

Cluster Processor Module

Status and Test Plan

CPM status

- Schematic sent before Christmas
- PCB manufactured in January
- PCB Back in February
- Module assembled in March
- Now: Visual inspection (pb with connector- back to RAL on Wednesday)
- Shorts checking and boundary scan after receipt of the BP

After preliminary tests, to do:

Time

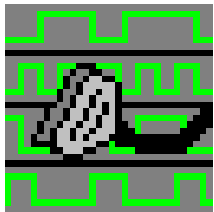


- Check clock(s) quality
- VME access
- Logical device codes:
 - I2C access (TTCrx dec)
 - Cluster Finding
 - Serialiser
- Real Time Data
- Time Slice Data

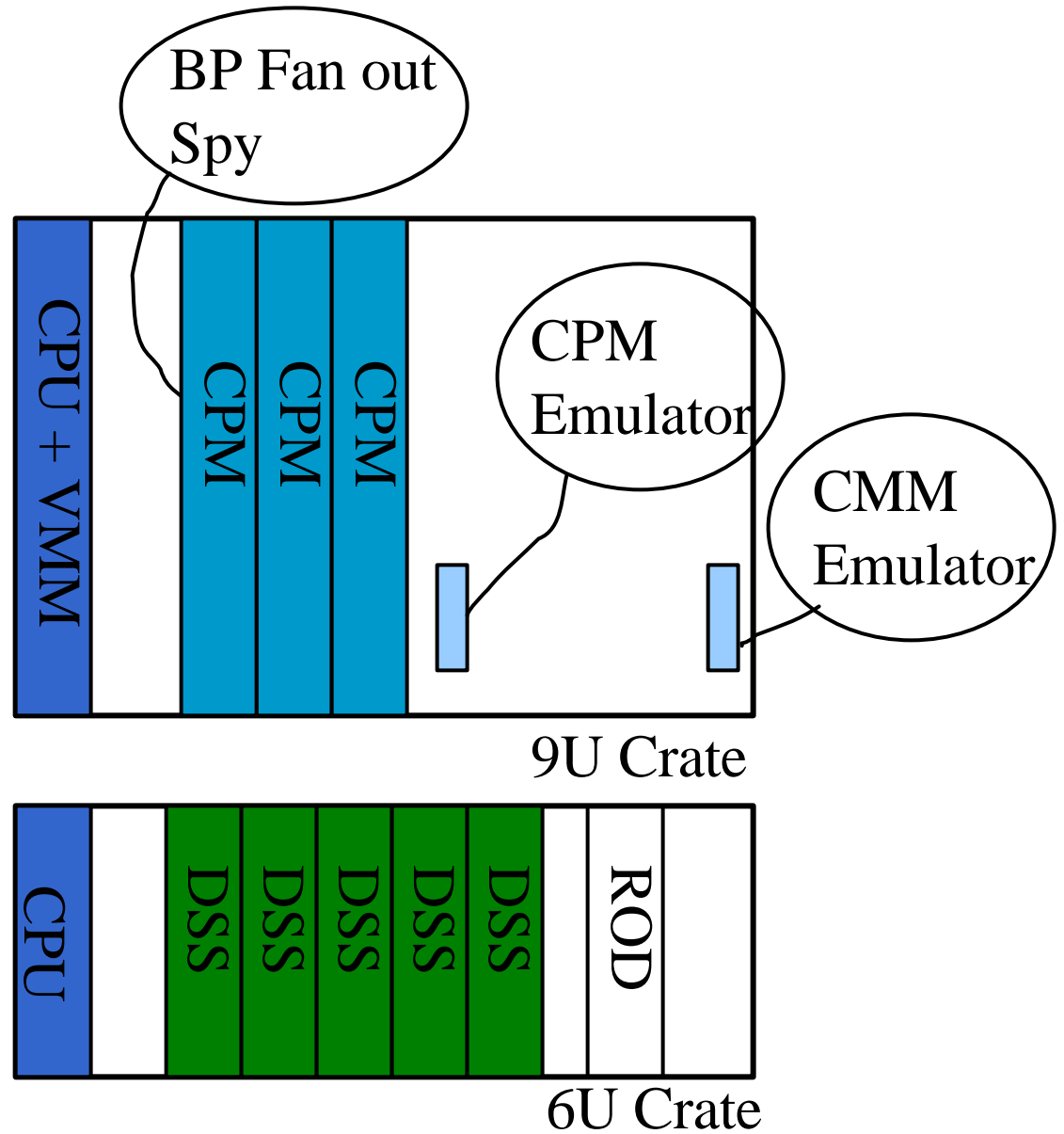
In the Lab



Linux RH7.2
HDMC



Logic
Analyzer
TLA611



H/W Status

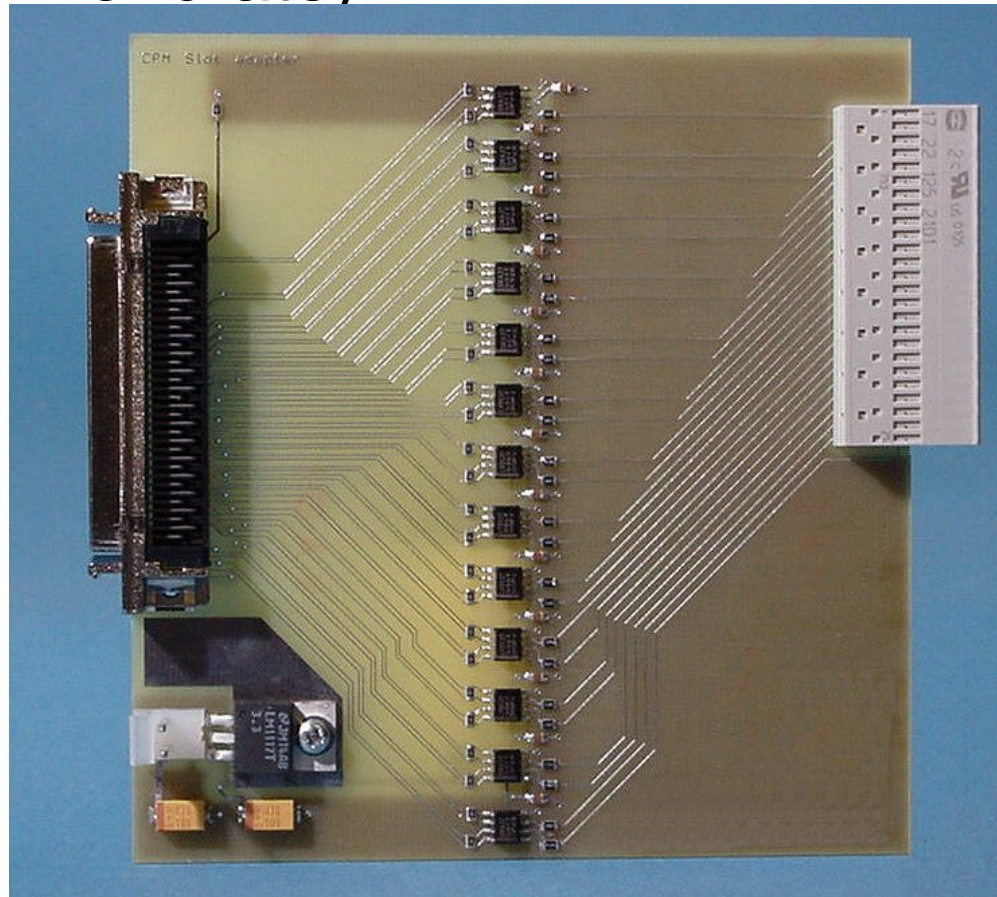
- One CPM, 3 more PCBs waiting for assembly
- CPM Emulator (For CMM): done
- CMM Emulator (For Real Time Data path testing): Layout nearly ready
- Back Plane Spy: Layout to be done, very simple, but not modular.

H/W Status

- One CPM emulator (CMM Testing) (Top or Bottom of crate):

DSS +LVDS
board → {

Supply {



} Backplane
(CMOS)

F/W Status

- F/W nearly completed: TTCrx controller, Version Register, etc...
- Receive CP model code:
 - Take 3 weeks to run it locally:
 - Understanding
 - Missing libraries
 - Version compatibility
 - Interest:
 - Understand CP logic
 - In case of CP H/W problem, we can write a test bench to reproduce and identify it
 - Save time for James to correct it if it is a F/W problem

S/W Status

- Modules Services have to be written
- CPM parts for HDMC nearly fully written
- Submodule layout has been used for CP chips and Serialiser Chips

HDMC Panel for CPM

Hardware Diagnostic, Monitoring and Control System (Version 0.2)

File Scripts Parts Windows Help

cpm.parts	Init. State	Attributes	State String
[-] Dummy Bus	accessible		""
[-] CPMRegisters	accessible	0x000000	""
[-] CPM CpSubmodule	uninitialised	0x07000	""
[-] Chip[0:0]<0>	uninitialised	0x7000	""
[-] Chip[0:1]<1>	uninitialised	0x7800	""
[-] Chip[0:2]<2>	uninitialised	0x8000	""
[-] Chip[0:3]<3>	uninitialised	0x8800	""
[-] Chip[0:4]<4>	uninitialised	0x9000	""
[-] Chip[0:5]<5>	uninitialised	0x9800	""
[-] Chip[0:6]<6>	uninitialised	0xA000	""
[-] Chip[0:7]<7>	uninitialised	0xA800	""
[-] CPChipRegisters	accessible	0x00	""
[-] CpFwVR	accessible	0x0; CPM.CpFwVR; Read/Write	""
[-] CpFwCR	accessible	0x01; CPM.CpFwCR; Read/Write	""
[-] CpFwSR	accessible	0x2; CPM.CpFwSR; Read/Write	""

Part: Assembler, Create Part

View: Crate, Create View

Connections

Help, Exit

View Name

Next Steps

- Still learning CP F/W
- Complete Modules Services for CPM
- H/W testing of CPM