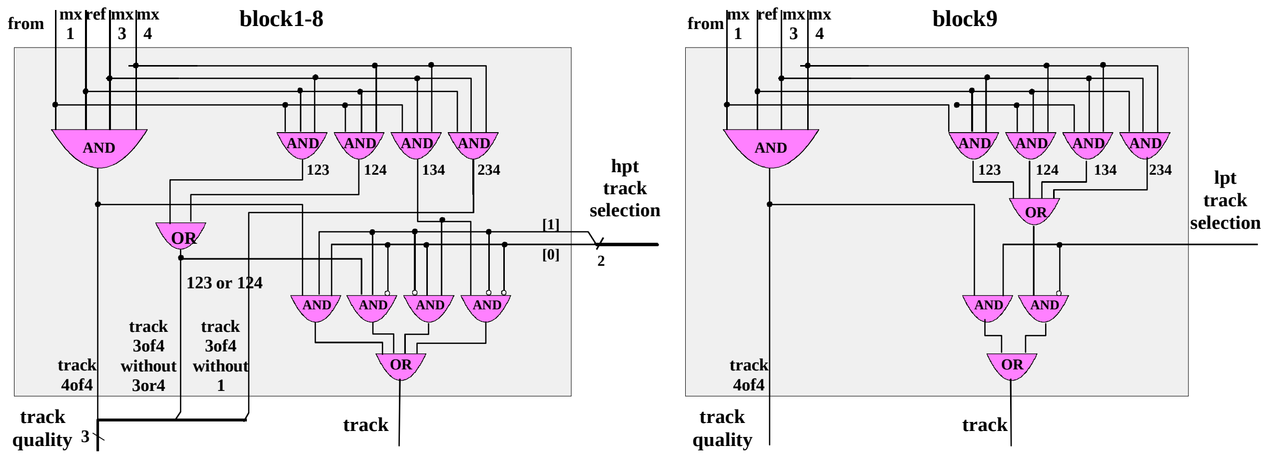

Figure 7: Track circuit for block 1-8 and 9

2.3. Quality selection

The purpose of this stage is to determine the highest quality of the tracks found by a PAC, separately for high p_T (blocks 1-8) and low p_T (block 9). After this highest quality is determined, the tracks which have lower quality are discarded. Then, the next stage (Sec. 2.5) - code selection is done only for tracks with the highest quality code (separately for high p_T and low p_T ones).

Figure 8 shows track quality selection circuits for blocks 1-8 and block 9. It finds the best quality track candidate and returns its code to track circuits to disable all track candidates with quality lower than selected.

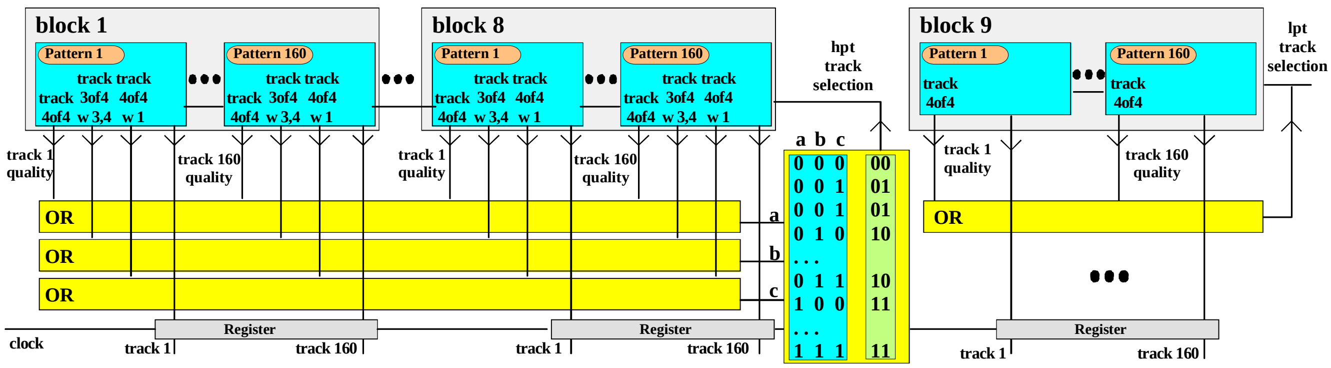

Figure 8: Quality selection circuits

The track selection code selects the coincidence to be passed to the output of tracks circuit. At this stage PAC delivers 1440 track signals. All track signals are latched at the end of this stage.

2.4. Track grouping and code programming circuit

For technical reasons (size of die) it was necessary to group together several tracks, for the purpose of subsequent momentum coding. The aim of track grouping and code programming circuit is to define the code belonging to the muon track or group of muon track signals. Track grouping circuit (Figure 9) combines the muon tracks into groups sharing the same track code. The number of tracks which can be individually programmed into specific momentum code is reduced from 1440 to 558. Table 8 and 9 in the Appendix show how the muon track signals are grouped for every track recognition block. The grouping is not programmable - tracks which are asummed to be dedicated to the same code should be programmed into patterns belonging to the same group.

Each group of tracks signal for blocks 1-8 can be programmed to one of 15 positive or negative codes. Each group

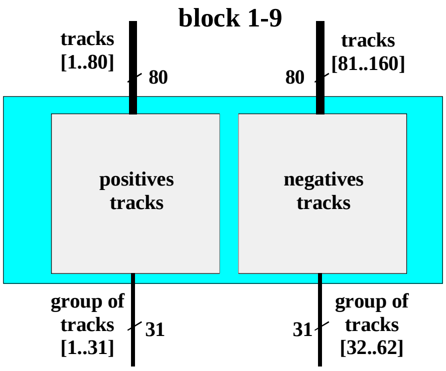

Figure 9: Track grouping

of tracks signal for block 9 can be programmed to one of 7 positive or negative codes (Figure 10).

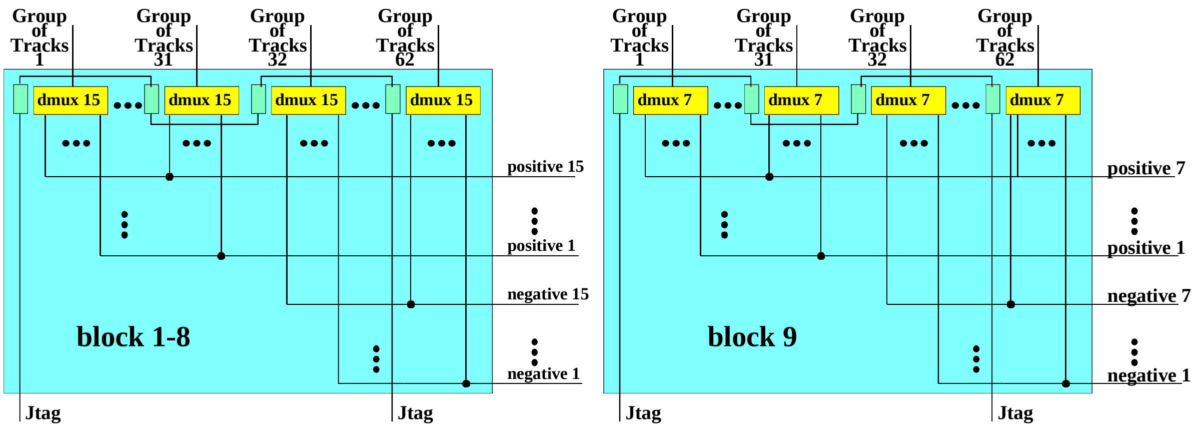

Figure 10: Code programming

Code programming circuit for block 1-8 consists of 1 to15 demultiplexers and 4 bit selection registers. Only one output line of the demultiplexers can be selected. Selection registers are programmed by JTAG interface. Code programming circuit for track recognition block 10 consist of 1 to 7 demultiplexers and 3 bit selection registers. Only one output line of the demultiplexers can be selected. Selection registers are programmed by JTAG interface.

All corresponding demultiplexer outputs for positive and negative groups of tracks are OR-ed together for blocks 1-8. They are kept separate for block 9. Four signal busses - hpt_positive [15..1], hpt_negative [15..1], lpt_positive [7..1], lpt_negative [7..1], contain the information on the codes of Muon tracks found by the PAC. No information is provided on the hit pattern of the selected track.

2.5. Code selection

Four separate code_of_the_biggest circuits finds the four highest codes on hpt_positive[15..1], hpt_negative[15..1], lpt_positive[7..1], lpt_negative[7..1] busses (Figure 11). Sign bits of positive and negative codes are programmable via JTAG.

The Higher_only circuits, hpt and lpt selects the highest momentum code from positive and negative tracks, separately for high and low p_T and appends their signs. LUT4 transforms 4 bit hpt code to 5 bit hpt code and LUT3 circuit transforms 3bit lpt code to 4bit lpt code. Programmable (via JTAG) LUT4 and LUT3 circuits allow us to define any kind of 4 to 5 bit (and 3 to 4) transformation functions.

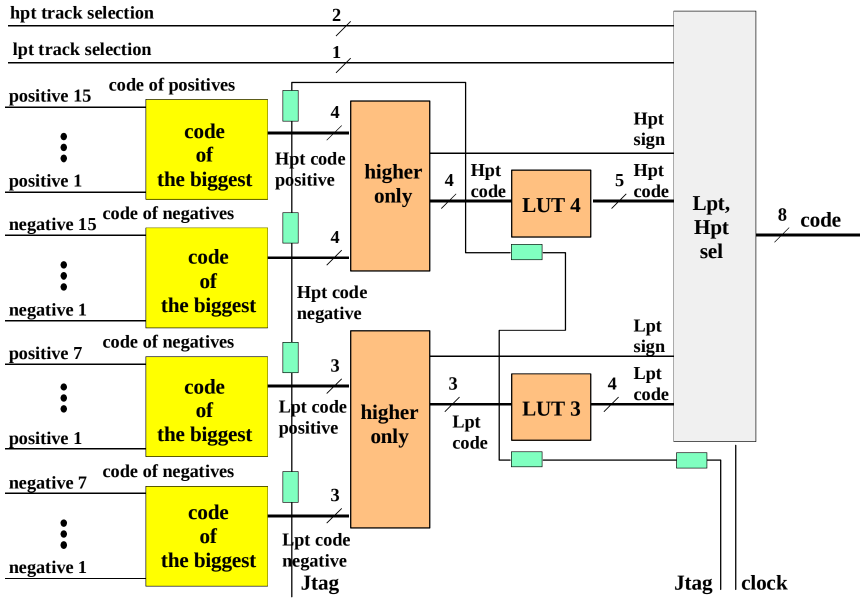

Figure 11: Code selection circuit

The final block Hpt_lpt_sel circuit selects the best track between hpt or lpt ones with its sign and quality bits. The selection is based on the quality bits and value of the selection bit (programmable via JTAG). This allows to differ the hpt, lpt selection in barrel and endcap regions (Figure 12).

```
BEGIN
 IF ((lpt_code(3 downto 0) /= "0000") and
           ((hpt_code(4 downto 0) = "00000") or
           ((flag = '0') and
              (hpt_code(4 downto 0) /= "00000") and
              (hpt_track_selection(1 downto 0) /= "11")  and
              (lpt_track_selection = '1')))) THEN
   code(7) <= lpt_sign;
   code(6) <= lpt_track_selection;
   code(5) <= lpt_track_selection;
   code(4) <= '0';
   code(3 downto 0) <= lpt_code(3 downto 0);
 ELSE
   code(7) <= hpt_sign;
   code(6 downto 5) <= hpt_track_selection(1 downto 0);
   code(4 downto 0) <= hpt_code(4 downto 0);
 END IF;
END PROCESS;
```

Figure 12: Hpt, lpt selection circuit - VHDL code

Finally output code signals are latched.

2.6. Boundary scan circuit

JTAG (IEEE 1149.1 standard) interface is used to program the patterns, codes and other selection registers to access terminals and internal registers.

EXTEST, INTEST, BYPASS, SAMPLE, IDCODE commands are provided.

10 data registers are provided for programming of PAC functions:

- i/o terminals (pins) 75 bits,
- block1 programming 2*80*4 bits + 2 *31*4 bits = 888 bits,
- block2 programming 2*80*4 bits + 2 *31*4 bits = 888 bits,

- block3 programming 2*80*4 bits + 2 *31*4 bits = 888 bits,
- block4 programming 2*80*4 bits + 2 *31*4 bits = 888 bits,
- block5 programming 2*80*4 bits + 2 *31*4 bits = 888 bits,
- block6 programming 2*80*4 bits + 2 *31*4 bits = 888 bits,
- block7 programming 2*80*4 bits + 2 *31*4 bits = 888 bits,
- block8 programming 2*80*4 bits + 2 *31*4 bits = 888 bits,
- block9 programming 2*80*4 bits + 2 *31*3 bits = 826 bits,
- masks, LUTs. signs, selection bits programming 124 bits + 16*5 bits + 8*4 bits + 5 bits = 241 bits,
- internal test register (groups of tracks) 9*62 bits = 558 bits,

2.7. Electrical and other specifications

3.3 V technology, 5V tolerant inputs. Positive logic for input and output signals and for selection registers.

Package - PGA120. Set-up and hold time less then 2 ns. Fan-in and fan-out. - 2. Frequency 40 MHz.

Acknowledgment

The authors are grateful to M.Gorski and K.Pozniak for stimulating discussions.

References

[1] I. Kudla *et al.*, *Trigger electronics for Muon RPC Chambers in CMS* , First LEB Workshop, Lisbon 1995 (218-221).
[2] Z. Jaworski *et al.*, *Resistive Plate Chamber (RPC) based muon trigger system for the CMS experiment - pattern comparator ASIC* , **NIM A 419 (1998) 707-710**.
[3] E. Piwowarska *et al.*, PAC prototype ASIC for the CMS RPC muon trigger, **CMS IN 1999/021**.
[4] G. Iaselli *et al*., *Study of Detailed Geometry of Barrel RPC Strips*, **CMS IN 2000/044**.
[5] G. Bruno *et al.*, *RPC System Geometry Simulated in CMSIM 118-120 and ORCA 4.2*, **CMS IN 2000/054**.
[6] *CMS Trigger TDR*, **CERN/LHCC 2000 - 38**.

Appendix

The following figures give details of the input configuration for block 1-8

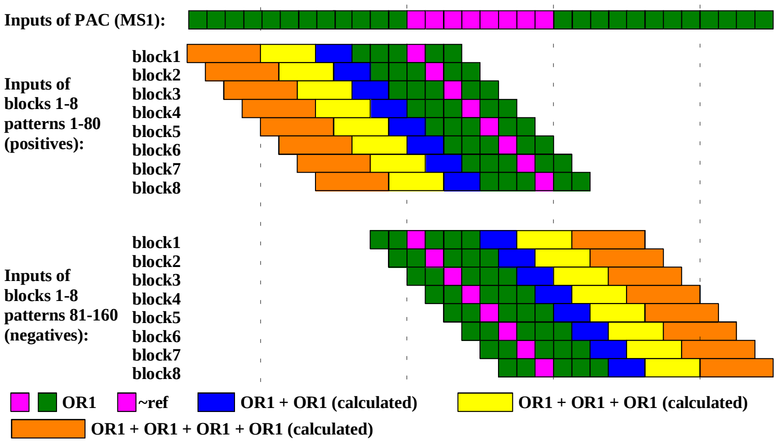

Figure 13: Configuration of layer 1 inputs

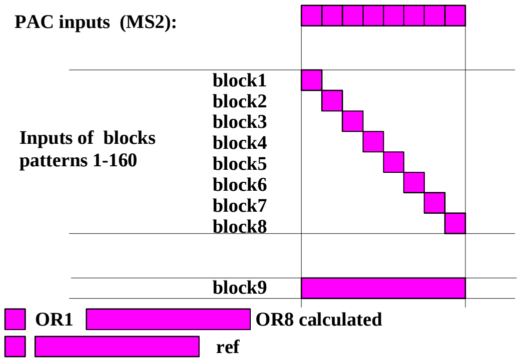

Figure 14: Input signals for block 1-8 and 9, configuration of layer 2 (reference)

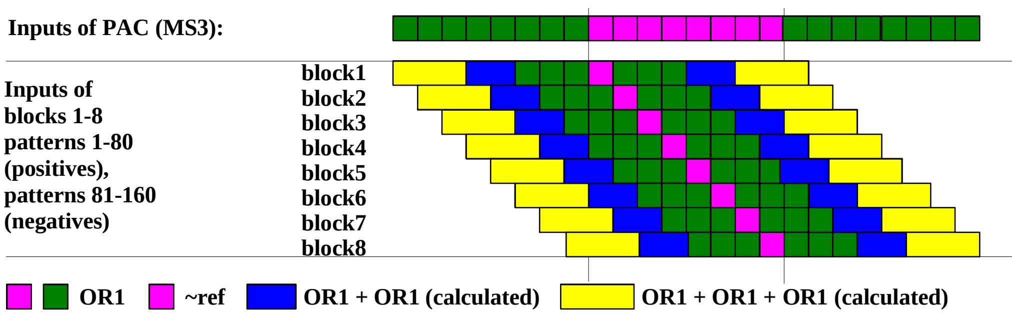

Figure 15: Configuration of layer 3 inputs

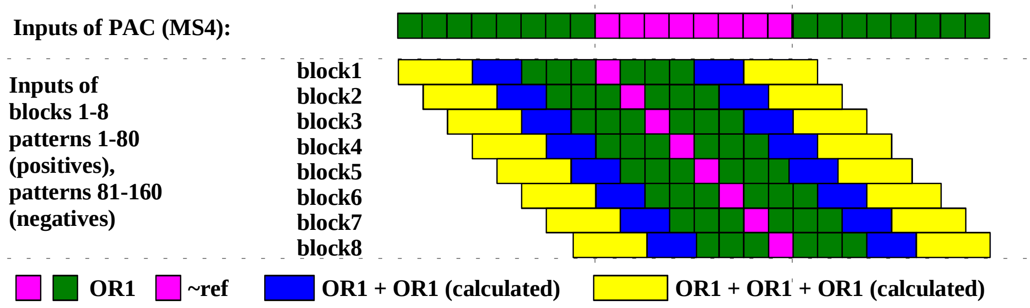

Figure 16: Configuration of layer 4 inputs

The following figures give details of the input configuration for block 9

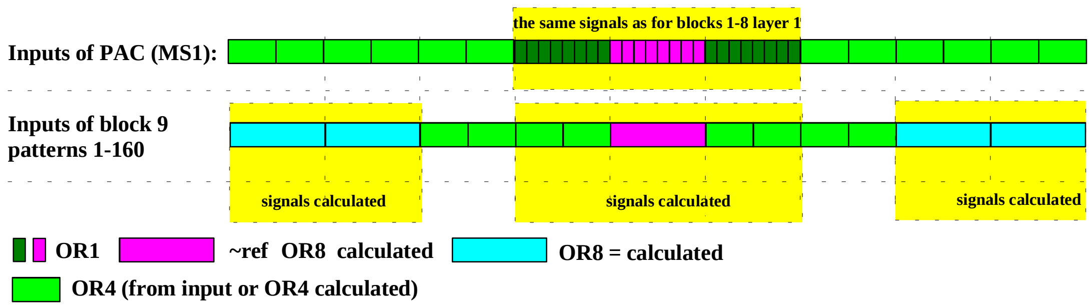

Figure 17: Configuration of layer 1 inputs

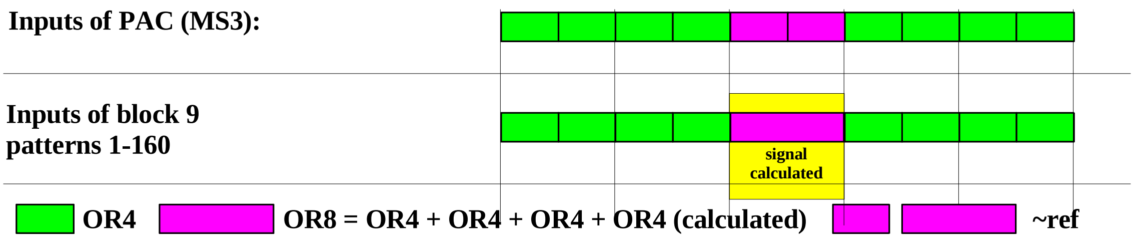

Figure 18: Configuration of layer 3 inputs

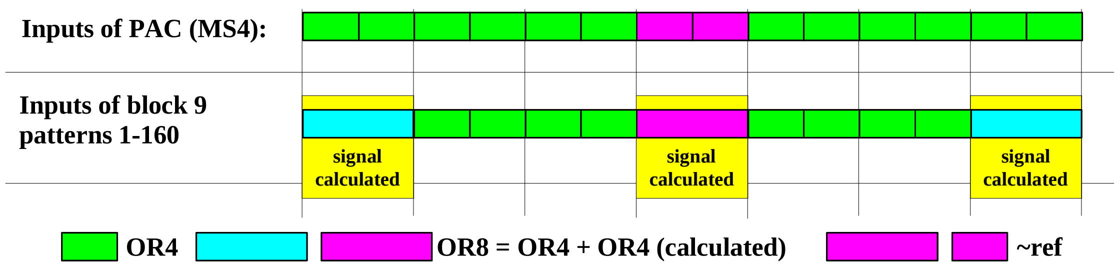

Figure 19: Layer 4 signal cone block 9

The following figures give details of the input configuration for block 1-8

Table 3: Layer 1 multiplexers positive patterns inputs

	Layer 1 negative patterns - Mux Mx1 inputs for different pattern blocks (related to reference strip signal)									
	block1	**block2**	**block3**	**block4**	**block5**	**block6**	**block7**	**block8**	**block9**	
0	GND	GND	GND	GND	GND	GND	GND	GND	GND	0
1	MS1_OR8 [25]+ MS1_OR4 [24]+ MS1_OR4 [23]+ MS1_OR4 [22]	MS1_OR8 [26]+ MS1_OR4 [25]+ MS1_OR4 [24]+ MS1_OR4 [23]	MS1_OR8 [27]+ MS1_OR4 [26]+ MS1_OR4 [25]+ MS1_OR4 [24]	MS1_OR8 [28]+ MS1_OR4 [27]+ MS1_OR4 [26]+ MS1_OR4 [25]	MS1_OR8 [29]+ MS1_OR4 [28]+ MS1_OR4 [27]+ MS1_OR4 [26]	MS1_OR8 [30]+ MS1_OR4 [29]+ MS1_OR4 [28]+ MS1_OR4 [27]	MS1_OR8 [31]+ MS1_OR4 [30]+ MS1_OR4 [29]+ MS1_OR4 [28]	MS1_OR8 [32]+ MS1_OR4 [31]+ MS1_OR4 [30]+ MS1_OR4 [29]	MS1_OR4 [12]+ MS1_OR4 [11]	1
2	MS1_OR1 [21]+ MS1_OR1 [20]+ MS1_OR1 [19]	MS1_OR1 [22]+ MS1_OR1 [21]+ MS1_OR1 [20]	MS1_OR1 [23]+ MS1_OR1 [22]+ MS1_OR1 [21]	MS1_OR1 [24]+ MS1_OR1 [23]+ MS1_OR1 [22]	MS1_OR1 [25]+ MS1_OR1 [24]+ MS1_OR1 [23]	MS1_OR8 [26]+ MS1_OR4 [25]+ MS1_OR4 [24]+	MS1_OR1 [27]+ MS1_OR1 [26]+ MS1_OR1 [25]	MS1_OR1 [28]+ MS1_OR1 [27]+ MS1_OR1 [26]	MS1_OR4 [10]+ MS1_OR4 [9]	2
3	MS1_OR1 [18]+ MS1_OR1 [17]	MS1_OR1 [19]+ MS1_OR1 [18]	MS1_OR1 [20]+ MS1_OR1 [19]	MS1_OR1 [21]+ MS1_OR1 [20]	MS1_OR1 [22]+ MS1_OR1 [21]	MS1_OR1 [23]+ MS1_OR1 [22]	MS1_OR1 [24]+ MS1_OR1 [23]	MS1_OR4 [25]+ MS1_OR4 [24]	MS1_OR4 [8]	3
4	MS1_OR1 [16]	MS1_OR1 [17]	MS1_OR1 [18]	MS1_OR1 [19]	MS1_OR1 [20]	MS1_OR1 [21]	MS1_OR1 [22]	MS1_OR1 [23]	MS1_OR4 [7]	4
5	MS1_OR1 [15]	MS1_OR1 [16]	MS1_OR1 [17]	MS1_OR1 [18]	MS1_OR1 [19]	MS1_OR1 [20]	MS1_OR1 [21]	MS1_OR1 [22]	MS1_OR1 [28]+ MS1_OR1 [27]+ MS1_OR1 [26]+ MS1_OR1 [25]	5
6	MS1_OR1 [14]	MS1_OR1 [15]	MS1_OR1 [16]	MS1_OR1 [17]	MS1_OR1 [18]	MS1_OR1 [19]	MS1_OR1 [20]	MS1_OR1 [21]	MS1_OR1 [24]+ MS1_OR1 [23]+ MS1_OR1 [22]+ MS1_OR1 [21]	6
7	MS1_OR1 [13]	MS1_OR1 [14]	MS1_OR1 [15]	MS1_OR1 [16]	MS1_OR1 [17]	MS1_OR1 [18]	MS1_OR1 [19]	MS1_OR1 [20]	MS1_OR1 [20]+ MS1_OR1 [19]+ MS1_OR1 [18]+ MS1_OR1 [17]+ MS1_OR1 [16]+ MS1_OR1 [15]+ MS1_OR1 [14]+ MS1_OR1 [13]	7
8	MS1_OR1 [12]	MS1_OR1 [13]	MS1_OR1 [14]	MS1_OR1 [15]	MS1_OR1 [16]	MS1_OR1 [17]	MS1_OR1 [18]	MS1_OR1 [19]	MS1_OR1 [12]+ MS1_OR1 [11]+ MS1_OR1 [10]+ MS1_OR1 [9]	8
9	MS1_OR1 [11]	MS1_OR1 [12]	MS1_OR1 [13]	MS1_OR1 [14]	MS1_OR1 [15]	MS1_OR1 [16]	MS1_OR1 [17]	MS1_OR1 [18]	MS1_OR1 [8]+ MS1_OR1 [7]+ MS1_OR1 [6]+ MS1_OR1 [5]+	9
									MS1_OR4 [6]	10
									MS1_OR4 [5]	11
									MS1_OR4 [4]+ MS1_OR4 [3]	12
									MS1_OR4 [2]+ MS1_OR4 [1]	13

Table 5: Layer 2 (reference) inputs

	Layer 2 (reference) – Reference signals (directly to track circuits)								
	block1	**block2**	**block3**	**block4**	**block5**	**block6**	**block7**	**block8**	**block9**
1	MS2_OR1 [1]	MS2_OR1 [2]	MS2_OR1 [3]	MS2_OR1 [4]	MS2_OR1 [5]	MS2_OR1 [6]	MS2_OR1 [7]	MS2_OR1 [8]	MS2_OR1 [1]+ MS2_OR1 [2]+ MS2_OR1 [3]+ MS2_OR1 [4]+ MS2_OR1 [5]+ MS2_OR1 [6]+ MS2_OR1 [7]+ MS2_OR1 [8]

Table 4: Layer 1 multiplexers positive patterns inputs

Layer 1 positive patterns - Mux Mx1 inputs for different pattern blocks (related to reference strip signal)

	block1	block2	block3	block4	block5	block6	block7	block8
0	GND	GND	GND	GND	GND	GND	GND	GND
1	MS1_OR1 [15]	MS1_OR1 [16]	MS1_OR1 [17]	MS1_OR1 [18]	MS1_OR1 [19]	MS1_OR1 [20]	MS1_OR1 [21]	MS1_OR1 [22]
2	MS1_OR1 [14]	MS1_OR1 [15]	MS1_OR1 [16]	MS1_OR1 [17]	MS1_OR1 [18]	MS1_OR1 [19]	MS1_OR1 [20]	MS1_OR1 [21]
3	MS1_OR1 [13]	MS1_OR1 [14]	MS1_OR1 [15]	MS1_OR1 [16]	MS1_OR1 [17]	MS1_OR1 [18]	MS1_OR1 [19]	MS1_OR1 [20]
4	MS1_OR1 [12]	MS1_OR1 [13]	MS1_OR1 [14]	MS1_OR1 [15]	MS1_OR1 [16]	MS1_OR1 [17]	MS1_OR1 [18]	MS1_OR1 [19]
5	MS1_OR1 [11]	MS1_OR1 [12]	MS1_OR1 [13]	MS1_OR1 [14]	MS1_OR1 [15]	MS1_OR1 [16]	MS1_OR1 [17]	MS1_OR1 [18]
6	MS1_OR1 [10]	MS1_OR1 [11]	MS1_OR1 [12]	MS1_OR1 [13]	MS1_OR1 [14]	MS1_OR1 [15]	MS1_OR1 [16]	MS1_OR1 [17]
7	MS1_OR1 [9]+ MS1_OR1 [8]	MS1_OR1 [10] MS1_OR1 [9]	MS1_OR1 [11]+ MS1_OR1 [10]	MS1_OR1 [12]+ MS1_OR1 [11]	MS1_OR1 [13]+ MS1_OR1 [12]	MS1_OR1 [14]+ MS1_OR1 [13]	MS1_OR1 [15]+ MS1_OR1 [14]	MS1_OR1 [16]+ MS1_OR1 [15]
8	MS1_OR1 [7]+ MS1_OR1 [6]+ MS1_OR1 [5]	MS1_OR1 [8]+ MS1_OR1 [7]+ MS1_OR1 [6]	MS1_OR1 [9]+ MS1_OR2 [8]+ MS1_OR2 [7]	MS1_OR1 [10]+ MS1_OR1 [9]+ MS1_OR1 [8]	MS1_OR1 [11]+ MS1_OR1 [10]+ MS1_OR1 [9]	MS1_OR1 [12]+ MS1_OR1 [11]+ MS1_OR1 [10]	MS1_OR1 [13]+ MS1_OR1 [12]+ MS1_OR1 [11]	MS1_OR1 [14]+ MS1_OR1 [13]+ MS1_OR1 [12]
9	MS1_OR1 [4]+ MS1_OR1 [3]+ MS1_OR1 [2]+ MS1_OR1 [1]	MS1_OR1 [5]+ MS1_OR1 [4]+ MS1_OR1 [3]+ MS1_OR1 [2]	MS1_OR1 [6]+ MS1_OR1 [5]+ MS1_OR1 [4]+ MS1_OR1 [3]	MS1_OR1 [7]+ MS1_OR1 [6]+ MS1_OR1 [5]+ MS1_OR1 [4]	MS1_OR1 [8]+ MS1_OR1 [7]+ MS1_OR1 [6]+ MS1_OR1 [5]	MS1_OR1 [9]+ MS1_OR1 [8]+ MS1_OR1 [7]+ MS1_OR1 [6]	MS1_OR1 [10]+ MS1_OR1 [9]+ MS1_OR1 [8]+ MS1_OR1 [7]	MS1_OR1 [11]+ MS1_OR1 [10]+ MS1_OR1 [9]+ MS1_OR1 [8]

block9	
GND	0
MS1_OR4 [12]+ MS1_OR4 [11]	1
MS1_OR4 [10]+ MS1_OR4 [9]	2
MS1_OR4 [8]	3
MS1_OR4 [7]	4
MS1_OR1 [28]+ MS1_OR1 [27]+ MS1_OR1 [26]+ MS1_OR1 [25]	5
MS1_OR1 [24]+ MS1_OR1 [23]+ MS1_OR1 [22]+ MS1_OR1 [21]	6
MS1_OR1 [20]+ MS1_OR1 [19]+ MS1_OR1 [18]+ MS1_OR1 [17]+ MS1_OR1 [16]+ MS1_OR1 [15]+ MS1_OR1 [14]+ MS1_OR1 [13]	7
MS1_OR1 [12]+ MS1_OR1 [11]+ MS1_OR1 [10]+ MS1_OR1 [9]	8
MS1_OR1 [8]+ MS1_OR1 [7]+ MS1_OR1 [6]+ MS1_OR1 [5]+	9
MS1_OR4 [6]	10
MS1_OR4 [5]	11
MS1_OR4 [4]+ MS1_OR4 [3]	12
MS1_OR4 [2]+ MS1_OR4 [1]	13

Table 6: Layer 3 multiplexer inputs

Layer 3 - Mux Mx3 inputs for different pattern blocks (related to reference strip signal)

	block1	block2	block3	block4	block5	block6	block7	block8
0	GND	GND	GND	GND	GND	GND	GND	GND
1	MS3_OR1 [17]+ MS3_OR1 [16]+ MS3_OR1 [15]	MS3_OR1 [18]+ MS3_OR1 [17]+ MS3_OR1 [16]	MS3_OR1 [19]+ MS3_OR1 [18]+ MS3_OR1 [17]	MS3_OR1 [20]+ MS3_OR1 [19]+ MS3_OR1 [18]	MS3_OR1 [21]+ MS3_OR1 [20]+ MS3_OR1 [19]	MS3_OR1 [22]+ MS3_OR1 [21]+ MS3_OR1 [20]	MS3_OR1 [23]+ MS3_OR1 [22]+ MS3_OR1 [21]	MS3_OR1 [24]+ MS3_OR1 [23]+ MS3_OR1 [22]
2	MS3_OR1 [14]+ MS3_OR1 [13]	MS3_OR1 [15]+ MS3_OR1 [14]	MS3_OR1 [16]+ MS3_OR1 [15]	MS3_OR1 [17]+ MS3_OR1 [16]	MS3_OR1 [18]+ MS3_OR1 [17]	MS3_OR1 [19]+ MS3_OR1 [18]	MS3_OR1 [20]+ MS3_OR1 [19]	MS3_OR1 [21]+ MS3_OR1 [20]
3	MS3_OR1 [12]	MS3_OR1 [13]	MS3_OR1 [14]	MS3_OR1 [15]	MS3_OR1 [16]	MS3_OR1 [17]	MS3_OR1 [18]	MS3_OR1 [19]
4	MS3_OR1 [11]	MS3_OR1 [12]	MS3_OR1 [13]	MS3_OR1 [14]	MS3_OR1 [15]	MS3_OR1 [16]	MS3_OR1 [17]	MS3_OR1 [18]
5	MS3_OR1 [10]	MS3_OR1 [11]	MS3_OR1 [12]	MS3_OR1 [13]	MS3_OR1 [14]	MS3_OR1 [15]	MS3_OR1 [16]	MS3_OR1 [17]
6	MS3_OR1 [9]	MS3_OR1 [10]	MS3_OR1 [11]	MS3_OR1 [12]	MS3_OR1 [13]	MS3_OR1 [14]	MS3_OR1 [15]	MS3_OR1 [16]
7	MS3_OR1 [8]	MS3_OR1 [9]	MS3_OR1 [10]	MS3_OR1 [11]	MS3_OR1 [12]	MS3_OR1 [13]	MS3_OR1 [14]	MS3_OR1 [15]
8	MS3_OR1 [7]	MS3_OR1 [8]	MS3_OR1 [9]	MS3_OR1 [10]	MS3_OR1 [11]	MS3_OR1 [12]	MS3_OR1 [13]	MS3_OR1 [14]
9	MS3_OR1 [6]	MS3_OR1 [7]	MS3_OR1 [8]	MS3_OR1 [9]	MS3_OR1 [10]	MS3_OR1 [11]	MS3_OR1 [12]	MS3_OR1 [13]
10	MS3_OR1 [5]+ MS3_OR1 [4]	MS3_OR1 [6]+ MS3_OR1 [5]	MS3_OR1 [7]+ MS3_OR1 [6]	MS3_OR1 [8]+ MS3_OR1 [7]	MS3_OR1 [9]+ MS3_OR1 [8]	MS3_OR1 [10]+ MS3_OR1 [9]	MS3_OR1 [11]+ MS3_OR1 [10]	MS3_OR1 [12]+ MS3_OR1 [11]
11	MS3_OR1 [3]+ MS3_OR1 [2]+ MS3_OR1 [1]	MS3_OR1 [4]+ MS3_OR1 [3]+ MS3_OR1 [2]	MS3_OR1 [5]+ MS3_OR1 [4]+ MS3_OR1 [3]	MS3_OR1 [6]+ MS3_OR1 [5]+ MS3_OR1 [4]	MS3_OR1 [7]+ MS3_OR1 [6]+ MS3_OR1 [5]	MS3_OR1 [8]+ MS3_OR1 [7]+ MS3_OR1 [6]	MS3_OR1 [9]+ MS3_OR1 [8]+ MS3_OR1 [7]	MS3_OR1 [10]+ MS3_OR1 [9]+ MS3_OR1 [8]

block9	
GND	0
MS2p_OR4 [10]	1
MS2p_OR4 [9]	2
MS2p_OR4 [8]	3
MS2p_OR4 [7]	4
MS2p_OR4 [6]+ MS2p_OR4 [5]	5
MS2p_OR4 [4]	6
MS2p_OR4 [3]	7
MS2p_OR4 [2]	8
MS2p_OR4 [1]	9

Table 7: Layer 4 multiplexer inputs

	Layer 4 - Mux Mx4 inputs for different pattern blocks (related to reference strip signal)								
	block1	**block2**	**block3**	**block4**	**block5**	**block6**	**block7**	**block8**	**block9**
0	GND	GND	GND	GND	GND	GND	GND	GND	GND
1	MS4_OR1 [17]+ MS4_OR1 [16]+ MS4_OR1 [15]	MS4_OR1 [18]+ MS4_OR1 [17]+ MS4_OR1 [16]	MS4_OR1 [19]+ MS4_OR1 [18]+ MS4_OR1 [17]	MS4_OR1 [20]+ MS4_OR1 [19]+ MS4_OR1 [18]	MS4_OR1 [21]+ MS4_OR1 [20]+ MS4_OR1 [19]	MS4_OR1 [22]+ MS4_OR1 [21]+ MS4_OR1 [20]	MS4_OR1 [23]+ MS4_OR1 [22]+ MS4_OR1 [21]	MS4_OR1 [24]+ MS4_OR1 [23]+ MS4_OR1 [22]	MS1p_OR4 [14]+ MS1p_OR4 [13]
2	MS4_OR1 [14]+ MS4_OR1 [13]	MS4_OR1 [15]+ MS4_OR1 [14]	MS4_OR1 [16]+ MS4_OR1 [15]	MS4_OR1 [17]+ MS4_OR1 [16]	MS4_OR1 [18]+ MS4_OR1 [17]	MS4_OR1 [19]+ MS4_OR1 [18]	MS4_OR1 [20]+ MS4_OR1 [19]	MS4_OR1 [21]+ MS4_OR1 [20]	MS1p_OR4 [12]
3	MS4_OR1 [12]	MS4_OR1 [13]	MS4_OR1 [14]	MS4_OR1 [15]	MS4_OR1 [16]	MS4_OR1 [17]	MS4_OR1 [18]	MS4_OR1 [19]	MS1p_OR4 [11]
4	MS4_OR1 [11]	MS4_OR1 [12]	MS4_OR1 [13]	MS4_OR1 [14]	MS4_OR1 [15]	MS4_OR1 [16]	MS4_OR1 [17]	MS4_OR1 [18]	MS1p_OR4 [10]
5	MS4_OR1 [10]	MS4_OR1 [11]	MS4_OR1 [12]	MS4_OR1 [13]	MS4_OR1 [14]	MS4_OR1 [15]	MS4_OR1 [16]	MS4_OR1 [17]	MS1p_OR4 [9]
6	MS4_OR1 [9]	MS4_OR1 [10]	MS4_OR1 [11]	MS4_OR1 [12]	MS4_OR1 [13]	MS4_OR1 [14]	MS4_OR1 [15]	MS4_OR1 [16]	MS1p_OR4 [8]+ MS1p_OR4 [7]
7	MS4_OR1 [8]	MS4_OR1 [9]	MS4_OR1 [10]	MS4_OR1 [11]	MS4_OR1 [12]	MS4_OR1 [13]	MS4_OR1 [14]	MS4_OR1 [15]	MS1p_OR4 [6]
8	MS4_OR1 [7]	MS4_OR1 [8]	MS4_OR1 [9]	MS4_OR1 [10]	MS4_OR1 [11]	MS4_OR1 [12]	MS4_OR1 [13]	MS4_OR1 [14]	MS1p_OR4 [5]
9	MS4_OR1 [6]	MS4_OR1 [7]	MS4_OR1 [8]	MS4_OR1 [9]	MS4_OR1 [10]	MS4_OR1 [11]	MS4_OR1 [12]	MS4_OR1 [13]	MS1p_OR4 [4]
10	MS4_OR1 [5]+ MS4_OR1 [4]	MS4_OR1 [6]+ MS4_OR1 [5]	MS4_OR1 [7]+ MS4_OR1 [6]	MS4_OR1 [8]+ MS4_OR1 [7]	MS4_OR1 [9]+ MS4_OR1 [8]	MS4_OR1 [10]+ MS4_OR1 [9]	MS4_OR1 [11]+ MS4_OR1 [10]	MS4_OR1 [12]+ MS4_OR1 [11]	MS1p_OR4 [3]
11	MS4_OR1 [3]+ MS4_OR1 [2]+ MS4_OR1 [1]	MS4_OR1 [4]+ MS4_OR1 [3]+ MS4_OR1 [2]	MS4_OR1 [5]+ MS4_OR1 [4]+ MS4_OR1 [3]	MS4_OR1 [6]+ MS4_OR1 [5]+ MS4_OR1 [4]	MS4_OR1 [7]+ MS4_OR1 [6]+ MS4_OR1 [5]	MS4_OR1 [8]+ MS4_OR1 [7]+ MS4_OR1 [6]	MS4_OR1 [9]+ MS4_OR1 [8]+ MS4_OR1 [7]	MS4_OR1 [10]+ MS4_OR1 [9]+ MS4_OR1 [8]	MS1p_OR4 [2]+ MS1p_OR4 [1]

Table 8: Groups of positive tracks

Nr	Groups	Tracks	Or's
1	group_of_tracks[1]	track[1] + track[2] + track[3] + track[4] + track[5] + track[6] + track[7] + track[8] + track[9] + track[10] + track[11] + track[12] + track[13] + track[14] + track[15] + track[16] ;	OR16
2	group_of_tracks[2]	track[17] + track[18] + track[19] + track[20] + track[21] + track[22] + track[23] + track[24];	OR8
3	group_of_tracks[3]	track[25] + track[26] + track[27] + track[28] + track[29] + track[30] + track[31] + track[32];	
4	group_of_tracks[4]	track[33] + track[34] + track[35] + track[36];	OR4
5	group_of_tracks[5]	track[37] + track[38] + track[39] + track[40];	
6	group_of_tracks[6]	track[41] + track[42] + track[43] + track[44];	
7	group_of_tracks[7]	track[45] + track[46] + track[47] + track[48];	
8	group_of_tracks[8]	track[49] + track[50];	OR2
9	group_of_tracks[9]	track[51] + track[52];	
10	group_of_tracks[10]	track[53] + track[54];	
11	group_of_tracks[11]	track[55] + track[56];	
12	group_of_tracks[12]	track[57] + track[58];	
13	group_of_tracks[13]	track[59] + track[60];	
14	group_of_tracks[14]	track[61] + track[62];	
15	group_of_tracks[15]	track[63] + track[64];	
16	group_of_tracks[16]	track[65];	OR1
17	group_of_tracks[17]	track[66];	
18	group_of_tracks[18]	track[67];	
19	group_of_tracks[19]	track[68];	
20	group_of_tracks[20]	track[69];	
21	group_of_tracks[21]	track[70];	
22	group_of_tracks[22]	track[71];	
23	group_of_tracks[23]	track[72];	
24	group_of_tracks[24]	track[73];	
25	group_of_tracks[25]	track[74];	
26	group_of_tracks[26]	track[75];	
27	group_of_tracks[27]	track[76];	
28	group_of_tracks[28]	track[77];	
29	group_of_tracks[29]	track[78];	
30	group_of_tracks[30]	track[79];	
31	group_of_tracks[31]	track[80];	

Table 9: Groups of negative tracks

Nr	Groups	Tracks	Or's
32	group_of_tracks[32]	track[81] + track[82] + track[83] + track[84] + track[85] + track[86] + track[87] + track[88] + track[89] + track[90] + track[91] + track[92] + track[93] + track[94] + track[95] + track[96] ;	OR16
33	group_of_tracks[33]	track[97] + track[98] + track[99] + track[100] + track[101] + track[102] + track[103] + track[104];	OR8
34	group_of_tracks[34]	track[105] + track[106] + track[107] + track[108] + track[109] + track[110] + track[111] + track[112];	
35	group_of_tracks[35]	track[113] + track[114] + track[115] + track[116];	OR4
36	group_of_tracks[36]	track[117] + track[118] + track[119] + track[120];	
37	group_of_tracks[37]	track[121] + track[122] + track[123] + track[124];	
38	group_of_tracks[38]	track[125] + track[126] + track[127] + track[128];	
39	group_of_tracks[39]	track[129] + track[130];	OR2
40	group_of_tracks[40]	track[131] + track[132];	
41	group_of_tracks[41]	track[133] + track[134];	
42	group_of_tracks[42]	track[135] + track[136];	
43	group_of_tracks[43]	track[137] + track[138];	
44	group_of_tracks[44]	track[139] + track[140];	
45	group_of_tracks[45]	track[141] + track[142];	
46	group_of_tracks[46]	track[143] + track[144];	
47	group_of_tracks[47]	track[145];	OR1
48	group_of_tracks[48]	track[146];	
49	group_of_tracks[49]	track[147];	
50	group_of_tracks[50]	track[148];	
51	group_of_tracks[51]	track[149];	
52	group_of_tracks[52]	track[150];	
53	group_of_tracks[53]	track[151];	
54	group_of_tracks[54]	track[152];	
55	group_of_tracks[55]	track[153];	
56	group_of_tracks[56]	track[154];	
57	group_of_tracks[57]	track[155];	
58	group_of_tracks[58]	track[156];	
59	group_of_tracks[59]	track[157];	
60	group_of_tracks[60]	track[158];	
61	group_of_tracks[61]	track[159];	
62	group_of_tracks[62]	track[160];	