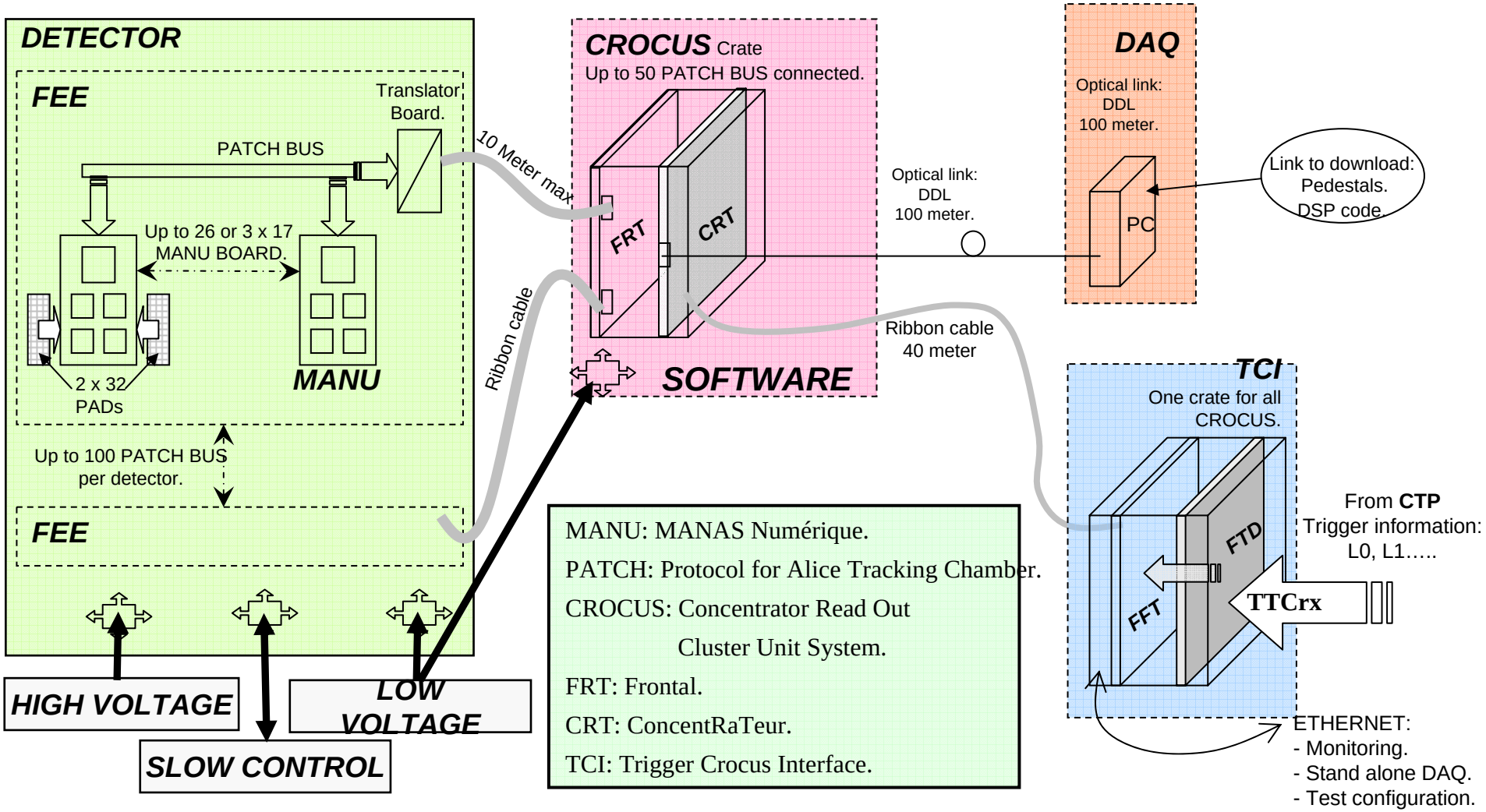
ARCHITECTURE.



MANU: MANAS Numérique.

PATCH: Protocol for Alice Tracking Chamber.

CROCUS: Concentrator Read Out Cluster Unit System.

FRT: Frontal.

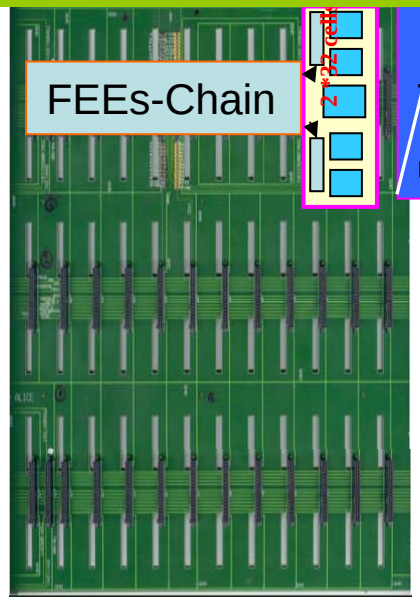
CRT: ConcentRaTeur.

TCI: Trigger Crocus Interface.

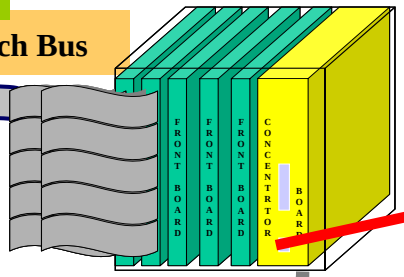
PMD Readout Architecture

Total no of cells : 221184
Total no of Modules : 48
1 module = 4608 cells.

1 CROCUS - 50 Patch Buses
6 CROCUS – 300 Patch Buses

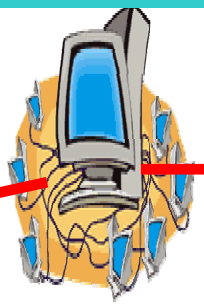


Patch Bus



CROCUS

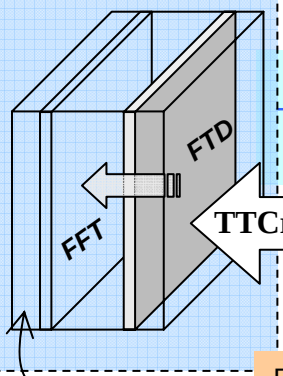
DDL



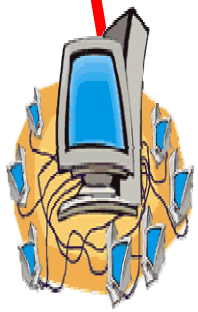
LDC

40 meters LVDS
LINKPORTS TRIGGER
BUSY, and L0

TCI
One crate for all CROCUS.



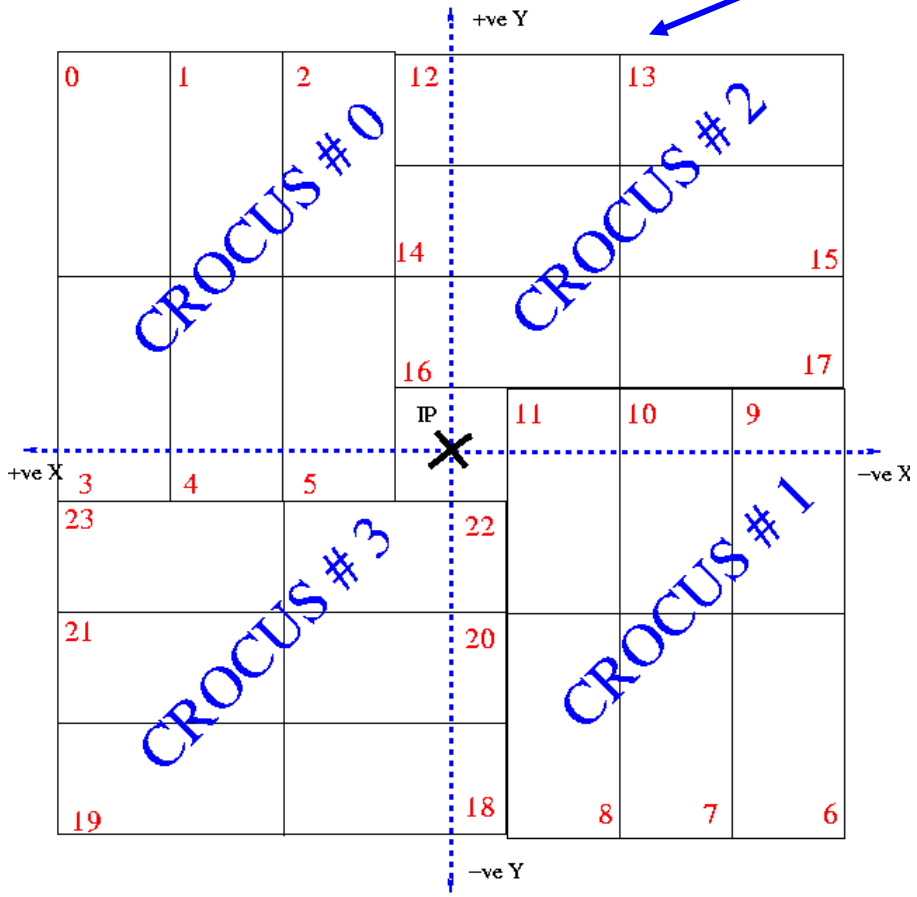
From **CTP**
Trigger information:
L0, L1…..



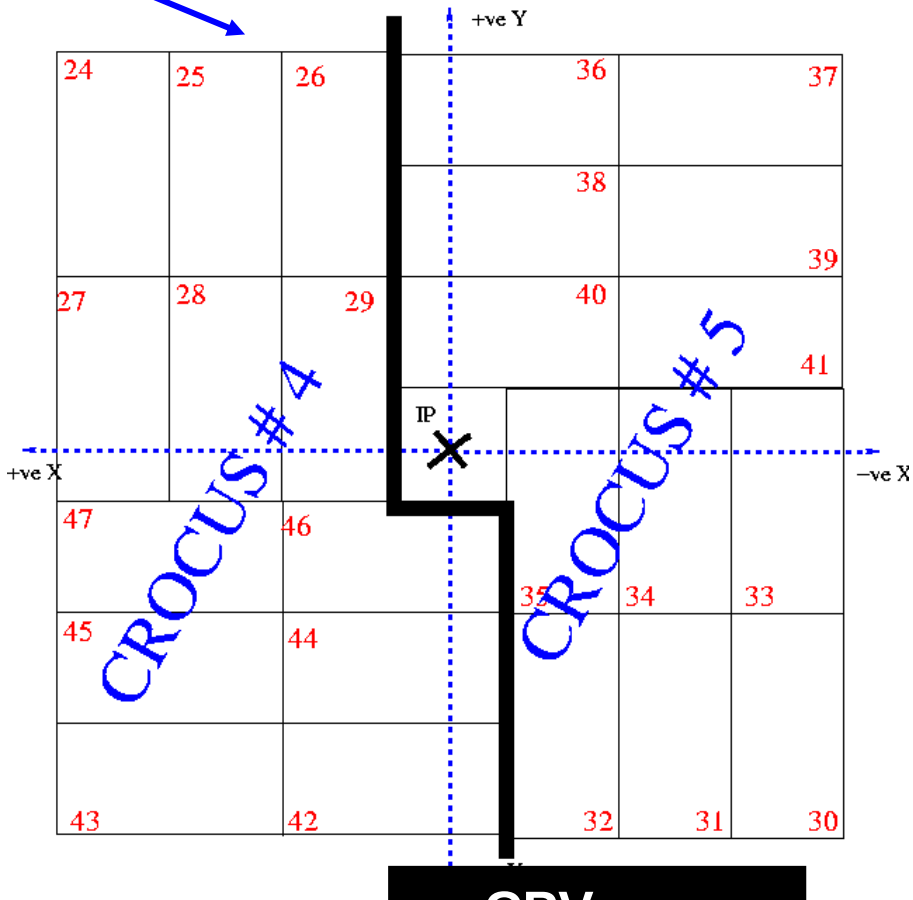
GDC

ETHERNET:
- Monitoring.
- Stand alone DAQ.
- Test configuration.

PMD (Two planes)



PRE–SHOWER PLANE – NUMBERS IN RED ARE MODULE NUMBERS



PRE–VETO PLANE – NUME

Preshower
- 24 modules
- 12 long type
- 12 short type

- 6 CROCUS
- 6 DDL

(4 PRESHOWER, 2 CPV)

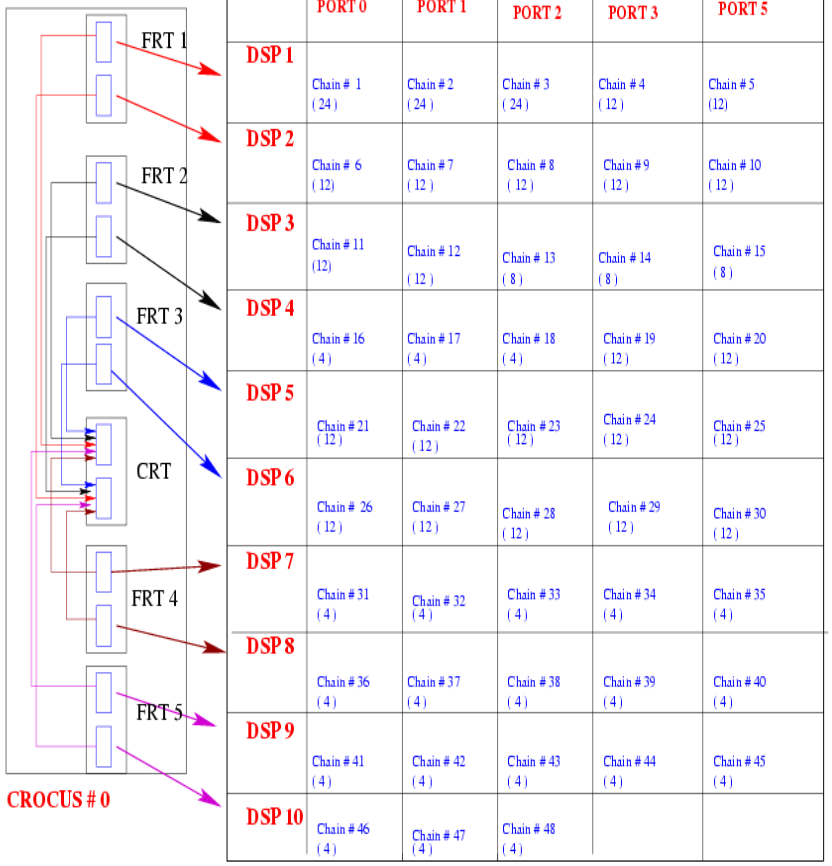
CPV
- 24 modules
- 12 long type
- 12 short type

CHAIN CONFIGURATION

CROCUS#0
➢ **48 chains**

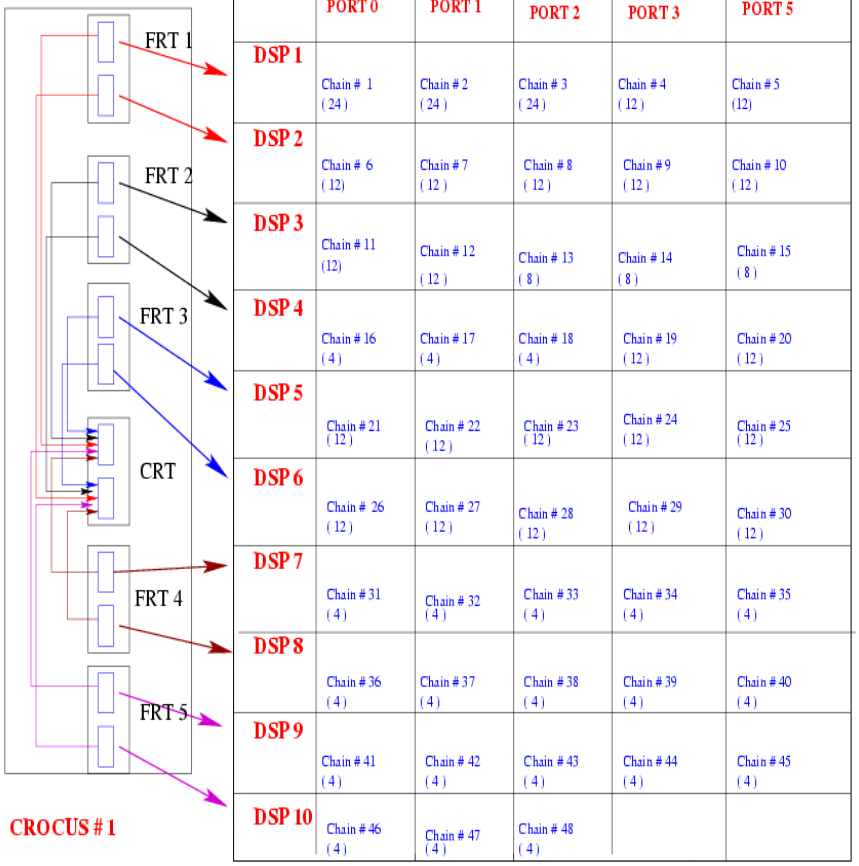
CROCUS#1
➢ **48 chains**

The No. of FEEs has been decided considering the occupancy in that zone



	LINK PORT 0	LINK PORT 1	LINK PORT 2	LINK PORT 3	LINK PORT 5
DSP 1	Chain # 1 (24)	Chain # 2 (24)	Chain # 3 (24)	Chain # 4 (12)	Chain # 5 (12)
DSP 2	Chain # 6 (12)	Chain # 7 (12)	Chain # 8 (12)	Chain # 9 (12)	Chain # 10 (12)
DSP 3	Chain # 11 (12)	Chain # 12 (12)	Chain # 13 (8)	Chain # 14 (8)	Chain # 15 (8)
DSP 4	Chain # 16 (4)	Chain # 17 (4)	Chain # 18 (4)	Chain # 19 (12)	Chain # 20 (12)
DSP 5	Chain # 21 (12)	Chain # 22 (12)	Chain # 23 (12)	Chain # 24 (12)	Chain # 25 (12)
DSP 6	Chain # 26 (12)	Chain # 27 (12)	Chain # 28 (12)	Chain # 29 (12)	Chain # 30 (12)
DSP 7	Chain # 31 (4)	Chain # 32 (4)	Chain # 33 (4)	Chain # 34 (4)	Chain # 35 (4)
DSP 8	Chain # 36 (4)	Chain # 37 (4)	Chain # 38 (4)	Chain # 39 (4)	Chain # 40 (4)
DSP 9	Chain # 41 (4)	Chain # 42 (4)	Chain # 43 (4)	Chain # 44 (4)	Chain # 45 (4)
DSP 10	Chain # 46 (4)	Chain # 47 (4)	Chain # 48 (4)		

PRESHOWER

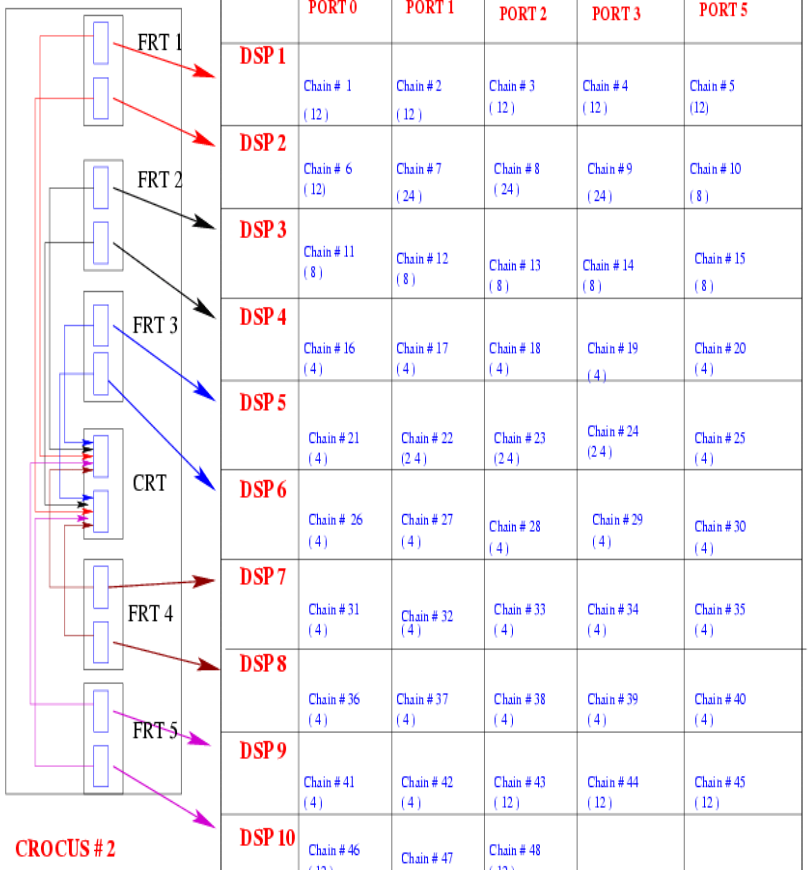


	LINK PORT 0	LINK PORT 1	LINK PORT 2	LINK PORT 3	LINK PORT 5
DSP 1	Chain # 1 (24)	Chain # 2 (24)	Chain # 3 (24)	Chain # 4 (12)	Chain # 5 (12)
DSP 2	Chain # 6 (12)	Chain # 7 (12)	Chain # 8 (12)	Chain # 9 (12)	Chain # 10 (12)
DSP 3	Chain # 11 (12)	Chain # 12 (12)	Chain # 13 (8)	Chain # 14 (8)	Chain # 15 (8)
DSP 4	Chain # 16 (4)	Chain # 17 (4)	Chain # 18 (4)	Chain # 19 (12)	Chain # 20 (12)
DSP 5	Chain # 21 (12)	Chain # 22 (12)	Chain # 23 (12)	Chain # 24 (12)	Chain # 25 (12)
DSP 6	Chain # 26 (12)	Chain # 27 (12)	Chain # 28 (12)	Chain # 29 (12)	Chain # 30 (12)
DSP 7	Chain # 31 (4)	Chain # 32 (4)	Chain # 33 (4)	Chain # 34 (4)	Chain # 35 (4)
DSP 8	Chain # 36 (4)	Chain # 37 (4)	Chain # 38 (4)	Chain # 39 (4)	Chain # 40 (4)
DSP 9	Chain # 41 (4)	Chain # 42 (4)	Chain # 43 (4)	Chain # 44 (4)	Chain # 45 (4)
DSP 10	Chain # 46 (4)	Chain # 47 (4)	Chain # 48 (4)		

CHAIN CONFIGURATION

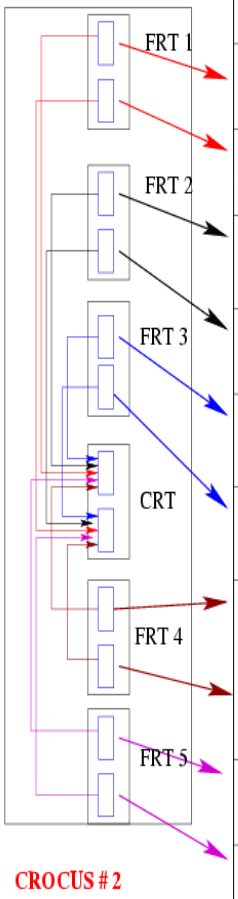
CROCUS#2
- **48 chains**

CROCUS#3
- **48 chains**



	LINK PORT 0	LINK PORT 1	LINK PORT 2	LINK PORT 3	LINK PORT 5
DSP 1	Chain # 1 (12)	Chain # 2 (12)	Chain # 3 (12)	Chain # 4 (12)	Chain # 5 (12)
DSP 2	Chain # 6 (12)	Chain # 7 (24)	Chain # 8 (24)	Chain # 9 (24)	Chain # 10 (8)
DSP 3	Chain # 11 (8)	Chain # 12 (8)	Chain # 13 (8)	Chain # 14 (8)	Chain # 15 (8)
DSP 4	Chain # 16 (4)	Chain # 17 (4)	Chain # 18 (4)	Chain # 19 (4)	Chain # 20 (4)
DSP 5	Chain # 21 (4)	Chain # 22 (2 4)	Chain # 23 (2 4)	Chain # 24 (2 4)	Chain # 25 (4)
DSP 6	Chain # 26 (4)	Chain # 27 (4)	Chain # 28 (4)	Chain # 29 (4)	Chain # 30 (4)
DSP 7	Chain # 31 (4)	Chain # 32 (4)	Chain # 33 (4)	Chain # 34 (4)	Chain # 35 (4)
DSP 8	Chain # 36 (4)	Chain # 37 (4)	Chain # 38 (4)	Chain # 39 (4)	Chain # 40 (4)
DSP 9	Chain # 41 (4)	Chain # 42 (4)	Chain # 43 (12)	Chain # 44 (12)	Chain # 45 (12)
DSP 10	Chain # 46 (12)	Chain # 47 (12)	Chain # 48 (12)		

PRESHOWER

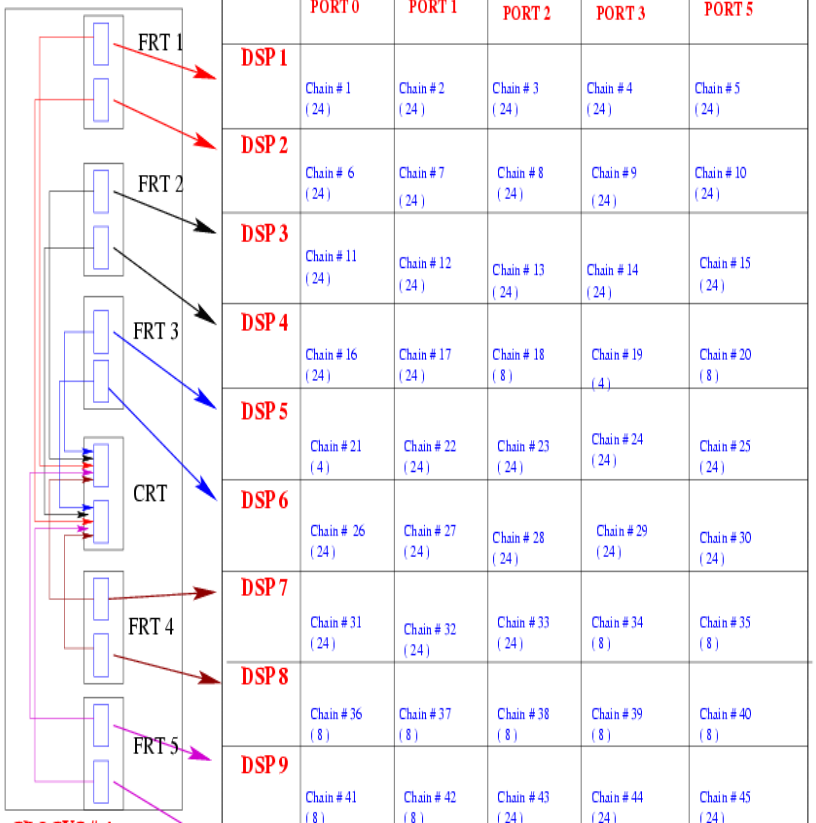


	LINK PORT 0	LINK PORT 1	LINK PORT 2	LINK PORT 3	LINK PORT 5
DSP 1	Chain # 1 (12)	Chain # 2 (12)	Chain # 3 (12)	Chain # 4 (12)	Chain # 5 (12)
DSP 2	Chain # 6 (12)	Chain # 7 (24)	Chain # 8 (24)	Chain # 9 (24)	Chain # 10 (8)
DSP 3	Chain # 11 (8)	Chain # 12 (8)	Chain # 13 (8)	Chain # 14 (8)	Chain # 15 (8)
DSP 4	Chain # 16 (4)	Chain # 17 (4)	Chain # 18 (4)	Chain # 19 (4)	Chain # 20 (4)
DSP 5	Chain # 21 (4)	Chain # 22 (2 4)	Chain # 23 (2 4)	Chain # 24 (2 4)	Chain # 25 (4)
DSP 6	Chain # 26 (4)	Chain # 27 (4)	Chain # 28 (4)	Chain # 29 (4)	Chain # 30 (4)
DSP 7	Chain # 31 (4)	Chain # 32 (4)	Chain # 33 (4)	Chain # 34 (4)	Chain # 35 (4)
DSP 8	Chain # 36 (4)	Chain # 37 (4)	Chain # 38 (4)	Chain # 39 (4)	Chain # 40 (4)
DSP 9	Chain # 41 (4)	Chain # 42 (4)	Chain # 43 (12)	Chain # 44 (12)	Chain # 45 (12)
DSP 10	Chain # 46 (12)	Chain # 47 (12)	Chain # 48 (12)		

CHAIN CONFIGURATION

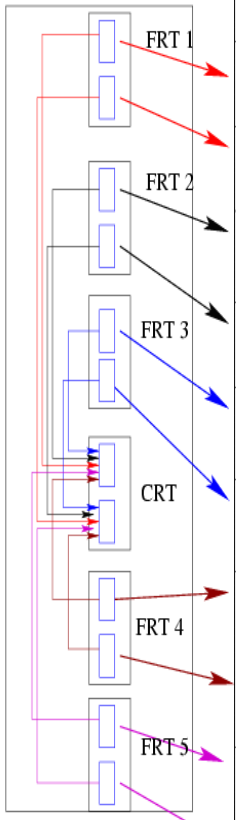
CROCUS#4
- **45 chains**

CROCUS#5
- **45 chains**



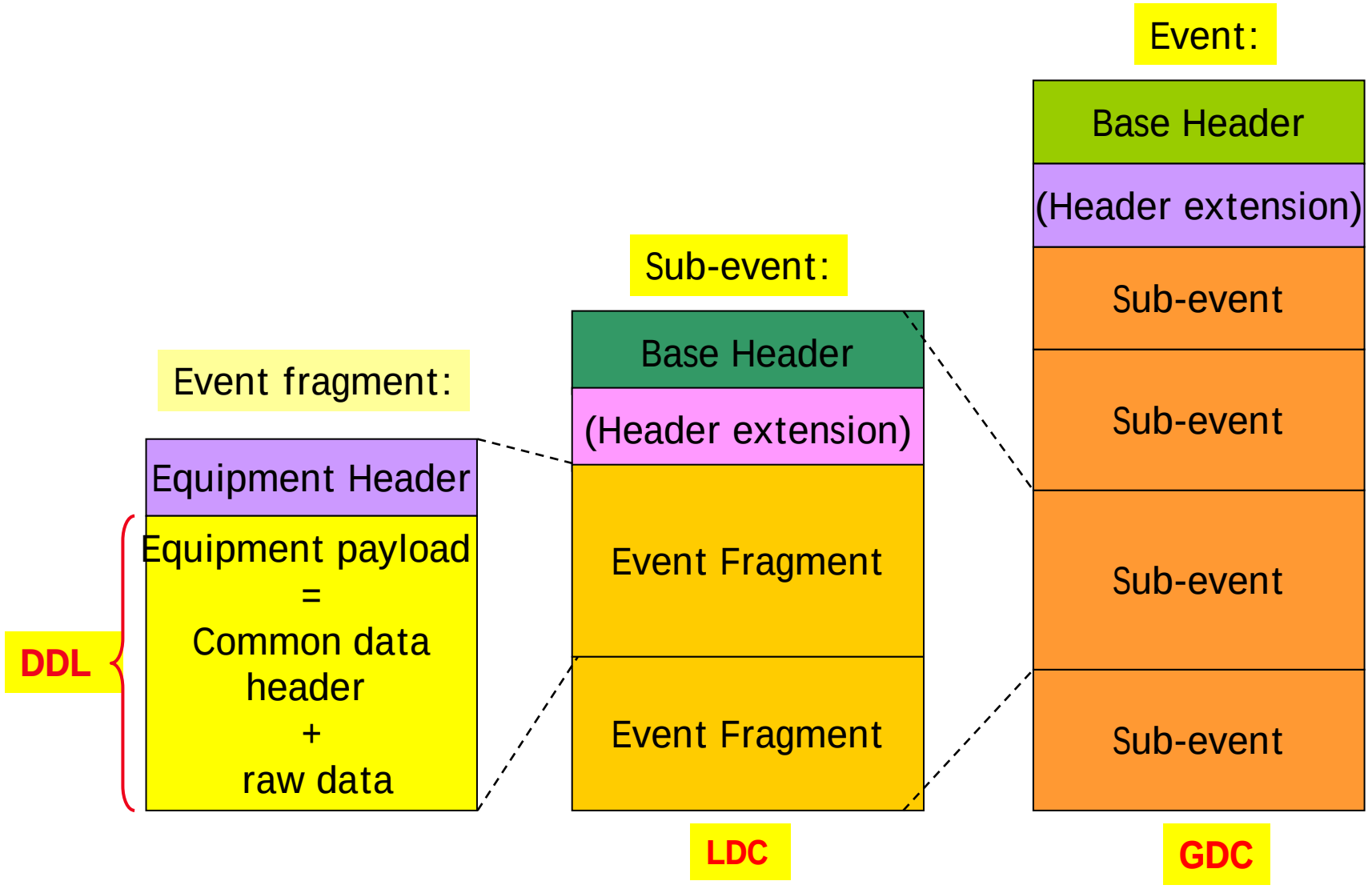
	LINK PORT 0	LINK PORT 1	LINK PORT 2	LINK PORT 3	LINK PORT 5
DSP 1	Chain # 1 (24)	Chain # 2 (24)	Chain # 3 (24)	Chain # 4 (24)	Chain # 5 (24)
DSP 2	Chain # 6 (24)	Chain # 7 (24)	Chain # 8 (24)	Chain # 9 (24)	Chain # 10 (24)
DSP 3	Chain # 11 (24)	Chain # 12 (24)	Chain # 13 (24)	Chain # 14 (24)	Chain # 15 (24)
DSP 4	Chain # 16 (24)	Chain # 17 (24)	Chain # 18 (8)	Chain # 19 (4)	Chain # 20 (8)
DSP 5	Chain # 21 (4)	Chain # 22 (24)	Chain # 23 (24)	Chain # 24 (24)	Chain # 25 (24)
DSP 6	Chain # 26 (24)	Chain # 27 (24)	Chain # 28 (24)	Chain # 29 (24)	Chain # 30 (24)
DSP 7	Chain # 31 (24)	Chain # 32 (24)	Chain # 33 (24)	Chain # 34 (8)	Chain # 35 (8)
DSP 8	Chain # 36 (8)	Chain # 37 (8)	Chain # 38 (8)	Chain # 39 (8)	Chain # 40 (8)
DSP 9	Chain # 41 (8)	Chain # 42 (8)	Chain # 43 (24)	Chain # 44 (24)	Chain # 45 (24)
DSP 10					

CPV

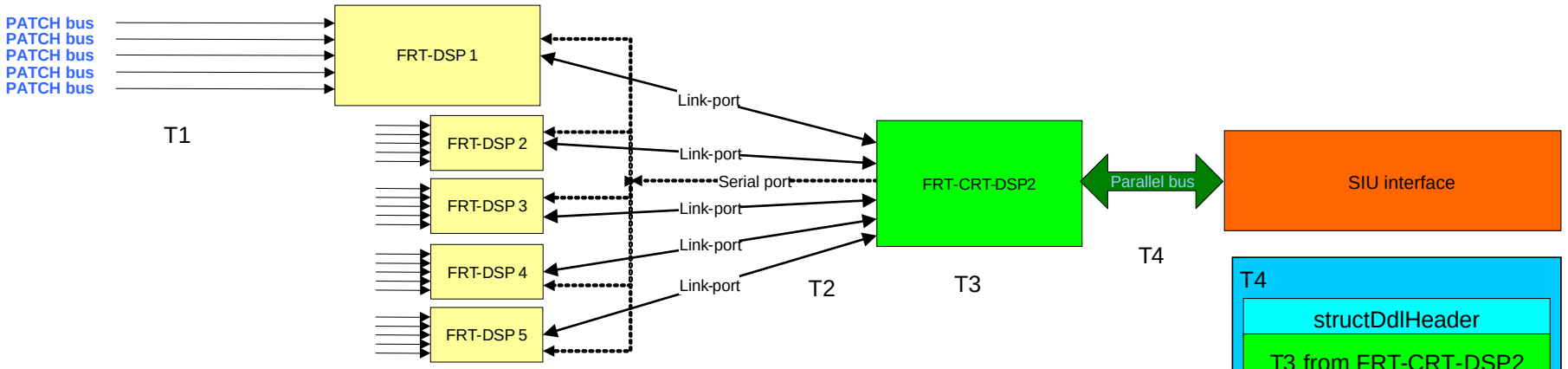


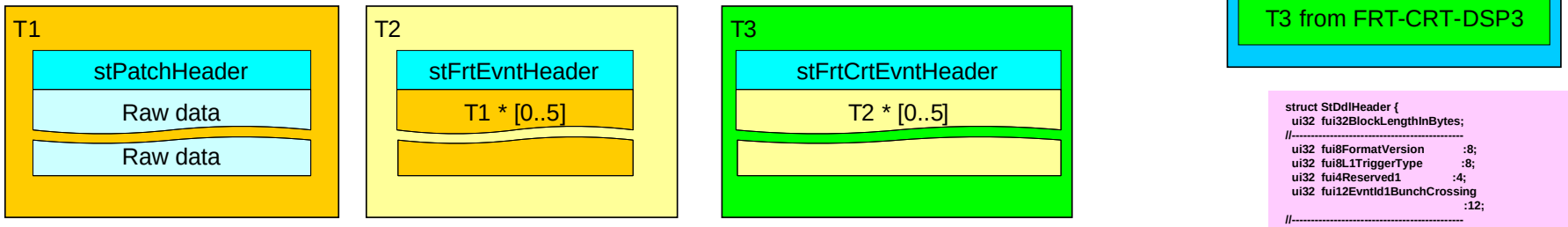
	LINK PORT 0	LINK PORT 1	LINK PORT 2	LINK PORT 3	LINK PORT 5
DSP 1	Chain # 1 (24)	Chain # 2 (24)	Chain # 3 (24)	Chain # 4 (24)	Chain # 5 (24)
DSP 2	Chain # 6 (24)	Chain # 7 (24)	Chain # 8 (24)	Chain # 9 (24)	Chain # 10 (24)
DSP 3	Chain # 11 (24)	Chain # 12 (24)	Chain # 13 (24)	Chain # 14 (24)	Chain # 15 (24)
DSP 4	Chain # 16 (24)	Chain # 17 (24)	Chain # 18 (8)	Chain # 19 (4)	Chain # 20 (8)
DSP 5	Chain # 21 (4)	Chain # 22 (24)	Chain # 23 (24)	Chain # 24 (24)	Chain # 25 (24)
DSP 6	Chain # 26 (24)	Chain # 27 (24)	Chain # 28 (24)	Chain # 29 (24)	Chain # 30 (24)
DSP 7	Chain # 31 (24)	Chain # 32 (24)	Chain # 33 (24)	Chain # 34 (8)	Chain # 35 (8)
DSP 8	Chain # 36 (8)	Chain # 37 (8)	Chain # 38 (8)	Chain # 39 (8)	Chain # 40 (8)
DSP 9	Chain # 41 (8)	Chain # 42 (8)	Chain # 43 (24)	Chain # 44 (24)	Chain # 45 (24)
DSP 10					

Event Logic Structure



DDL Data Format






```
struct StPatchHeader {
  ui32  fuiPatchDataKey;//0xB000000B
  ui32  fuiTotalLength;
  ui32  fuiRawDataLength;
  ui32  fuiPATCHID;
}
```

```
struct StFrtEventHeader {
  ui32  fuiFrtDataKey;   // 0xF000000F
  ui32  fuiTotalLength;
  ui32  fuiRawDataLength;
  ui32  fuiDspID;
  ui32  fuiCrt1L1aTriggerCounter;
  ui32  fuiMiniEvntIdBunchCrossing;
  ui32  fuiFrtL1aTriggerCounter;
  ui32  fuiFrtL1rTriggerCounter;
  bool  fbEvenPadded;
  ui32  fuiErr;
}
```

Note:

:

If (bEvenPadded=true T3 block will terminate by junk32 bit word

```
struct StFrtCrtEventHeader {
  ui32  fuiFrtCrtDataKey;          //0xFC0000FC Filled by FrtCrtEventDumper ( Frt-Crt 2&3 )
  ui32  fuiTotalLength;            // Filled by FrtCrtEventDumper ( Frt-Crt 2&3 )
  ui32  fuiRawDataLength;          // Filled by FrtCrtEventDumper ( Frt-Crt 2&3 )
  ui32  fuiFrtCrtDspID;            // Filled by FrtCrtEventDumper ( Frt-Crt 2&3 )
  ui32  fuiCrt1L0TriggerCounter;   // Filled by DdlDumpingManager (Crt1)
  ui32  fuiMiniEvntIdBunchCrossing;  // Filled by DdlDumpingManager (Crt1)
  ui32  fuiEvntId1BunchCrossing;   // Filled by DdlDumpingManager (Crt1)
  ui32  fuiEvntId2OrbitNumber;     // Filled by DdlDumpingManager (Crt1)
//  ui32  fuiEvenPadding;          // Filled by FrtCrtEventDumper ( Frt-Crt 2&3 )
};
```

Note:

m_uiEvenPadding is just used to make structFrtCrtEvntHeader header size even m_uiEvenPadding=0xDEADBEEF

DDL TRAILERS 2 Words :
0xD000000D

```
struct StDdlHeader {
  ui32  fui32BlockLengthInBytes;
//-------------------------------------------
  ui32  fui8FormatVersion          :8;
  ui32  fui8L1TriggerType          :8;
  ui32  fui4Reserved1              :4;
  ui32  fui12EvntId1BunchCrossing
                                   :12;
//-------------------------------------------
  ui32  fui8Reserved2
                                   :8;
  ui32  fui24EvntId2OrbitNumber
                                   :24;
//-------------------------------------------
  ui32  fui8BlockAttributes        :8;
  ui32  fui24ParticipSubDetectors
                                   :24;
//-------------------------------------------
  ui32  fui4Reserved3
                                   :4;
  ui32  fui16StatusErrBits         :16;
  ui32  fui12MiniEvntIdBunchCrossing :12;
//-------------------------------------------
  ui32  fui32TriggerClassesLow;
//-------------------------------------------
  ui32  fui4RoiLow
                                   :4;
  ui32  fui10Reserved4
                                   :10;
  ui32  fui18TriggerClassesHigh    :18;
//-------------------------------------------
  ui32  fui12RoiHigh
                                   ;
}
```

DATA WORD (FEE)

```
typedef struct
  {
  unsigned adc  :12;
  unsigned channel :6;
  unsigned fee   :11;
  unsigned unused :2;
  unsigned parity :1;
  }PmdRawDataWord;
```

MANAS Adr. 0

MANAS Adr. 1

MANAS Adr. 2

MANAS Adr. 3

0	0
0	1
1	0
1	1

P = Parity bit (1 bit)

M = FEE board address (11 bits)

G = MANAS address (2 bits)

C = Channel number (4 bits)

D = Data (12 bits)

32	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1
P	0	0	M	M	M	M	M	M	M	M	M	M	M	G	G	C	C	C	C	D	D	D	D	D	D	D	D	D	D	D	D

Output Word

DDL / Equipment ID scheme

Detector Id	Detector Name	DDL Offset	# of DDL	DDL Index	Connected to	Equipment Id
9	PMD	2304	6	0	CROCUS#0	2304
				1	CROCUS#1	2305
				2	CROCUS#2	2306
				3	CROCUS#3	2307
				4	CROCUS#4	2308
				5	CROCUS#5	2309

Hardware mapping

Idea is to convert

DDL No., patch Bus No., FEE No., Channel No. to the (x,y) coordinate

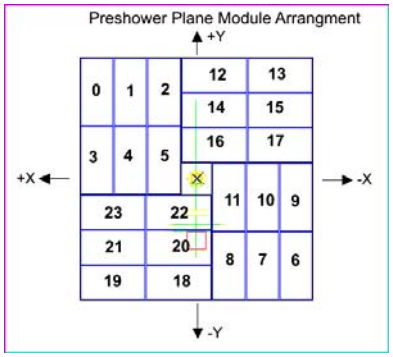
Done in 2 steps

1. DDL No., PATCH Bus No., FEE address give the physical location Of the FEE Board

2. FEE Board, Channel No give the (x,y) coordinate of the cell

Hardware Mapping : Step 1.

Module # 0



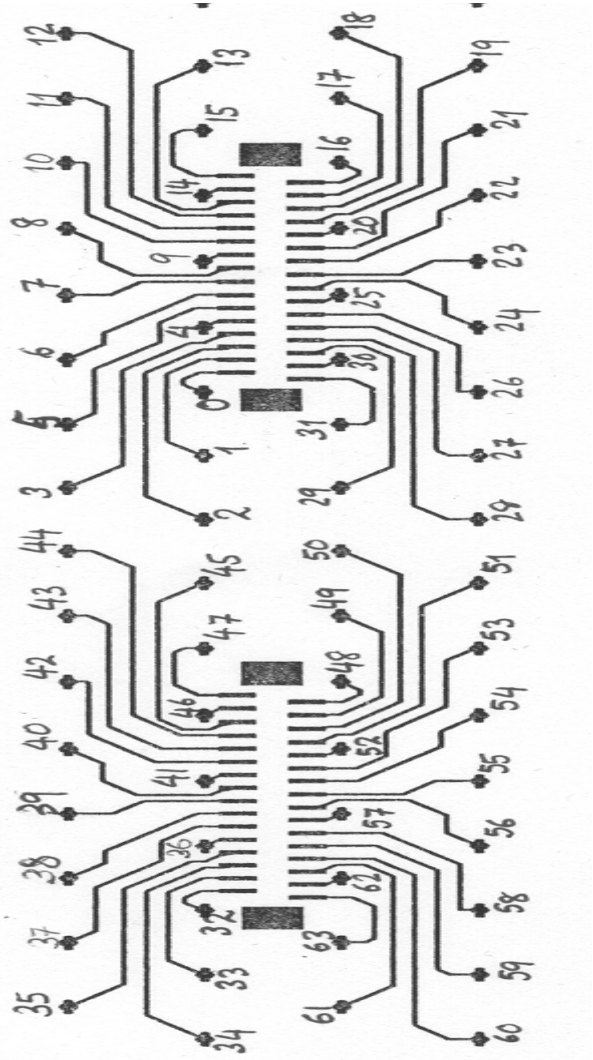
- There are six mapping file for six DDLs.
- PMD_Mapping_ddl*.dat contains this mapping.
- Once the FEE Board physical loacation within a module is identified, Step-2 is applied

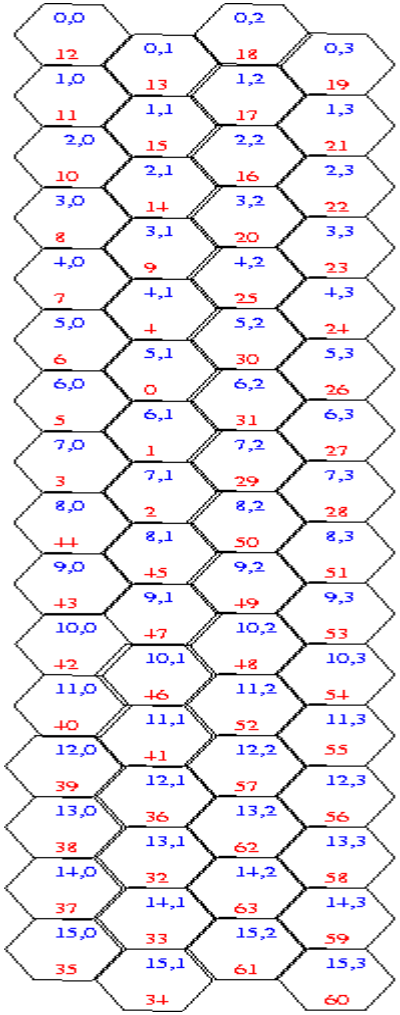
	0	1	2	3	4	5	6	7	8	9	10	11
BB	12	13	14	15	16	17	18	19	20	21	22	23
TB 0	0	1	2	3	4	5	6	7	8	9	10	11
BB	12	13	14	15	16	17	18	19	20	21	22	23
TB 1	0	1	2	3	4	5	6	7	8	9	10	11
BB	12	13	14	15	16	17	18	19	20	21	22	23
TB 2	0	1	2	3	4	5	6	7	8	9	10	11

0 8 15 23 31 39 47

0 15 31 47 63 79 95

Channel Number to (x,y) : Step 2.





This conversion is done
Inside the program

Status :

Hardware Mapping : done (Committed to CVS)

Raw data reconstruction : in place (CVS), to be tested in this test beam

Raw data format : in place (CVS), to be tested in this test beam

Removal of gAlice : by 31st October, 2006

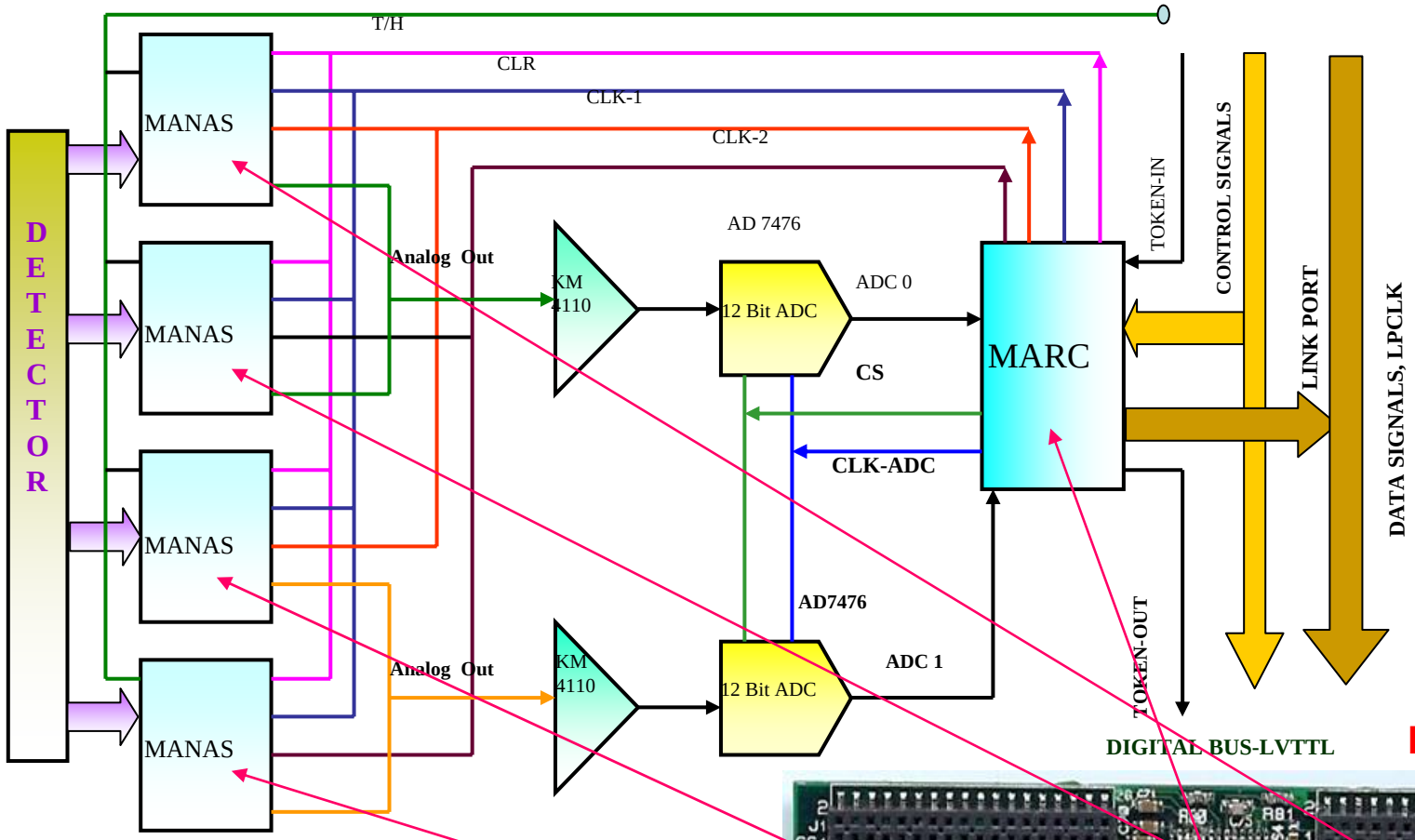
Raw2(s)Digits : by 31st October, 2006

DDL / Equipment ID scheme

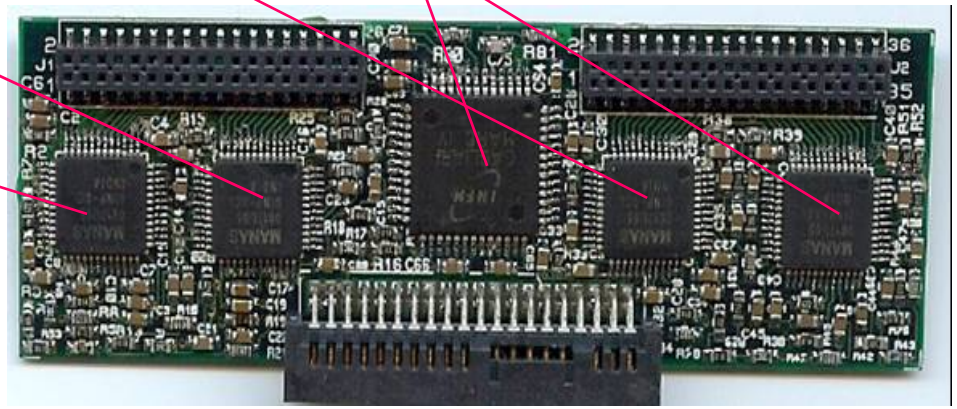
Detector ID	Detector Name	DDL Offset	#Of DDL	DDL Index	Equipment – ID	
8	CPV	2048	2	4	2052	Connected to CROCUS#4
				5	2053	Connected to CROCUS#5
9	PRE–SHOWER	2304	4	0	2304	Connected to CROCUS#0
				1	2305	Connected to CROCUS#1
				2	2306	Connected to CROCUS#2
				3	2307	Connected to CROCUS#3

FEE BOARD CONFIGURATION

CAL



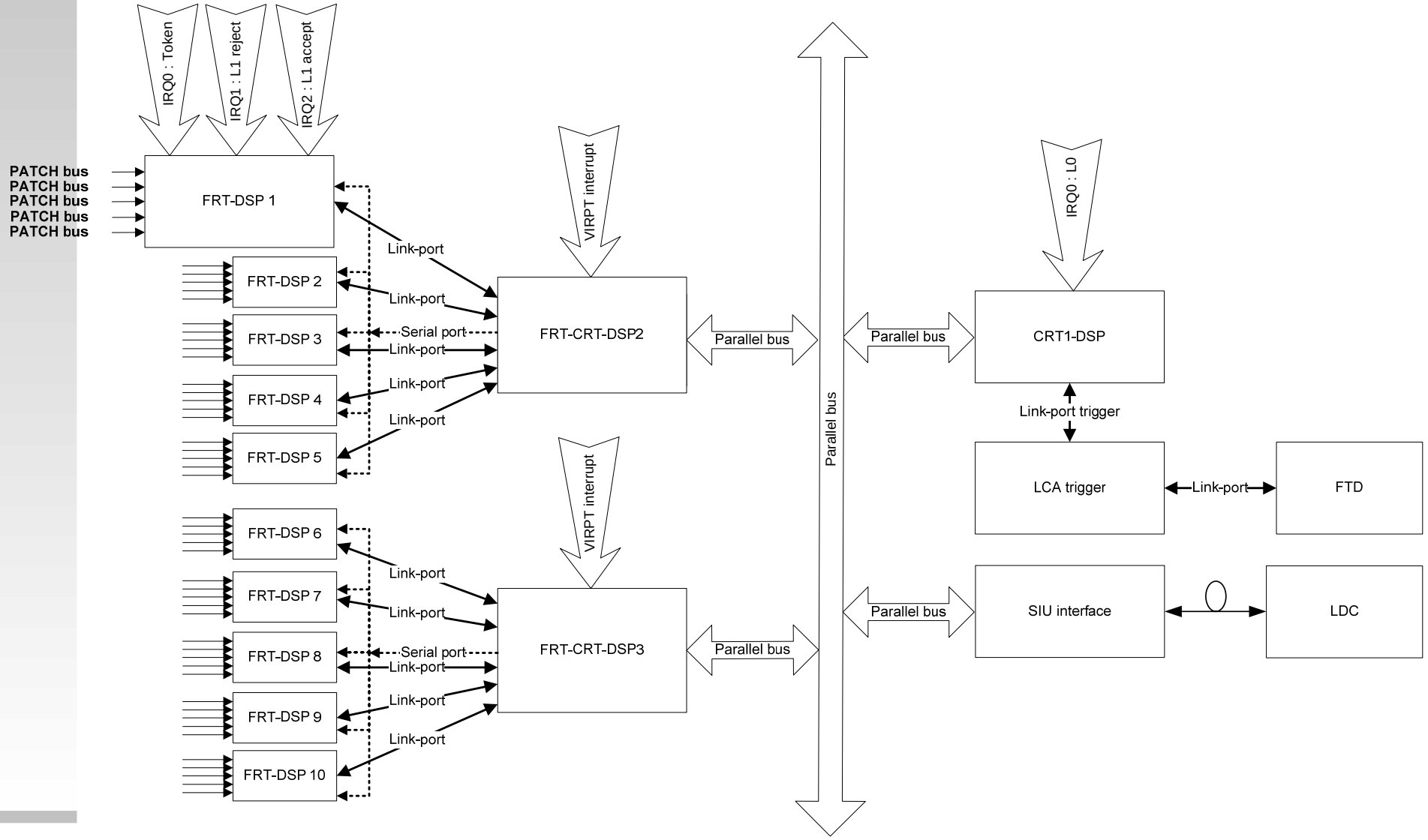
FEE board



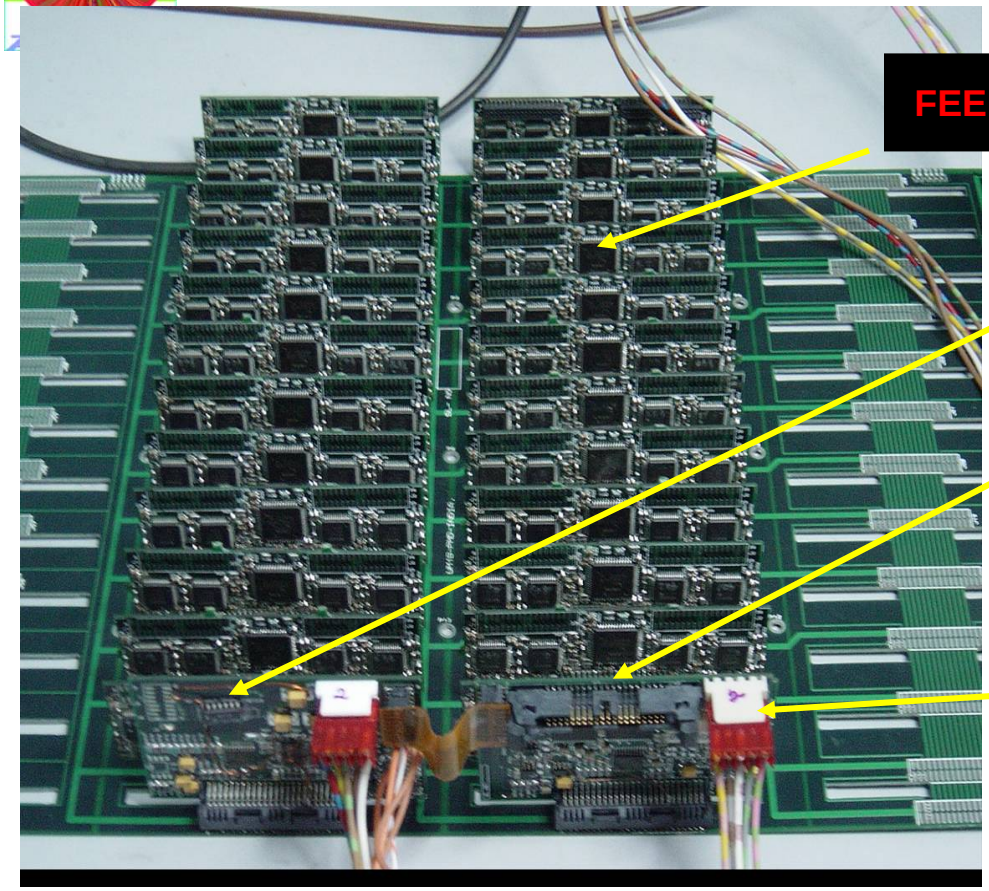
Main Components:

1. MANAS (Multiplexed-Analog-Signal-processor)
2. MARC (Muon-Arm-Readout-Chip)
3. ADC (Analog to Digital Converter)

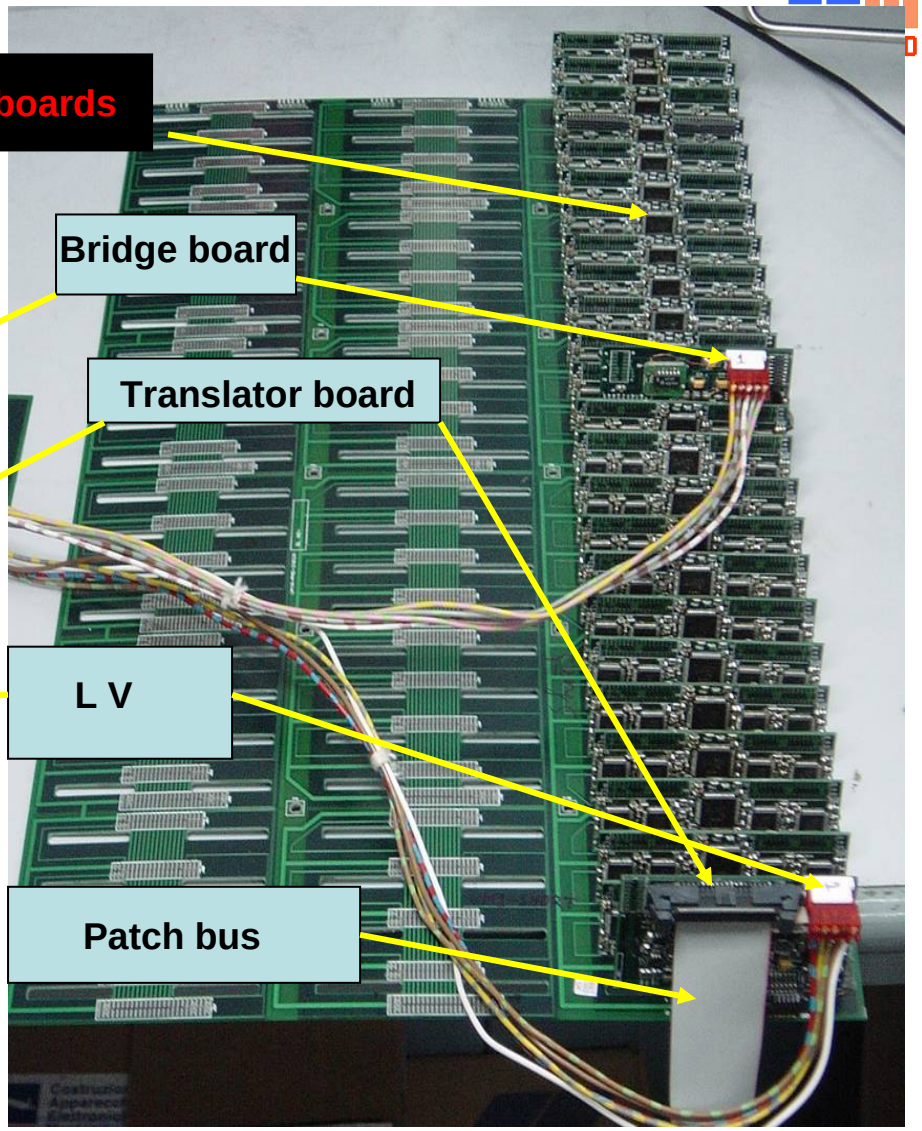
CROCUS SYSTEM MAIN SIGNALS & I/O PORTS



Chain to FEE board connection



FEE board arrangement in short type module



FEE board arrangement in long type module

DATA WORD

MANAS Adr. 0	0	0
MANAS Adr. 1	0	1
MANAS Adr. 2	1	0
MANAS Adr. 3	1	1

P = Parity bit / M = MANU Address (11 Bit)

G = MANAS Address

C= Channel number / D = Data (12 Bit)

32	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1
P	0	0	M	M	M	M	M	M	M	M	M	M	M	G	G	C	C	C	C	D	D	D	D	D	D	D	D	D	D	D	D

Output Word