Test Plan for the Production Version of the CMM.

Ian Brawn, 23 November 2007

General Notes

These tests will be used both as part of module-acceptance tests on receipt of CMMs from manufacture, and also later, as part of the procedure to diagnose faults and confirm that repaired modules are fully functional. Each CMM should be subjected to all of the tests in this document before being handed over to the customer. The tests should be carried out by a test engineer in the RAL ESS laboratory. Testing should take approximately one day per CMM.

These tests are designed to identify any physical faults in a CMM, such as broken PCB tracks or poor solder joints. They are not designed to test the CMM firmware, which has been previously tested and is running in factory-tested devices. Thus the programmable devices on the CMM have all of their interfaces tested, but not all of their internal functionality.

Further tests, not documented here, will be conducted by the customer on receipt of the CMMs: the functionality of each module will be confirmed in a test rig where each CMM will be tested with all of its interfaces operating concurrently. This will require a realistic environment, including neighbouring modules, which is under the control of the online data acquisition suite. The environment will provide precisely controlled inputs, including level-1 accepts at high rates with known timing, and the CMM readout will be compared bit-by-bit with predicted outputs obtained by detailed simulation of module algorithms. These tests by the customer will also check for low-rate errors and link stability over a sustained period.

In addition to the CMM, tests 13 to 15 here also fully test the CMM RTM.

All data and address values shown in this document are given as hexadecimal numbers, unless stated otherwise.

The Tests

1. Solder bridges:

1.1. Set solder bridges SB18–SB21, SB27–SB30 and SB34–SB37 to define module ID B, which comprises the Hardware Revision and Module Serial numbers as shown in Table 1. Making a solder bridge pulls the corresponding bit to ground. The fields should be defined as follows:

- Null: all bits grounded.
- Hardware Revision number: which is 5 for the production CMMs.
- Module Serial Number: a unique, incremental serial number for each CMM of the current revision, starting at 1.

Table 1. Module ID B. As an example, to set Spare ID = 0, Hardware Revision = 5 and Module Serial Number = 1F, make the links marked with an ‘X’.

bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
bridge	70	69	68	67	18	19	20	21	27	28	29	30	34	35	36	37
	X	X	X	X	X		X		X	X	X					
Field	Spare ID				Hardware Revision number				Module Serial Number							

1.2. Set solder bridges SB43–SB66 on the rear of the board to define the TTC & I2C addresses for the module as follows:
- Of the odd-numbered bridges on the top row, make only bridge SB61 (third from right) to pull TTC_DOUT(2) to VVC.
- Of the even-numbered bridges on the bottom row, make all bridges **except** SB62 (third from right), to pull all the other address lines to ground.

This defines the TTC I2C ID as 04, for all CMMs.

2. Power

2.1. Remove fuse FS1 and adjust the 1.8V power supply via variable resistor RV2.

2.2. Replace fuse FS1 and power up the module.

2.3. Check for smoke.

2.4. Measure the 5V, 3.3V, 2.5V and 1.8V power supplies and check that they are in spec (± 5%).

2.5. Confirm power LEDs are illuminated.

3. Boundary Scan

3.1. Boundary scan the three JTAG chains on the CMM. This will test approximately half the nets on the board for shorts and broken connections (although not at full operating frequency).

4. Configuring the CPLDs

4.1. Use Xilinx Impact to configure the following CPLDs via the JTAG chain:
- U70: XCR3384XL (1st in JTAG chain): System ACE Interface
  – use `cmm_ace_interface_01.jed`
- U71: XCR3384XL (2nd in JTAG chain): VME controller
  – use `vme_interface_08.jed`

Refer to the schematics further information on the JTAG chain.

5. Clock

5.1. Power down the CMM.

5.2. Ensure that the TTC system is correctly set up in the processor crate.

5.3. Plug the TTCDec daughter card into the CMM.

5.4. Power on the CMM. The 40 MHz oscillator on the TTCDec should now be providing a clock to the module. Check that the following LEDs are lit:
- CLK (40 MHz clock present),
- DLR (DAQ G-link locked)
- RLR (RoI G-link locked).
- TTC (TTC Ready)

6. VME Access

6.1. Read the following VME registers and check that they return the expected values.
- Module ID Register A (**address 0000**) should return **2417**
- Module ID Register B (**address 0002**) should return **050*x***,
  where *x* is the module serial number set in action 1.1 above.

6.2. Verify that the CMM Board Select LED, labelled BS, lights up temporarily as each VME access is made.

7. Geographic Address Decryption

7.1. Power down the CMM.

7.2. If the CMM is not in CMM slot 0 of the processor crate, move it there.

7.3. Power up the CMM.

7.4. Using the dial on the backplane, set the processor crate number to that shown in Table 2 (initially, use the value shown on the top line). Note that the crate number required depends upon the type of crate in use: production or prototype. This is because the polarity of some of the geographic address signals differs between the two types of crate.

7.5. **Write** to the Control Pulse Register (**address 0006**) the **data 0001** to reset the CMM.

7.6. **Read** the Control Mode Register (**address 0004**) and check that the contents match that shown in Table 2 for the current crate number and slot position.

7.7. Repeat steps 7.4 to 7.6 for each crate number shown in Table 2 in which the CMM is in slot CMM 0.

7.8. Power down the CMM and move it to CMM slot 1. Please note the following:
- The CMM cannot be hot-swapped: it must be powered down before being moved between slots.
- CMM slots 0 and 1 are not the same as CPM/JMM slots 0 and 1. The two sets of slots are distinct and are labelled as such on the processor backplane.

7.9. Repeat steps 7.4 to 7.6 for each crate number shown in Table 2 in which the CMM is in slot CMM 1.

7.10. Once finished, power down the Crate and move the CMM back to slot 0. This will ensure better access to probe the board, should anything go horribly wrong in the following tests.

Table 2. Geographical address decription for the production and prototype versions of the crate.

Crate Number		**Slot Number**	**Control Mode Register**
Production	**Prototype**		
0	7	CMM 0	0000
1	6	CMM 0	0004
2	5	CMM 0	0008
4	3	CMM 0	0010
0	7	CMM 1	0002

Table 3. The cmm_atst firmware collection: Geographical address vs. firmware load for production and prototype versions of crate.

Crate Number		Slot Number	Configuration Name	Control Mode Register	Crate FPGA ID Register	System FPGA ID Register
Production	**Prototype**					
0	7	CMM 0	cmm_id0	0000	CF00	5F00
1	6	CMM 0	cmm_id0	0004	CF00	5F00
2	5	CMM 0	cmm_id0	0008	CF00	5F00
3	4	CMM 0	cmm_id1	000C	CF01	5F01
4	3	CMM 0	cmm_id4	0010	CF04	5F04
5	2	CMM 0	cmm_id5	0014	CF05	5F05
0	7	CMM 1	cmm_id2	0002	CF02	5F02
1	6	CMM 1	cmm_id2	0006	CF02	5F02
2	5	CMM 1	cmm_id2	000A	CF02	5F02
3	4	CMM 1	cmm_id3	000E	CF03	5F03
4	3	CMM 1	cmm_id6	0012	CF06	5F06
5	2	CMM 1	cmm_id7	0016	CF07	5F07

Table 4. The standard CMM firmware collection (cmm_*xx*): geographical address vs. firmware load for production and prototype versions of crate. Shown for information only.

Crate Number		Slot Number	Configuration Name	Control Mode Register	Crate FPGA ID Register	System FPGA ID Register
Production	**Prototype**					
0	7	CMM 0	ey_crate	0000	*xx*00	*xx*00
1	6	CMM 0	ey_crate	0004	*xx*00	*xx*00
2	5	CMM 0	ey_crate	0008	*xx*00	*xx*00
3	4	CMM 0	ey_system	000C	*xx*04	*xx*04
4	3	CMM 0	e_crate	0010	*xx*02	*xx*02
5	2	CMM 0	e_system	0014	*xx*06	*xx*06
0	7	CMM 1	t_crate	0002	*xx*00	*xx*00
1	6	CMM 1	t_crate	0006	*xx*00	*xx*00
2	5	CMM 1	t_crate	000A	*xx*00	*xx*00
3	4	CMM 1	t_system	000E	*xx*04	*xx*04
4	3	CMM 1	jet_crate	0012	*xx*01	*xx*01
5	2	CMM 1	jet_system	0016	*xx*05	*xx*05

8. FPGA Configuration, Geographic Address Override and Firmware Selection

8.1. Power up the CMM and insert a blank flash card. Check that the AERR LED is illuminated. Power down the CMM and remove the flash card.

8.2. From the CMM firmware archive (http://www.te.rl.ac.uk/esdg/atlas-flt/firmware%20folder/CMM-V3/CMM_V3_firmware.htm) download a copy of the file cmm_acetest.zip. Unpack the zip and copy the contents to the flash card, such that the root directory of the card contains the file xilinx.sys and the folder cmm_atst. This is the System ACE collection that will be used for the following tests.

8.3. With the CMM powered off, insert the flash card into the CMM.

8.4. Power up the CMM. You should observe the following behaviour of the CMM LEDs:

- ASTAT should blink on and off for a duration of less than 30 seconds, indicating that configuration is in progress. Once configuration is finished, ASTAT should be lit permanently.
- ALL_DONE should be lit permanently once configuration is finished (as indicated by ASTAT).
- AERR should not light up at any time.

If behaviour other than this is observed, consult the Xilinx System ACE data sheet, DS080 at www.xilinx.com.

8.5. Run the Labview programme **SystemAceTest.vi.** This will test the following areas of functionality: FPGA configuration, firmware selection as a function of geographic address, and geographic address override.

8.6. SystemAceTest.vi will return the contents of the Crate and System FPGA ID registers as a function of the geographic address of the module. Verify these values match those shown in Table 3.

8.7. Repeat this test twice, to be sure.

9. VME Status Register and FPGA DLLs

9.1. From the CMM firmware archive (http://www.te.rl.ac.uk/esdg/atlas-flt/firmware%20folder/CMM-V3/CMM_V3_firmware.htm) download a copy of the current CMM System ACE collection. This will be a zip file named cmm_*xx*.zip, where *xx* is the highest revision number available. Unpack the zip and copy the file cp_sys.ace into the root directory of the CMM flash card. Ensure that this directory contains no other .ace or .sys files.

This is the normal CMM-CP System-level firmware. It is necessary for the next two tests.

9.2. **Read** the VME Status Register (**address 0008**). Check that the following bits are set:

- 5: Crate FPGA loaded,
- 6: System FPGA loaded,
- 8: Crate FPGA DLL locked,
- 9: System FPGA DLL locked.

9.3. **Write** to the Control Pulse Register (**address 0006**) the **data 0100** to reset the FPGA DLLs.

9.4. **Read** the Status Register (**address 0008**) again and check that the DLLs have re-locked.

10. VME Register and Memory Access Tests

10.1. Make sure the CMM is configured as CMM-CP system-level module (see section 9.1)

10.2. Open the labview programme **ALL_CMM_TESTS.vi** and select 'YES' to enable the following functions:
- Check Read / Write Registers
- Check Read Only Registers
- Check Memories
- Write Report.

10.3. Run the programme ALL_CMM_TESTS.vi. The results will be written to the file [*root*]/TestReports/CMM_ALL_TEST_RESULTS.txt, where [*root*] is the root directory path shown in the ALL_CMM_TESTS GUI.

10.4. Review the file ALL_CMM_TESTS.vi and check that no failures are reported.

11. Firmware Load

11.1. With a Xilinx download cable connect the CMM FPGA JTAG chain to a PC running Xilinx Impact software.

11.2. Using Impact in boundary-scan mode, download the following firmware directly to the FPGAs:
- Crate FPGA: crt_tst_rtio
- System FPGA: sys_tst_rtio

11.3. Verify from the Impact display that the configuration process completes successfully.

12. Crate FPGA to System FPGA data transmission

12.1. This test requires the diagnostic firmware contained in tst_rtio.ace. Copy this file from the CMM firmware archive (http://www.te.rl.ac.uk/esdg/atlas-flt/firmware%20folder/CMM-V3/CMM_V3_firmware.htm) to the root directory of the CMM flash card and delete any xilinx.sys file from this directory. This will cause System ACE to always use tst_rtio.ace to configure the CMM FPGAs, independent of the CMM's geographic (configuration) address. (For more information about System ACE file structures see the Xilinx System ACE data sheet: http://direct.xilinx.com/bvdocs/publications/ds080.pdf).

12.2. Insert the flash card into the CMM to trigger the re-configuration of the FPGAs.

12.3. **Write** to the Control Mode register (**address 0004**) the **data 0000**. This selects the local crystal oscillator as the ROD clock source, which avoids having to initialize the TTC system.

12.4. Run the Labview programme **CrateFPGA2SystemFPGADataTx.vi**. This programme will run three test patterns through two sets of Crate-System FPGA tracks, making six tests in total. After each test the programme will display the data captured in the System FPGA memories and ask you to visually inspect them. The data caputed should be as described below.

- Due to the width of the VME bus & CMM memories the data will be split across two memory blocks, like this:
  - LSB Block: 16 least-significant bits of data;
  - MSB Block: 9 most-significant bits of data.
- The following test patterns should be visible in the data:
  - ramp, 16-bits wide, repeating in time and repeating across the data width every 16-bits;
  - walking bit, repeating in time and repeating across the data width every 16-bits;
  - alternating 1's and 0's across all data.
- Each of these patterns should be displayed in turn (one per test) in System Memory 1, while System Memory 2 is zero. Each pattern should be displayed in System Memory 2, while System Memory 1 is zero

13. Crate FPGA Cable Output and RTM

13.1. The tst_rtio.ace firmware should be loaded for this test (see section 12.1).

13.2. Set up a DSS with an LVDS-Receiver daughter card (GIO Rx) in position 2 (the lower daughter card). Connect the GIO Rx to socket 1 (the uppermost socket) of the CMM RTM using a 1.5 m long cable. Make sure that the DSS is being driven from the TTC clock.

13.3. Run the Labview programme **CrateFPGACableOutputTest.vi**. This programme will run three test patterns from the Crate FPGA though the cable to the DSS. After each test you will be asked to visually inspect the data captured in the DSS (see below). The programme will then instruct you to move the RTM–DSS cable to socket 2 on the RTM, after which it will re-transmit the three test patterns. Again visual inspection of the data will be required.

13.4. The data captured in the DSS should have the following features.

- Due to the width of the DSS memories the data will be split across two memory blocks as shown below (note that the two RTM sockets carry data of different widths).
  - Memory A: 16 least-significant bits of data;
  - Memory B: 10 most-significant bits of data when using RTM socket 1
    6 most-significant bits of data when using RTM socket 2
- The following test patterns should be visible in the data:
  - ramp, 16-bits wide, repeating in time and repeating across the data width every 16-bits;
  - walking bit, repeating in time and repeating across the data width every 16-bits;
  - alternating 1's and 0's across all data.
- Each of these patterns should be displayed in turn for each position of the cable.

14. System FPGA Output

14.1. This test requires the CMM to be loaded with the tst_rtio.ace firmware (see section 12.1).

14.2. **Write** to the Control Mode register (**address 0004**) the **data 0006**. This will select the local crystal oscillator as the clock source and set the geographic address of the CMM to be 3, corresponding to a system-level module.

14.3. Using the Labview programme **FillMemoryWithPatternABC.vi**, write **pattern A** to the following:
- base address **C000**, for **400** locations

14.4. The CMM should now be transmitting test pattern A to the two front-panel output sockets (SK1 and SK2). Each should receive the same pattern. Unfortunately there is no way to test this other than with a scope. For each connector, check both pins of all the differential pairs and verify that the output is as shown in Table 5.

Table 5. Output expected on both front-panel sockets, SK1 and SK2, for the System FPGA output tests.

Socket Pin Pair	Location on front-panel	Expected Output
(34/68)	Top of socket	Ground
(26/60) – (33/67)		8-bit ramp with LSB on pair (26/60)
(18/52) – (25/59)		8-bit ramp with LSB on pair (18/52)
(10/44) – (17/51)		8-bit ramp with LSB on pair (10/44)
(2/36) – (9/43)		8-bit ramp with LSB on pair (2/36)
(1/35)	Bottom of socket	40 MHz clock

15. System FPGA Cable Input and RTM

15.1. This test requires the DSS to be loaded with the tst_rtio.ace firmware (see section 12.1).

15.2. Set up a DSS with an LVDS-Transmitter daughter card (GIO Tx) in position 1 (the uppermost daughter card). Connect the GIO Tx to socket 1 (the uppermost socket) of the CMM RTM using a 1.5 m long cable. Make sure that the DSS is being driven from the TTC clock.

15.3. On the **CMM**: **write** to the Control Mode register (**address 0004**) the **data 0020**. This will select TTC clk40des as a clock source and set the geographic address of the CMM to be 0, corresponding to a crate-level module.

15.4. On the **CMM**: **read** the Control Mode register (**address 0004**) to verify that bit 5, TTC Clock Enable has been set by the above write. If it has not this may indicate that the TTC is not ready.

15.5. On the **CMM**: **write** to the I^2C Control register (a**ddress0040**) the **data 239B**. This will enable clk40des2, via an I2C write to the TTCrx .

15.6. On the **CMM**: **write** to the Cable Output Control register (**address 0038**) the **data 0000** to ensure the Cable Output Enable bit is clear.

15.7. On the **DSS**: using the Labview programme **dss-registers.vi**, assert the **Reset** for **Daughter Card 1**, and keep asserting it until the 2nd green LED from the top of Daughter Card 1 comes on and stays on. This shows the daughter card is in the correct mode for this test. Changing the mode of the daughter card is a hit-and-miss affair for reasons beyond the scope of this document.

15.8. On the **CMM**: **write** to **address 0020** the **data 0001**. This will set the System Record Enable bit of the System Record Control register (see Table 12). Data will now be captured in the System Cable Input Memories. These memories roll over, so they will always contain the most recent 256 data slices.

15.9. On the **CMM**: **write** to **address 0020** the **data 0000**, to clear the System Record Enable bit. This will freeze the data in the System Local Input Memories and allow them to be read out from VME.

15.10. Using the labview programme **ReadCMMMemory.vi**, read System Cable Input Memory 1 and verify that it has received correctly the test data from the DSS. See section 15.13 for the VME map of the System Cable Input Memories, and section 15.14 for the features that the captured data should display.

15.11. Move the DSS–RTM cable from socket 1 of the RTM to socket 2 and repeat steps 15.8–15.10, except this time examine the data in System Cable Input Memory 2.

15.12. Move the DSS–RTM cable from socket 3 of the RTM to socket 2 and repeat steps 15.8–15.10, except this time examine the data in System Cable Input Memory 3.

15.13. The System Cable Input Memories are mapped to VME as follows:

- address 5800–59FC: 16 least significant bits of System Cable Input Memory 1
- address 5A00–5BFC: 11 most significant bits of System Cable Input Memory 1
- address 5C00–5DFC: 16 least significant bits of System Cable Input Memory 2
- address 5E00–5FFC: 11 most significant bits of System Cable Input Memory 2
- address 6000–61FC: 16 least significant bits of System Cable Input Memory 3
- address 6200–63FC: 10 most significant bits of System Cable Input Memory 3

A full VME map for the tst_rtio firmware is given in Table 12.

15.14. The data captured in the CMM memories should have the following features.

- The data will be split in width across two memory blocks as shown in section 15.13. Note that the data does not occupy the full width of both memory blocks. Any unoccupied bits within a memory block are always located at the most-significant end of the data bus (above the data) and should be ignored for this test.
- The least significant bit of the data can be ignored (in the tst_rtio firmware it is a clock).
- The other bits of the data should show a walking bit pattern. This will appear only intermittently in the data (i.e., regularly but not continuously) as the data width over which the bit walks is greater than that received by the CMM.
- A single discontinuity in the pattern may be visible. This will be the point at which data capture was stopped (there is no co-ordination between the starting and stopping of data capture, the roll-over of the CMM memories, or the roll-over of the test pattern).
- The important feature of the pattern is that, in those parts of the memory where the walking bit is present, each bit should be asserted in turn (allowing for the unused bits and one possible discontinuity as described above).

16. Crate FPGA Backplane Input

16.1. This test requires the CMM to be loaded with the tst_rtio.ace firmware (see section 12.1).

16.2. Power down the CMM crate. If the CMM is not in CMM slot 0, move it there. Please note:

- The CMM cannot be hot-swapped: it must be powered down before being moved between slots.
- CMM slots 0 and 1 are not the same as CPM/JMM slots 0 and 1. The two sets of slots are distinct and are labelled as such on the processor backplane.

16.3. Plug the CPM Emulator into CPM/JMM slot 0 in the crate. Connect a cable from the LVDS-Transmitter daughter card (GIO Tx) on a DSS to the CPM emulator. This cable should be 1.5m long.

16.4. Power up the CMM and DSS.

16.5. **Write** to the Control Mode register (**address 0004**) the **data 0020**. This will select TTC clk40des as a clock source and set the geographic address of the CMM to be 0, corresponding to a crate-level module.

16.6. On the **CMM**: **Read** the Control Mode register (**address 0004**) to verify that bit 5, TTC Clock Enable has been set by the above write. If it has not this may indicate that the TTC is not ready.

16.7. On the **CMM**: **write** to both Backplane Timing registers A and B (**addresses 0016** and **0018**) the **data AAAA**. This will set the phase of the clocks used to latch the incoming data to be offset from the CMM and DSS system clocks by 180^{o}.

16.8. On the **DSS**: using the Labview programme **dss-registers.vi**, assert the **Reset** for **Daughter Card 1**, and keep asserting it until the 2^{nd} green LED from the top of Daughter Card 1 comes on and stays on. This shows the daughter card is in the correct mode for this test. Changing the mode of the daughter card is a hit-and-miss affair for reasons beyond the scope of this document.

16.9. On the **CMM**: **write** to the Crate Record Control register (**address 001C**) the **data 0001**. This will set the CMM Crate Record Enable bit and data will now be captured in the Crate Input Memories. These memories roll over, so they will always contain the most recent 256 data slices.

16.10. On the **CMM**: **write** to the Crate Record Control register (**address 001C**) the **data 0000**, to clear the Crate Record Enable bit. This freezes the data in the Crate Input Memories and enables them to be read out from VME.

16.11. Using the labview programme **ReadCMMMemory.vi**, read the Crate Input Memory corresponding to the CPM/.JEM slot in use (see Table 6) and verify that it has received correctly the test pattern loaded to the DSS. See section 16.12 for the features that the captured data should display.

16.12. The data captured in the CMM memories should have the following features.

- The data will be split in width across two memory blocks as shown in section 15.13. Note that the data does not occupy the full width of both memory blocks. Any unoccupied bits within a memory block are always located at the most-significant end of the data bus (above the data) and should be ignored for this test.
- The least significant bit of the data can be ignored (in the tst_rtio firmware it is a clock).
- The other bits of the data should show a walking bit pattern. This will appear only intermittently in the data (i.e., regularly but not continuously) as the data width over which the bit walks is greater than that received by the CMM. Note that the bit will walk from the most-significant to the least-significant bit in the bus (due to a feature of the CPM Emulator).
- A single discontinuity in the pattern may be visible. This will be the point at which data capture was stopped (there is no co-ordination between the starting and stopping of data capture, the roll-over of the CMM memories, or the roll-over of the test pattern).
- The important feature of the pattern is that, in those parts of the memory where the walking bit is present, each bit should be asserted in turn (allowing for the unused bits and one possible discontinuity as described above).

16.13. Power down the CMM crate and move the CPM Emulator to the next CPM/JMM slot. Repeat steps 16.4–16.11 for each CPM/JMM slot in the crate, each time examining the data in the corresponding Crate Input Memory blocks (see Table 6).

Table 6. The VME map of the Crate Input Memories.

Address	Data Captured
1000–11FC:	16 LSBs from CPM/MM slot 0
1200–13FC:	8 MSBs from CPM/MM slot 0
1400–11FC:	16 LSBs from CPM/MM slot 1
1600–13FC:	8 MSBs from CPM/MM slot 1
1800–11FC:	16 LSBs from CPM/MM slot 2
1A00–13FC:	8 MSBs from CPM/MM slot 2
1C00–11FC:	16 LSBs from CPM/MM slot 3
1E00–13FC:	8 MSBs from CPM/MM slot 3
2000–11FC:	16 LSBs from CPM/MM slot 4
2200–13FC:	8 MSBs from CPM/MM slot 4
2400–11FC:	16 LSBs from CPM/MM slot 5
2600–13FC:	8 MSBs from CPM/MM slot 5
2800–11FC:	16 LSBs from CPM/MM slot 6
2A00–13FC:	8 MSBs from CPM/MM slot 6
2C00–11FC:	16 LSBs from CPM/MM slot 7
2E00–13FC:	8 MSBs from CPM/MM slot 7
3000–11FC:	16 LSBs from CPM/MM slot 8
3200–13FC:	8 MSBs from CPM/MM slot 8
3400–11FC:	16 LSBs from CPM/MM slot 9
3600–13FC:	8 MSBs from CPM/MM slot 9
3800–11FC:	16 LSBs from CPM/MM slot 10
3A00–13FC:	8 MSBs from CPM/MM slot 10
3C00–11FC:	16 LSBs from CPM/MM slot 11
3E00–13FC:	8 MSBs from CPM/MM slot 11
4000–11FC:	16 LSBs from CPM/MM slot 12
4200–13FC:	8 MSBs from CPM/MM slot 12
4400–11FC:	16 LSBs from CPM/MM slot 13
4600–13FC:	8 MSBs from CPM/MM slot 13
4800–11FC:	16 LSBs from CPM/MM slot 14
4A00–13FC:	8 MSBs from CPM/MM slot 14
4C00–11FC:	16 LSBs from CPM/MM slot 15
4E00–13FC:	8 MSBs from CPM/MM slot 15

17. G-link Output

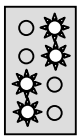
17.1. Make sure a **jumper** is fitted to **PL2** on the CMM. This provides power to the laser transceivers on the CMM.

17.2. This test requires the CMM to be loaded with the **tst_glo** firmware: remove the file tst_rtio.ace from the root directory of the CMM flash card and replace it with tst_glo.ace, which can be downloaded from the CMM firmware archive (http://www.te.rl.ac.uk/esdg/atlas-flt/firmware%20folder/CMM-V3/CMM_V3_firmware.htm).

17.3. Make sure that the switch on the front of the DSS clock daughter card is in the lower position.

17.4. Use a pair of fibre cables to connect the CMM DAQ and RoI G-link outputs to a G-link Optical Receiver daughter board on a DSS.

17.5. Check that the LEDs on the G-link Optical Receiver board show that the G-links have locked. They should look like this:



17.6. On the **CMM**: **write** to the DAV Length register (**address 001E**) the **data 0010**. This defines the length of the readout events transmitted by the CMM (in clock cycles).

17.7. On the **DSS**: using the Labview programme **FillDSSMemoryWithPatternABC.vi**, clear all memories for the Optical Receiver daughter card.

17.8. On the **DSS**: using the Labview programme **dss-registers.vi**, in the Mother Control register set the bit Test Mode 1, and clear the bits Test Mode 2 and Test Mode 3. This will put the DSS into Test Mode 1.

17.9. On the **DSS**: using the Labview programme **dss-registers.vi**, in the Mother Control register set the bit Start Test.

17.10. On the **CMM**: **write** the data shown for Test 1 in **Table 7** to the CMM G-link Select register (**address 0016**). This register only exists in the tst_glo version of the CMM firmware (see Table 13 for the full VME map of this firmware). It controls which G-link is used to transmit data for each test, and whether the data are sent directly from the Crate FPGA or via the System FPGA.

17.11. On the **CMM**: using the labview programme **FillMemoryWithPatternABC.vi** write **Test Pattern B** to the following memory block:
- address 6C00 for 200 locations

17.12. On the **CMM**: **write** to the CMM Go register (**address 0010**) the **data 0001**, to initiate the G-link data transfer. (This simulates the effect an L1A signal on the CMM.)

17.13. On the DSS: using the labview programme **ReadDSSMemory.vi**, read the memories for the two G-link channels connected to the CMM. For the G-link selected in step 17.10 the memory should contain the first 16 (decimal) words of Test Pattern B, which is a walking bit. Note that the G-link is only 20 bits wide, so only bits 0–19 of the test pattern will be present. The memories for the unused G-link channel should be clear.

17.14. On the **CMM**: **write** to the CMM Go register (**address 0010**) the **data 0001** six more times, and monitor the G-link Active LED on the CMM for the G-link channel in use (i.e. the LED labelled DD for the DAQ G-link or RD for the RoI G-link). Verify that it lights up temporarily as the G-link transmits data.

17.15. On the **DSS**: Re-read the memories for the G-links connected to the CMM. Those for the G-link selected in step 17.10 should now contain the first 112 (decimal) words of Test Pattern B (walking bit); those for the other G-link should still be clear.

17.16. Repeat the above tests, steps 17.10— 17.15 for Tests 2—4 shown in Table 7.

Table 7. G-link tests.

Test	Data Source & G-link	Contents of G-link Select Register (address 0016)
1	CRT FPGA & DAQ G-link	0000
2	SYS FPGA & DAQ G-link	0002
3	CRT FPGA & RoI G-link	0001
4	SYS FPGA & RoI G-link	0003

18. Miscellaneous LEDs

18.1. This test requires the CMM to be loaded with the tst_glo firmware (see section 17.2).

18.2. For each test given in **Table 8**, **write** the data shown to the Crate LED register (**address 0020**) and the System LED register (**address 0024**) and check that the corresponding LED is illuminated.

18.3. **Write** to both the Crate LED register (**address 0020**) and the System LED register (**address 0024**) the **data 0000** and check that none of the LEDs listed in Table 8 are illuminated.

Table 8. LED tests.

Test	Crate LED Reg address 0020)	System LED Reg (address 0024)	LED Illuminated
1	0001	0000	EF: DAQ FIFO Empty
2	0002	0000	FF: DAQ FIFO Full
3	0004	0000	CHT: Crate Hits
4	0008	0000	PER: Parity Error
5	0000	0001	SHT: System Hits

19. The TTC System and the VME-I2C TTCrx Interface

19.1. Power up the CMM crate and the TTC system and check the TTC LED in the CMM front panel is illuminated.

19.2. **Write** to the TTCrx Control register (**address 0040)** the data **0000**, to initiate an I2C read of Fine Delay register 1.

19.3. **Read** the TTCrx Status register (**address 0042**) and check that it contains the **data 0000**.

19.4. **Write** to the TCrx Control register (**address 0040**) the **data 2034**. This should initiate an I2C write to the TTCrx Fine Delay register 1 (of data 34).

19.5. **Write** to the TTCrx Control register (**address 0040**) the **data 2256**. This should initiate an I2C write to the TTCrx Course Delay register (of data 56).

19.6. **Write** to the TTCrx Control register (**address 0040**) the **data 0000**, to initiate an I2C read of Fine Delay register 1.

19.7. **Read** the TTCrx Status register (**address 0042**) and check that it now contains the **data 0034**.

19.8. **Write** to the TTCrx Control register (**address 0040**) the **data 0200**, to initiate an I2C read of the Course Delay register.

19.9. **Read** the TTCrx Status register (**address 0040**) and check that it now contains the **data 0056**.

20. CAN Microcontroller

20.1. This test requires the Accemic debugging tool. Download and install if necessary.

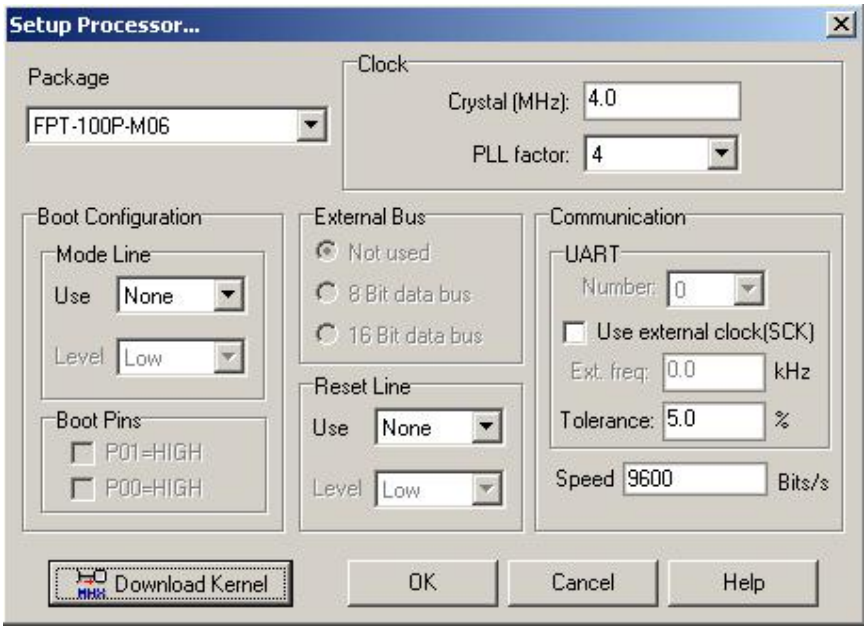
20.2. If the CAN microcontroller on the TCM has already been programmed skip straight to step 20.15, otherwise plod on through steps 20.3–20.14.

20.3. Connect the CAN programming cable to the "CAN bus PROG" socket on the TCM front-panel.

20.4. On the **TCM**: **write** to the Control Register (VME **address 0004**) the **data 0003**. This will put the microcontroller into programming mode.

20.5. Start the Acemic software tool. When the “wait 10 sec” window has disappeared, under the Preferences menu, select Processor. Select MB90F594A/G and Fujitsu in the dropdown menus and click Next.

20.6. Set the processor with the following values:



20.7. Press the Download Kernel button. Ignore any messages about the microprocessor mode and just press OK.

20.8. If the Setup Processor dialogue box doesn't disappear when the download is complete just press the OK button. Again, ignore any messages about the microprocessor mode.

20.9. On the **TCM**: **write** to the Control Register (VME **address 0004**) the **data 0001**. This will put the microcontroller into programming mode. This will put the microcontroller into normal mode.

20.10. Within Acemic, from the File menu, select Load File and navigate to the generic.mhx file (you'll need to view files of type Any).

20.11. When Acemic has has finished loading, under the Tools menu select Automatic Start and check the “Automatic start after reset” box. Press OK.

20.12. Press the Run Application button (a blue triangle pointing right).

20.13. The ITx and IRx LEDs on the TCM front-panel should now be flashing: a single flash approximately every six seconds.

20.14. Within Acemic, from the Start menu select Disconnect, then remove the CAN programming cable from the TCM.

20.15. Connect CAN programming cable from the TCM to the CAN PROG socket on the **CMM** front-panel.

20.16. On the **CMM**, fit a jumper to **PL5** (bottom left of PCB, component side) and press the **CAN RST** button on the front-panel. This will put the microcontroller into programming mode.

20.17. Within the Acemic software, under the Preferences menu, select Processor. Select MB90F594A/G and Fujitsu in the dropdown menus and click Next. Set the processor with the same values as shown for 20.6.

20.18. Press the Download Kernel button. Ignore any messages about the microprocessor mode and just press OK.

20.19. If the Setup Processor dialogue box doesn't disappear when the download is complete just press the OK button. Again, ignore any messages about the microprocessor mode.

20.20. On the CMM remove the jumper from PL5 and press the CAN RST button again.

20.21. Within Acemic, from the File menu, select Load File and navigate to the generic.mhx file (you'll need to view files of type Any).

20.22. When Acemic has finished loading, under the Tools menu, select Automatic Start and check the “Automatic start after reset” box. Press OK.

20.23. Press the Run Application button (a blue triangle pointing right).

20.24. You should now observe the following behaviour from the TCM ITx and IRx LEDs: they should flash twice, within a duration of about 1 second. There should then be a much longer interval, of about six seconds, before they flash twice again. This cycle will repeat indefinitely.

20.25. Within Accemic, select the I/O box and view the contents of data register 15 for CAN 0, DTR0(15). The data should approximate those shown in Table 9. A difference of less than 4 from the expected value can be regarded as correct.

Table 9. Contents of CAN 0 Data Register 15

Buffer No.	**Expected Value dec (hex)**	**Meaning**
1	92 (5C)	Supply voltage on 1.8 V line × 51.1 (dec)
2	130 (82)	Supply voltage on 2.5 V line × 51.1 (dec)
3	168 (A8)	Supply voltage on 3.3 V line × 51.1 (dec)
4	11 (B)	Temperature of RoI G-link × 51.1 (dec)
5	11 (B)	Temperature of DAQ G-link × 51.1 (dec)
6	30 (E2)	Temperature of Crate FPGA × 51.1 (dec)
7	30 (E7)	Temperature of System FPGA × 51.1 (dec)
8	20 (E3)	Temperature of I2C FPGA × 51.1 (dec)

21. Loading the CMM Trigger firmware

21.1. The final stage in commissioning the CMM is to load it with the firmware it will use in ATLAS. This is the collection named cmm_*xx*, (where *xx* is the current firmware revision number) which was tested in section 9. If this collection is still present on the flash card, copy the xilinx.sys file from the cmm_xx directory into the root directory of the card, overwriting any previous versions of the file. This will instruct System ACE to use the cmm_xx collection when configuring the CMM

21.2. If the cmm_xx collection is not already present on the flash card it can be obtained from the CMM firmware archive (http://www.te.rl.ac.uk/esdg/atlas-flt/firmware%20folder/CMM-V3/CMM_V3_firmware.htm). Download the file cmm_xx.zip, where *xx* is the highest revision number available. Unpack this and copy the contents to the flash card, such that the root directory of the card contains the file xilinx.sys and the folder cmm_xx. Ensure any previous xilinx.sys file in this directory is overwritten.

21.3. Insert the flash card into the CMM.

21.4. Power up the CMM and check that it configures (see section 8).

21.5. Power down the CMM.

22. Test Patterns

The following patterns are referred to in the above tests.

Table 10. Test Patterns.

Test Pattern A: byte-wide ramp	Test Pattern B: walking bit	Test Pattern C: oscillating bits
0000	0001	0000
0101	0002	FFFF
0202	0004	0000
0303	0008	FFFF
0404	0010	0000
0505	0020	FFFF
0606	0040	0000
0707	0080	FFFF
0808	0100	0000
0909	0200	FFFF
0A0A	0400	0000
0B0B	0800	FFFF
:	:	:
etc.	etc.	etc.

23. VME Address Maps

Table 11. VME address map for an e/γ-system CMM.

Byte Address (hex)	Register type	Register Name	Size in bytes	Description
00000	RO	ModuleIdA	2	Module ID Register A
00002	RO	ModuleIdB	2	Module ID Register B
00004	RW	ControlModeReg	2	Control Mode Register
00006	RW	ControlPulseReg	2	Control Pulse Register
00008	RO	StatusReg	2	Status Register
0000A	RO	FifoStatusReg	2	FIFO Status Register
0000C	RO	BpEReg	2	Backplane Parity Error Register
0000E	RO	CEReg	2	Cable Link Parity Error Register
00010	RW	BpDisReg	2	Backplane Link Disable Register
00012	RW	CdisReg	2	Cable Link Disable Register
00014	RO	PCReg	2	Parity Count Register
00016	RW	BpTimingRegA	2	Backplane Timing Register A
00018	RW	BpTimingRegB	2	Backplane Timing Register B
0001A	RW	CTimingReg	2	Cable Link Timing Register
0001C	RW	PipeDelay	2	Pipeline Delay Register
0002E	RW	DaqSliceReg	2	DAQ Slice Register
00020	RW	DaqOffsetBpData	2	DAQ addr offset, backplane data
00022	RW	DaqOffsetCrtReslts	2	DAQ addr offset, crate results
00024	RW	DaqOffsetCblData	2	DAQ addr offset, cable data
00026	RO	DaqOffsetSysReslts	2	DAQ addr offset, system results
00028	RO	RoiOffsetBpData	2	RoI addr offset, backplane data
0002A	RO	RoiOffsetCrtReslts	2	RoI addr offset, crate results
0002C	RO	RoiOffsetCblData	2	RoI address offset, cable data
0003E	RO	RoiOffsetSysReslts	2	RoI address offset, system results
00030	RO	FfAddrCrtDaq	2	FIFO address pointers, crate DAQ
00032	RO	FfAddrSysDaq	2	FIFO address pointers, system DAQ
00034	RO	FfAddrCrtRoi	2	FIFO address pointers, crate RoI
00036	RO	FfAddrSysRoi	2	FIFO address pointers, system RoI
00038	RW	CblOutCntrl	2	Cable Output Control Register
00040	RW	TtcrxControl	2	TTCrx Control Register
00042	RO	TtcrxStatus	2	TTCrx Status Register
00050	RO	CmmCId	2	Crate FPGA firmware version
00052	RO	CmmSId	2	System FPGA firmware version

00054	RO	I2cId	2	I2C FPGA firmware version
00056	RO	VmeId	2	VME CPLD firmware version
0005C	RW	CanAccessA	2	CAN Access Register A
0005E	RW	CanAccessB	2	CAN Access Register B
00060	RO	SumEtThr1Reg	2	Sum-ET Threshold 1
00062	RO	SumEtThr2Reg	2	Sum-ET Threshold 2
00064	RO	SumEtThr3Reg	2	Sum-ET Threshold 3
00066	RO	SumEThr4Reg	2	Sum-ET Threshold 4
00068	RO	JetEtThr1Reg	2	Jet-ET Threshold 1
0006A	RO	JetEtThr2Reg	2	Jet-ET Threshold 2
0006C	RO	JetEtThr3Reg	2	Jet-ET Threshold 3
0006E	RO	JetEThr4Reg	2	Jet-ET Threshold 4
001FA	RW	TtcI2Cid	2	TTCrx I2C ID
001FC	RO	TtcBrcst	2	TTC BRCST data register
001FE	RO	TtcDq	2	TTC DQ register
00200	RO	TtcDump	32	TTC Dump RAM
01000	RW	DprCrtFpgaIput	16384	DPR Access: Crate FPGA Input Data
05000	RW	DprCrtFpgaOput	2048	DPR Access: Crate FPGA Output Data
05800	RW	DprSysFpgaIput	3072	DPR Access: System FPGA Input Data
06400	RW	DprSysFpgaOput	2048	DPR Access: System FPGA Output Data
06C00	RW	FifoCrtFpgaIput	16384	FIFO Access: Crate FPGA Input Data
0AC00	RW	FifoCrtFpgaOput	2048	FIFO Access: Crate FPGA Output
0B400	RW	FifoSysFpgaIput	3072	FIFO Access: Sys FPGA Input Data
0C000	RW	FifoSysFpgaOput	2048	FIFO Access: Sys FPGA Output Data
0C800	RO	DprRoi	3072	DPR Access: RoI Data
0D400	RO	FifoRoi	3072	FIFO Access: RoI Data
0E000	RO	DprMainJetEtLut	8192	Main Jet-ET LUT
010000	RO	DprFwdJetEtLut	8192	Forward Jet-ET LUT
012000	RO	MissEtLut	16384	Missing-ET LUT

Table 12. VME address map for crt_tst_rtio & sys_tst_rtio.

Byte Address (hex)	Register type	Register Name	Size in bytes (hex)	Description
00000	RO	ModuleIdA	2	Module ID Register A
00002	RO	ModuleIdB	2	Module ID Register B
00004	RW	ControlModeReg	2	Control Mode Register
00006	RW	ControlPulseReg	2	Control Pulse Register
00008	RO	StatusReg	2	Status Register
00016	RW	BpTimingRegA	2	Backplane Timing Register A
00018	RW	BpTimingRegB	2	Backplane Timing Register B
0001A	RW	CTimingReg	2	Cable Link Timing Register
0001C	RW	CrtRcrdCntrl	2	Crate Record Control Register
00020	RW	SysRcrdCntrl	2	System Record Control Register
00022	RW	CrtLoopReg	2	Crate Loop Register
00024	RW	SysLoopReg	2	System Loop Register
00038	RW	CblOutCntrl	2	Cable Output Control Register
00040	RW	TtcrxControl	2	TTCrx Control Register
00042	RO	TtcrxStatus	2	TTCrx Status Register
00050	RO	CmmCId	2	Crate FPGA firmware version
00052	RO	CmmSId	2	System FPGA firmware version
00054	RO	I2cId	2	I2C FPGA firmware version
00056	RO	VmeId	2	VME CPLD firmware version
0005C	RW	CanAccessA	2	CAN Access Register A
0005E	RW	CanAccessB	2	CAN Access Register B
001FA	RW	TtcI2Cid	2	TTCrx I2C ID
001FC	RO	TtcBrcst	2	TTC BRCST data register
001FE	RO	TtcDq	2	TTC DQ register
00200	RO	TtcDump	32	TTC Dump RAM
01000	RW	DprCrtFpgaIput	100	CRT Input DPR 0, LSBs
01200	RW	DprCrtFpgaIput	100	CRT Input DPR 0, MSBs
01400	RW	DprCrtFpgaIput	200	CRT Input DPR 1, LSBs + MSBs
01800	RW	DprCrtFpgaIput	200	CRT Input DPR 2, LSBs + MSBs
01C00	RW	DprCrtFpgaIput	200	CRT Input DPR 3, LSBs + MSBs
02000	RW	DprCrtFpgaIput	200	CRT Input DPR 4, LSBs + MSBs
02400	RW	DprCrtFpgaIput	200	CRT Input DPR 5, LSBs + MSBs
02800	RW	DprCrtFpgaIput	200	CRT Input DPR 6, LSBs + MSBs
02C00	RW	DprCrtFpgaIput	200	CRT Input DPR 7, LSBs + MSBs

03000	RW	DprCrtFpgaIput	200	CRT Input DPR 8, LSBs + MSBs
03400	RW	DprCrtFpgaIput	200	CRT Input DPR 9, LSBs + MSBs
03800	RW	DprCrtFpgaIput	200	CRT Input DPR 10, LSBs + MSBs
03C00	RW	DprCrtFpgaIput	200	CRT Input DPR 11, LSBs + MSBs
04000	RW	DprCrtFpgaIput	200	CRT Input DPR 12, LSBs + MSBs
04400	RW	DprCrtFpgaIput	200	CRT Input DPR 13, LSBs + MSBs
04800	RW	DprCrtFpgaIput	200	CRT Input DPR 14, LSBs + MSBs
04C00	RW	DprCrtFpgaIput	200	CRT Input DPR 15, LSBs + MSBs
05800	RW	DprSysFpgaIput	100	SYS Cable Input DPR 1, LSBs
05A00	RW	DprSysFpgaIput	100	SYS Cable Input DPR 1, MSBs
05C00	RW	DprSysFpgaIput	200	SYS Cable Input DPR 2, LSBs + MSBs
06000	RW	DprSysFpgaIput	200	SYS Cable Input DPR 3, LSBs + MSBs
06400	RW	DprSysFpgaIput	200	SYS Local Input DPR 1, LSBs + MSBs
06800	RW	DprSysFpgaIput	200	SYS Local Input DPR 2, LSBs + MSBs
06C00	RW	DprCrtFpgaOput	100	CRT Local Output DPR 1, LSBs
06E00	RW	DprCrtFpgaOput	100	CRT Local Output DPR 1, MSBs
07000	RW	DprCrtFpgaOput	200	CRT Local Output DPR 2, LSBs + MSBs
07400	RW	DprCrtFpgaOput	200	CRT Cable Output DPR 1, LSBs + MSBs
07800	RW	DprCrtFpgaOput	200	CRT Cable Output DPR 2, LSBs + MSBs
0C000	RW	DprSysFpgaOput	100	SYS Output DPR 1, LSBs
0C200	RW	DprSysFpgaOput	100	SYS Output DPR 1, MSBs
0C400	RW	DprSysFpgaOput	200	SYS Output DPR 2, LSBs + MSBs

Table 13. VME address map for crt_tst_glo & sys_tst_glo

Byte Address (hex)	Register type	Register Name	Size in bytes	Description
00000	RO	ModuleIdA	2	Module ID Register A
00002	RO	ModuleIdB	2	Module ID Register B
00004	RW	ControlModeReg	2	Control Mode Register
00006	RW	ControlPulseReg	2	Control Pulse Register
00008	RO	StatusReg	2	Status Register
00010	RW	GoReg	2	Readout Go Register
00016	RW	GlinkSelReg	2	G-link Selecr Register
0001E	RW	DavLengthReg	2	DAV Length Register
00020	RW	CrtLEDReg	2	Crate FPGA LED Register
00024	RW	SysLEDReg	2	System FPGA LED Register
00022	RW	LoopReg	2	Loop Register
00040	RW	TtcrxControl	2	TTCrx Control Register
00042	RO	TtcrxStatus	2	TTCrx Status Register
00050	RO	CmmCId	2	Crate FPGA firmware version
00052	RO	CmmSId	2	System FPGA firmware version
00054	RO	I2cId	2	I2C FPGA firmware version
00056	RO	VmeId	2	VME CPLD firmware version
0005C	RW	CanAccessA	2	CAN Access Register A
0005E	RW	CanAccessB	2	CAN Access Register B
001FA	RW	TtcI2Cid	2	TTCrx I2C ID
001FC	RO	TtcBrcst	2	TTC BRCST data register
001FE	RO	TtcDq	2	TTC DQ register
00200	RO	TtcDump	32	TTC Dump RAM
06C00	RW	TestRAM	100	Test RAM, bits 0-15
06E00	RW	TestRAM	100	Test RAM, bits 16-19

24. System ACE Collection

Table 14. CMM System ACE collection.

Configuration Address	Configuration Name	Crate Firmware	System Firmware
0	ey_crate	cmm_cpcrt_vc	cmm_cpsys_null
1	ey_system	cmm_cpcrt_vs	cmm_cpsys
2	t_crate	cmm_cpcrt_vc	cmm_cpsys_null
3	t_system	cmm_cpcrt_vs	cmm_cpsys
4	e_crate	cmm_ecrt_vc	cmm_esys_null
5	e_system	cmm_ecrt_vs	cmm_esys
6	jet_crate	cmm_jetcrt_vc	cmm_jetsys_null
7	jet_system	cmm_jetcrt_vs	cmm_jetsys