

VME Arbitration by the BM

25 July, 2000

1. Introduction

This document describes the control and timing of VME operations that occur during normal data taking. Two VME modules interact with the custom boards of the STT: the CPU and the VBD. These two modules are used for five tasks:

- 1) Downloading and Initialization
- 2) Testing
- 3) SCL_INIT
- 4) Monitoring during data taking
- 5) Readout of L3 data from the STT system.

Since tasks 1) and 2) do not occur during data taking they will not be considered further here.

The other tasks are complicated by the fact that the VBD disobeys standard VME protocol. Once it has been granted the VME bus at the start of the L3 data transfer cycle, it does not relinquish VME until it has finished its entire transfer for its crate. This requires additional bus arbitration during the three tasks that occur during normal data taking (SCL_INIT, monitoring, L3 readout). This arbitration is handled by the Buffer Manager (BM) element of the FRC.

2. Arbitration

The arbitration scheme used by the BM to grant the VME bus to only one at a time of the three tasks (SCL_INIT, monitoring, L3 readout) that want it is quite simple. Once a task has control of the bus, the BM will not send out a request for another task to start until the first one has finished. This assumes that all three tasks have equal priority. The conditions for granting a VME bus request from each of the three tasks are therefore:

- i) $grant(scl_init) = req(scl_init) \ \& \ !(srdy \parallel mon_req)$
- ii) $grant(monitor) = req(monitor) \ \& \ !(srdy \parallel init_req)$
- iii) $grant(vbd_xfer) = req(vbd_xfer) \ \& \ !(init_req \parallel mon_req)$