# RCC Functional Requirements

| Revision | Date | Author | Remarks |
|---|---|---|---|
| 0.1 | 15 August 2000 | J.C.Hill | Initial version created from various existing documents |
| 0.2 | 18 August 2000 | J.C.Hill | Revision based on comments from Tom Meyer |
| 0.3 | 25 August 2000 | J.C.Hill | Further minor revision |
| 0.4 | 28 August 2000 | J.C.Hill | Extra requirement for human interface after comment from Andy Parker |

## 1. Introduction

This document describes the requirements of the SCT and Pixel RCC (ROD Crate Controller). The information has been taken from a variety of other documents, but is meant to be usable in a standalone way, except where specific references are made to other documents.
I have included DAQ requirements even though it is not obvious that all the DAQ functionality will be in the RCC. Maybe at a later date (some of) these need to be moved to another document?

## 2. Requirements

1. RCC crate interface.

   The RCC must have a VME64x-compatible master interface.

   Justification:
   VME64x is the ATLAS standard interface for ROD crates. Both the ROD and TIM have VME64x slave interfaces, so the RCC must have a compatible interface to be able to communicate with them, and it must act as the crate master.

2. RCC external interface.

   The RCC must have an interface to the LAN.

   Justification:
   To allow communication with other RCCs and the ATLAS DAQ. The exact technology to be used is not yet defined, but is expected to be a commercial standard such as Fast Ethernet.

3. RCC VME modes of operation

   The RCC must support A32 and A24 modes of operation, and be able to do programmed I/O, block transfers and handle interrupts.

   Justification:
   A32 and A24 are the usual modes of addressing in VME. Both the ROD and TIM will use A32, but for flexibility it is necessary that A24 be possible. Block transfers may be useful with the ROD. Both the ROD and the TIM may generate interrupts under certain conditions.

4. Basic RCC access to ROD

   The RCC must be able to access the registers on the ROD (read and/or write as necessary).

   Justification:
   RCC has to communicate with the ROD (eg. configuration, monitoring, event access) via the handful of registers provided for this purpose. The access is via standard VME memory mapping.

5. RCC initialisation of ROD.

   The RCC must be able to initialise and configure a ROD.

Justification:
The RCC has to set a ROD up in the appropriate way depending on what it is expected to do (eg. normal running, calibration, etc.).

6. Basic RCC access to BOC

   The RCC must be able to access the threshold and timing adjustments on the BOC.

   Justification:
   The RCC has to be able to set up the BOC for correct operation.
   Note:
   Access to the BOC will be indirectly via the associated ROD over the setup bus.

7. RCC setup of synchronisation of front end signals.

   The RCC must be able to adjust the timing on the BOC with respect to data from front end modules.
   Justification:
   The timing of data from the front ends will vary due to different cable lengths, varying time of flight of particles from the interaction, etc. The BOC must clock in the data on the same clock pulse and at a phase that guarantees the data is correct.

8. RCC access to DCS

   RCC will communicate with DCS where necessary.
   Justification:
   The “select” line, which chooses the primary or secondary fibre on a module and the “hard reset” line, which performs a power-up reset on a module are both controlled via the low voltage units, and hence via DCS. In addition, any calibrations that change DCS settings will require communication with DCS.
   Note:
   The exact details of RCC-DCS interaction are still the subject of much debate.

9. RCC control of calibrations

   The RCC must setup the various components for a calibration and control the operation and handling of results.
   Justification:
   Some calibrations will be controlled by a ROD running autonomously, but the RCC must still deal with the result of the calibration. In other cases, the RCC will have to communicate with DCS (eg. to change low and/or high voltage settings) – in these cases RCC will drive the calibration also.

10. RCC access to event data

    The RCC must be able to access event data on the RODs at up to 1kHz.

Justification:
In local running, data will be read from the RODs into the RCC as the normal data flow route. In global running, the RCC must be able to sample data for monitoring purposes. 1kHz is considered a reasonable rate, but the requirement could still be met at a lower rate.
Note:
No minimum rate that is considered acceptable has been specified so far. .

11. RCC access to histograms

The RCC must be able to read and zero histograms accumulated on the ROD.

Justification:
The ROD will build histograms from the data flowing through it. The RCC must be able to read these histograms for monitoring and/or calibration purposes, and for storing in a reference database. The RCC must also be able to zero a histogram – for example if a front end module has been disabled and the effect of this needs to be checked.

12. RCC access to ROD error counters

The RCC must be able to read and zero error counters on ROD.

Justification:
The ROD will count various errors. These must be accessible for monitoring purposes. The RCC will also have to forward them to the central DAQ (eg. for the status displays). The RCC must also be able to zero an error counter, for example after a problem has been fixed.

13. RCC access to ROD test memories

The RCC must have read and write access to the test memories on the ROD.
Justification:
The RCC must be able to debug a ROD. There are various test memories included on the ROD to allow testing of a part of it independently of other areas – these must be accessible in read and write modes to allow test patterns to be written and checked.

14. RCC control of ROD

The RCC must be able to reset and/or turn off a ROD completely.
Justification:
If a ROD is misbehaving badly enough (eg. not responding to commands) , the RCC must be able to reset it. *In extremis,* if this fails to recover the situation, the RCC must be able to disable a ROD completely if it is causing problems (eg. corrupting the data going up the S-link or generating a very high continuous data rate).

15. RCC monitoring of ROD.

    The RCC must monitor the behaviour of each ROD.
    Justification:
    To know whether a ROD is misbehaving, the RCC must monitor it – is it lagging behind the triggers, is it logging an abnormal number of errors? The exact parameter space is not possible to determine at this time.

16. Basic RCC access to TIM

    The RCC must be able to access the registers on the TIM (read and/or write where appropriate).
    Justification:
    The RCC has to communicate with the TIM (set up trigger conditions, monitor behaviour etc.) via the registers on the TIM. These registers are standard VME memory-mapped.

17. RCC initialisation of TIM.

    The RCC must be able to initialise and configure the TIM.

    Justification:
    The TIM needs to be set up in the correct mode (eg. for local or global running).

18. RCC monitoring of TIM

    The RCC needs to monitor the operation of the TIM.
    Justification:
    The RCC needs to check that the TIM is performing correctly.

19. TIM generating fast commands

    The RCC needs to be able to generate fast commands in the TIM via a single VME command.
    Justification:
    Fast Commands are just that – their generation should be with the minimum delay possible, which means in this case that only a single VME command should be required.

20. Basic RCC communication with central DAQ.

    The RCC will need to communicate with the central DAQ.
    Justification:
    The RCC will be the only module in the VME crate with direct communication to the central DAQ. It must act on Run Control commands, send back the correct responses, make information available and interact with the databases.

21. RCC access to databases

The RCC needs read and write access to the various local and global databases.

Justification:
The RCC will need to read the configuration database to know how to set up the RODs, BOCs and TIM. It will need to write calibration results to a database. Some databases may be local, with an external agent keeping local and global databases in synchronisation.

22. RCC making error counters available.

The RCC will send the error counters from the various components in the ROD crate to the central DAQ.
Justification:
The central DAQ will wish to display error counts on the Status Display (which will probably be a hierarchy of displays) – the RCC must make them available in the appropriate form.

23. RCC handling Run Control commands

The RCC must respond correctly to Run Control.
Justification:
The RCC is the interface between the central DAQ and the ROD crate hardware and software. It must react appropriately in response to Run Control commands. In local running, the Run Control may be running in the RCC, but the logical behaviour must be the same.

24. RCC sending histograms to central DAQ.

The RCC must send a histogram (either requested or automatically) to the central DAQ.
Justification:
The central DAQ may wish to monitor the whole detector by looking at the histograms from all RCC, or the histograms may be stored in a database (eg. at end of run).

25. RCC interaction with operator

There must be a human interface to the RCC.

Justification:
A local operator will want access to error counters, histograms and local events, either in a spy mode in global running, or as the run controller in local running. In global running especially, this will not involve Run Control itself in every case.