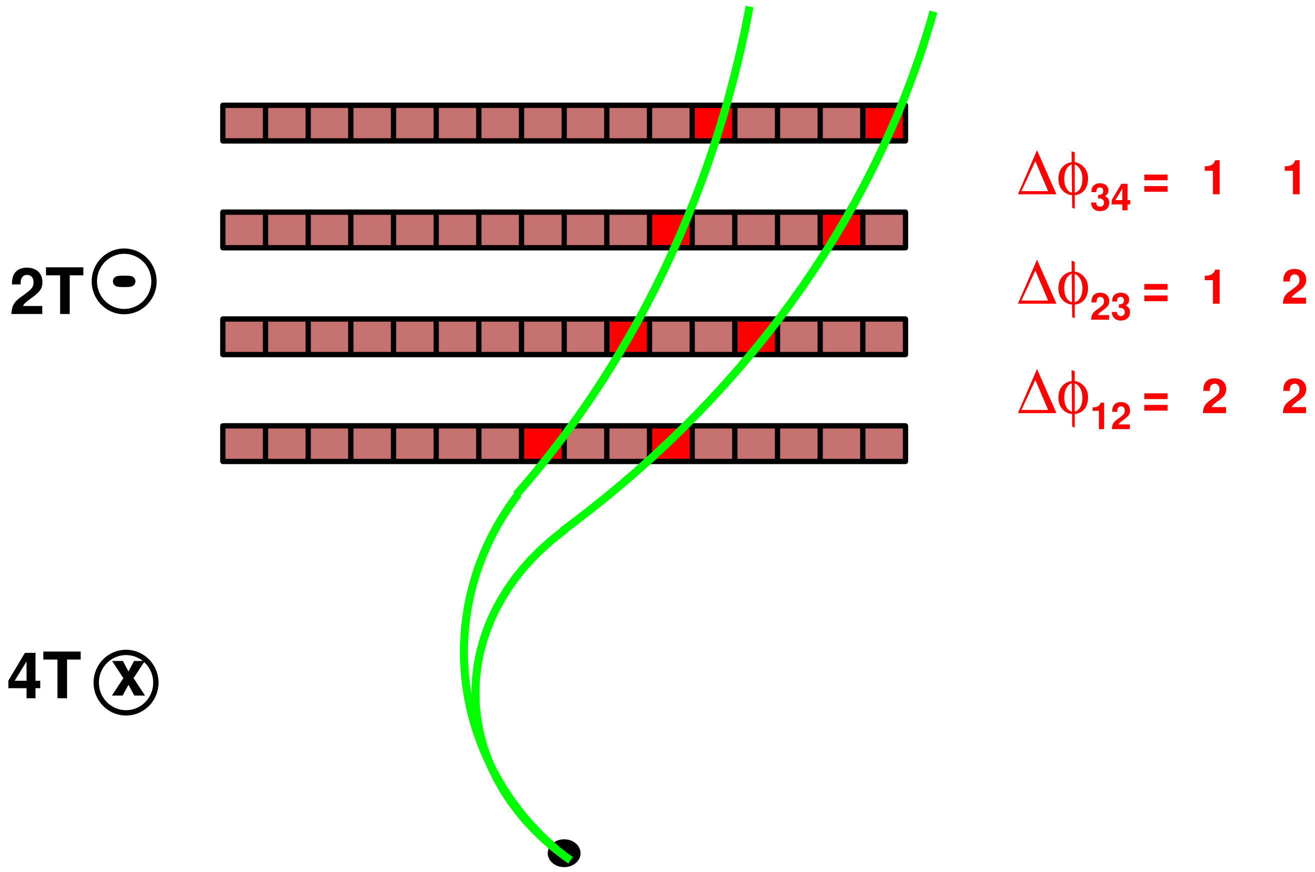
# RPC Muon Trigger Simulation
# MRPC 101




- A muon crossing 4 RPC planes create a pattern of hits.
- The magnetic field bends the muon track in the r$\phi$ plane, thus patterns are different for different $p_t$.
- Thus a $p_t$ cut can be done by defining a set of valid patterns.
- Patterns observed in every event are compared with the valid patterns by Pattern Comparator Trigger (PACT).

# *Simulation of the PACT algorithm*

**In order to prepare a set of valid patterns we simulate single muons with a given $p_t$ in a certain $|\eta|$ interval.**

**The obtained pattern distribution has usually a long tail of rare patterns due to multiple scattering and E−losses.**

**We cut it at 98% of events. In addition a pattern to be selected must appear in at least 2 events.**

**example: patterns for $p_t$=50 GeV, $1.5<|\eta|<1.6$**

| **4–3** | **3–2** | **2–1** | # events | Σ% | **4–3** | **3–2** | **2–1** | # events | Σ% |
|---|---|---|---|---|---|---|---|---|---|
| −1 | 0 | 0 | 4399 | 45.22 | −2 | 0 | 1 | 2 | 99.89 |
| −1 | −1 | 0 | 2527 | 71.20 | −1 | −1 | −1 | 2 | 99.91 |
| −2 | 0 | 0 | 1997 | 91.72 | −3 | 0 | 1 | 1 | 99.92 |
| −1 | 0 | −1 | 677 | 98.68 | 0 | 0 | 1 | 1 | 99.93 |
| −1 | −1 | 1 | 50 | 99.20 | −2 | −8 | −6 | 1 | 99.94 |
| −2 | 0 | −1 | 27 | 99.48 | −2 | 0 | 2 | 1 | 99.95 |
| −2 | −1 | 0 | 13 | 99.61 | −4 | −2 | −2 | 1 | 99.96 |
| 0 | −1 | 0 | 8 | 99.69 | −2 | −1 | −1 | 1 | 99.97 |
| 0 | 0 | 0 | 8 | 99.77 | 1 | 1 | 1 | 1 | 99.98 |
| −1 | 0 | 1 | 5 | 99.83 | −1 | −1 | −8 | 1 | 99.99 |
| 0 | −1 | 1 | 4 | 99.87 | −2 | −1 | 1 | 1 | 100.00 |

**Valid patterns are stored in an external file as a list of possible combination of $\Delta\phi$'s and associated $p_t$ values.**

**The assignment is independent of $\phi$, but it depends on $\eta$ and on the logical strip size IOR=1, 2, 4 or 8.**

# *MTRIG 005 $\Rightarrow$ MRPC 101*

**Previous version of the RPC Trigger software:**

## MTRIG 005

- **ducumented in CMS TN/94–174**
- **last release: December 94**
- **stand alone program compatible with CMSIM 006**

**New version:**

## MRPC 101

- **written from scratch by Marcin Konecki (Warsaw)**
- **ducumented in**
  ***/afs/cern.ch/user/w/wrochna/public/html/mrpc/mrpc.ps***
  ***http://cmsdoc.cern.ch/~wrochna/mrpc/mrpc.ps***
- **first release: December 95**
- **implemented in CMSIM 101**
- **stored in $CMZ/mrpc.cmz**

**Patch //MRPC/RPCDES containes:**

- **$MRPARx: parameter definitions**
- **$MR_xxxx: structure definitions**
- **$MRCxxxx: COMMON statements**

**Patch //MRPC/RPCKIT containes routines for:**

- **definition of control card MRPC**
- **reading in input files**
- **RPC digitisation**
- **RPC trigger simulation**

# *Definition of control card MRPC*

**Flow chart:**

**MU_FFKEY → MU_FFKEY_RPC**

**Card syntax and defaults:**

```
   MRPC       F            0
C         L_RPC_TRIG   MRDEBUG   (common /MRCFLAG/)
C           on/off    debug level
```

**`L_RPC_TRIG` is a switch enabling the RPC trigger simulation. The default is `.FALSE.`**

**`MRDEBUG` controlls the debugging level in coincidence with `DEBU` card of `GEANT`.**

- **available values: –1, 0, 1, 3, 4, 5, 6, 7, 11, 12.**
- **default: 0 – no debug, only warnings and errors reported.**
- **recomended for tutorials: 5 – prints $\eta/\phi$ map of segments** (requires 166 character wide display)

**It is foreseen that in the future there will be another card (called e.g. `TRGP`) controlling the overall trigger processing simulation and the variable `L_RPC_TRIG` will be handled by this card.**

**The debugging level also will be moved to the `DBUG` card with a keyword `MRPC`.**

**The `MRPC` card, however, will still be used to controll various parameters of the `MRPC` algorithm.**

# *Reading in input files*

**Flow chart:**

**M_INITG $\rightarrow$ MU_INITG_RPC**

**Two ASCII files are foreseen:**

- **directory: $CMDB/MUON**
- **default names: mrt1.dat and mrt2.dat**
- **keywords: MRT1 and MRT2**

**mrt1.dat containes the list of valid patterns.**

**mrt2.dat is not yet used in this version. It is forseen for the future, but the entry is already reserved for the internal developement.**

# *RPC digitisation*

**Flow chart:**

## GUDIGI → MRDIGI

**The following geometry information is used:**

- **IDTYPE = IDRPCB, IDRPCF**
- **Sensitive volume names as listed in $MRPAR3:**

| | |
|---|---|
| **RP1W, RP2W, RP3W, RP4W,** | **barrel** |
| **RW11, RW12, RW13, RW21, RW31, RW41** | **forward** |

- **Barrel wheel number IWHEEL is defined according to the following table:**

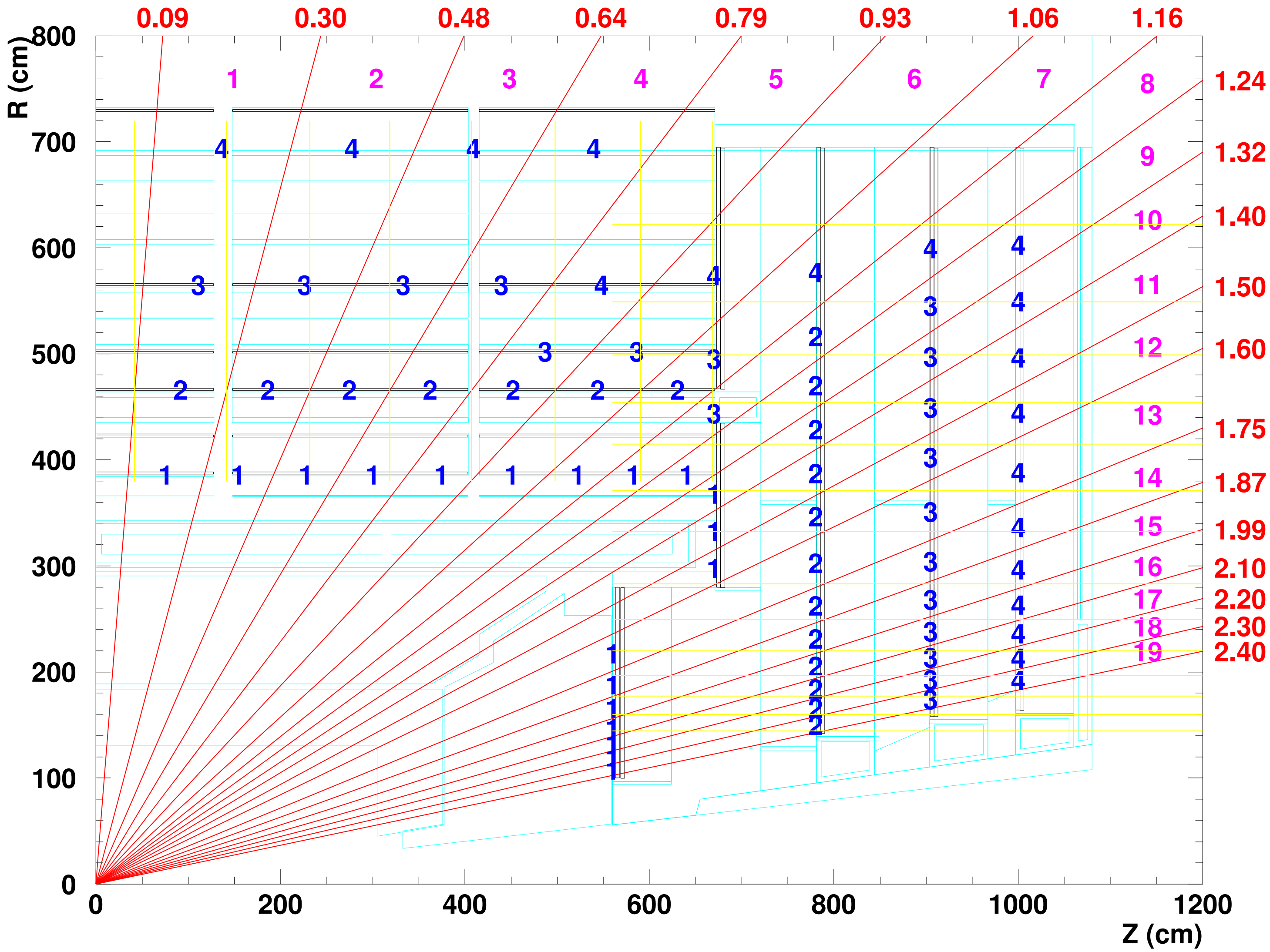
| | | | | | |
|---|---|---|---|---|---|
| **NAMES(NLEVEL−4)** | **MBXX** | **MBXX** | **MCXX** | **MBXX** | **MBXX** |
| **NUMBER(NLEVEL−4)** | **4** | **3** | **1** | **2** | **1** |
| **IWHEEL** | **1** | **2** | **3** | **4** | **5** |

- **There are two identical RPC planes in MB1 (volume RP1W) and two in MB2 (RP2W). They are distinuguished by a copy number = NUMBER (NLEVEL−2).**

**In future we prefer to have different volumes instead of copies, eg. RP5W instead of RP1W/2 and RP6W instead of RP2W/2.**

**It is forseen to store digitised hits in some kind of DIGI banks, but that part of CMSIM is not yet very stable. At the moment the digitisation (MRDIGI) communicates with the trigger simulation (MRTRIG) through the common /MRCDIHI/.**

# *PACT segmentation*




**segment:** $\Delta\phi$**=2.5° (0÷143)** $\times$ $\Delta\eta\approx$**0.1 (–19÷19)**

# *RPC trigger simulation output*

**Output of the algorithm is a list of muons with their $\eta$, $\phi$ and $p_t$ estimated by the trigger logic. It has to be send to the global trigger routine (not existing yet) in a form to be agreed.**

**At the moment it is stored in the common /MRCNMUS/:**

```
PARAMETER (N_MAX_MUONS = 1000)
INTEGER NMUONS      ! number of muons found
RECORD /MR_MUON/ MUONS(N_MAX_MUONS)
COMMON /MRCNMUS/ NMUONS,MUONS
```

**where `MR_MUON` is defined in `$MR_MUON`:**

```
STRUCTURE /MR_MUON/
 INTEGER ITOW       ! η index (–19..19)   (see $MRPAR1)
 INTEGER ISEG       ! φ index (0..143)
 INTEGER IPATNUM    ! pattern number  (for internal use)
 INTEGER IQUALITY   ! quality bit  (0 – 3/4,  1 – 4/4)
 INTEGER NOR        ! OR=1, 2, 4, 8
 INTEGER ILOW_PT    !  high (0) / low (1) pt
 REAL    PT         !  pt assigned
END STRUCTURE
```

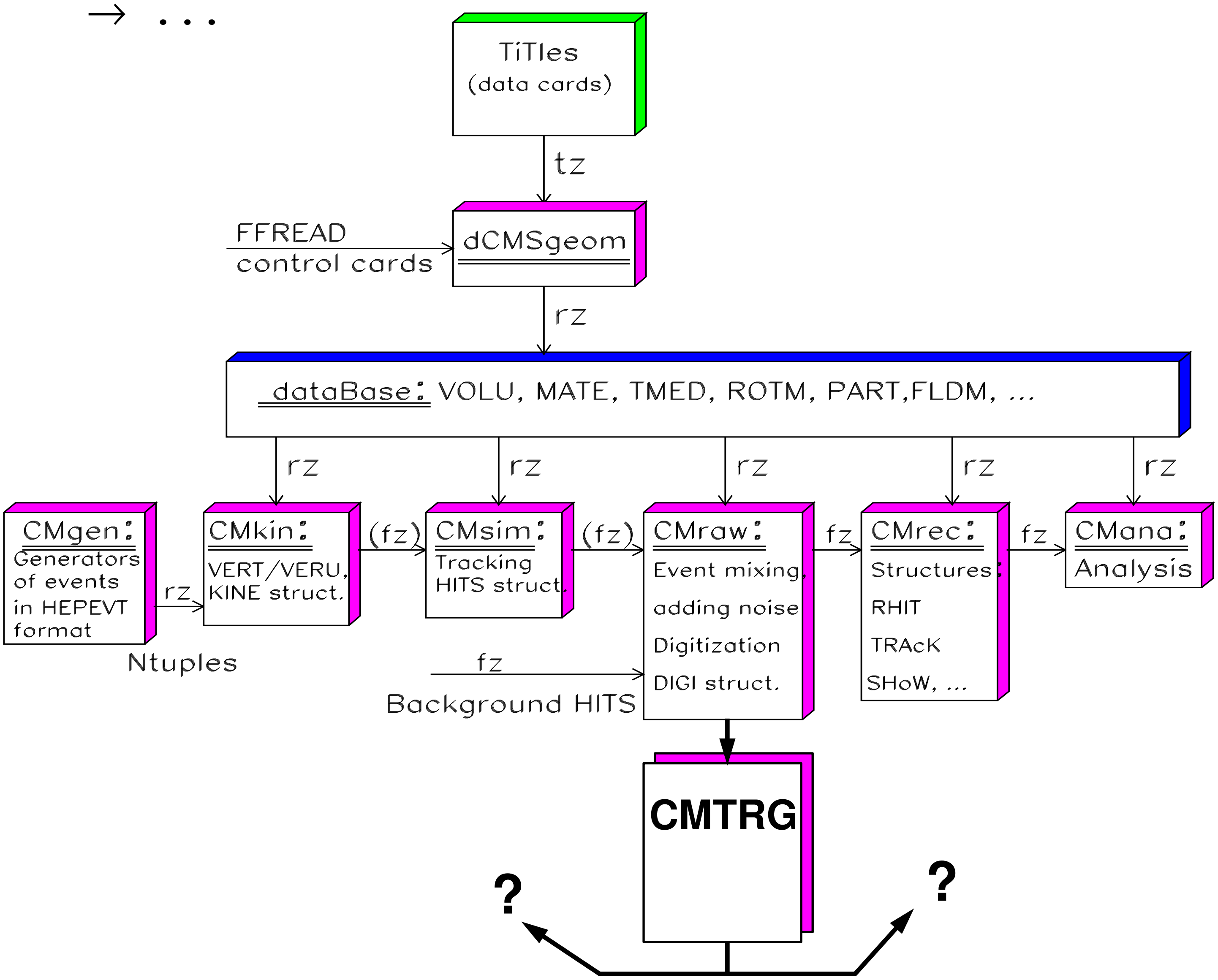
# Example MRPC 101 output

```
** MRSORTMUONS ** == BEFORE ==:MUON<S>(   1): MUON.ITOW=  -2 MUON.ISEG=  32 MUON.IPATNUM=  19 MUON.IQUALITY= 0 MUON.NOR= 1 MUON.ILOW_PT= 0 MUON.PT=  22.39
** MRSORTMUONS ** == BEFORE ==:MUON<S>(   2): MUON.ITOW=  -2 MUON.ISEG=  33 MUON.IPATNUM=  17 MUON.IQUALITY= 0 MUON.NOR= 1 MUON.ILOW_PT= 0 MUON.PT=  24.01
** MRSORTMUONS ** == BEFORE ==:MUON<S>(   3): MUON.ITOW=  -1 MUON.ISEG=  32 MUON.IPATNUM=  18 MUON.IQUALITY= 0 MUON.NOR= 1 MUON.ILOW_PT= 0 MUON.PT=  23.16
** MRSORTMUONS ** == BEFORE ==:MUON<S>(   4): MUON.ITOW=  -1 MUON.ISEG=  33 MUON.IPATNUM=  17 MUON.IQUALITY= 0 MUON.NOR= 1 MUON.ILOW_PT= 0 MUON.PT=  24.97
** MRSORTMUONS ** INDEXES AFTER SORTING:  4,  2,  3,  1,
       0                                                                                                   1
       0         1         2         3         4         5         6         7         8         9         0         1         2         3         4
       012345678901234567890123456789012345678901234567890123456789012345678901234567890123456789012345678901234567890123456789012345678901234567890123
-19 -->................................................................................................................................................<---19
-18 -->................................................................................................................................................<---18
-17 -->................................................................................................................................................<---17
-16 -->................................................................................................................................................<---16
-15 -->................................................................................................................................................<---15
-14 -->................................................................................................................................................<---14
-13 -->................................................................................................................................................<---13
-12 -->................................................................................................................................................<---12
-11 -->................................................................................................................................................<---11
-10 -->................................................................................................................................................<---10
 -9 -->................................................................................................................................................<-- -9
 -8 -->................................................................................................................................................<-- -8
 -7 -->................................................................................................................................................<-- -7
 -6 -->................................................................................................................................................<-- -6
 -5 -->................................................................................................................................................<-- -5
 -4 -->................................................................................................................................................<-- -4
 -3 -->................................................................................................................................................<-- -3
 -2 -->................................--o.............................................................................................................<-- -2
 -1 -->................................-+o.............................................................................................................<-- -1
  0 -->................................ooo.............................................................................................................<--  0
  1 -->................................................................................................................................................<--  1
  2 -->................................................................................................................................................<--  2
  3 -->................................................................................................................................................<--  3
  4 -->................................................................................................................................................<--  4
  5 -->................................................................................................................................................<--  5
  6 -->................................................................................................................................................<--  6
  7 -->................................................................................................................................................<--  7
  8 -->................................................................................................................................................<--  8
  9 -->................................................................................................................................................<--  9
 10 -->................................................................................................................................................<-- 10
 11 -->................................................................................................................................................<-- 11
 12 -->................................................................................................................................................<-- 12
 13 -->................................................................................................................................................<-- 13
 14 -->................................................................................................................................................<-- 14
 15 -->................................................................................................................................................<-- 15
 16 -->................................................................................................................................................<-- 16
 17 -->................................................................................................................................................<-- 17
 18 -->................................................................................................................................................<-- 18
 19 -->................................................................................................................................................<-- 19
       012345678901234567890123456789012345678901234567890123456789012345678901234567890123456789012345678901234567890123456789012345678901234567890123
       0         1         2         3         4         5         6         7         8         9         0         1         2         3         4
       0                                                                                                   1
** MRSORTMUONS ** == SORTED ==:MUON<S>(   1): MUON.ITOW=  -1 MUON.ISEG=  33 MUON.IPATNUM=  17 MUON.IQUALITY= 0 MUON.NOR= 1 MUON.ILOW_PT= 0 MUON.PT=  24.97
```

# *RPC trigger simulation flow*

**GTRIG**

- → ...
- → **GUDIGI** → ...
- → **CMTRG**
  - → **MRTRIG** → **MRMARKTOWERS**
  - → **MRFINDTOWERS**
  - → **MRCHECKCONES**
    - → **MRLOSTPLANE** → **MRCHECKPLANE**
    - → **MRPICKMUON** → **MRCHECKVPAT**
    - → **MRPRINTMUON**
  - → **MRSORTMUONS** → **MRPRINTMUON**
- → **CMRHIT** → ...
- → ...



# *Trigger Processing simulation in CMSIM*