

# THE ATLAS READ OUT DATA FLOW CONTROL MODULE AND THE TTC VME INTERFACE PRODUCTION STATUS

Per Gällnö, CERN, Geneva, Switzerland

(email: [per.gallno@cern.ch](mailto:per.gallno@cern.ch))

## Abstract

The ATLAS detector data flow from the Front End to the Read Out Drivers (ROD) has to be controlled in order to avoid that the ROD data buffers get filled up and hence data getting lost. This is achieved using a throttling mechanism for slowing down the Central Trigger Processor (CTP) *Level One Accept* rate. The information about the state of the data buffers from hundreds of ROD modules are gathered in daisy-chained fan-in ROD-BUSY modules to produce a single *Busy* signal to the CTP. The features and the design of the ROD-BUSY module will be described in this paper.

The RD-12 TTC system VMEbus interface, TTCvi, will be produced by an external electronics manufacturer and will then be made available to the users via the CERN EP/ESS group. The status of this project is given.

## INTRODUCTION

### Dead-Time Control Review

The data flow in the ATLAS sub-detector acquisition systems needs to be controlled in order to prevent information losses in the case the data buffers in the Front End, Read Out Drivers (ROD) or Read Out Buffers (ROB) get saturated.[1],[2]

Three different mechanisms to control the data flow will be implemented:

- By *Back pressure* using a XON/XOFF protocol on the read-out links between the ROD's and the ROB's.
- By *Throttling* to slow down the level one (LVL1) trigger rate from the CTP when the ROD data buffers are nearly filled.
- By *Prevention* introducing a constant dead-time combined with one set by a pre-programmed algorithm in the CTP in order to avoid buffer overflow in the Front End. The constant dead-time is chosen to be 4 BC's after each LVL1 and the algorithm, called "leaky bucket", limits the number of LVL1 to 8 in any window of 80 $\mu$ s.[3]

The introduction of a dead-time by a *throttling* mechanism is based on a ROD busy signalling scheme informing the Central Trigger Processor about the state of the ROD data buffers as each ROD is able to produce a ROD-Busy signal when its buffer is filled up. The busy signals from each ROD are summed and monitored in ROD-Busy Modules connected in a tree structure to finally produce a veto signal for the CTP. The ROD Busy signalling scheme and associated hardware will be described in this context.

## ROD BUSY STATUS SUMMING AND MONITORING

### System Overview

The Read-Out Drivers (ROD), of which there will be several hundred in the ATLAS experiment, buffer, process and format the data from the Front End electronics before being sent to the Read-Out Buffers (ROB).

If the data buffers in the ROD are close to get filled up the Level-1 trigger rate must be reduced. A way of achieving this is to send a busy flag to the CTP to introduce a dead-time.[4]

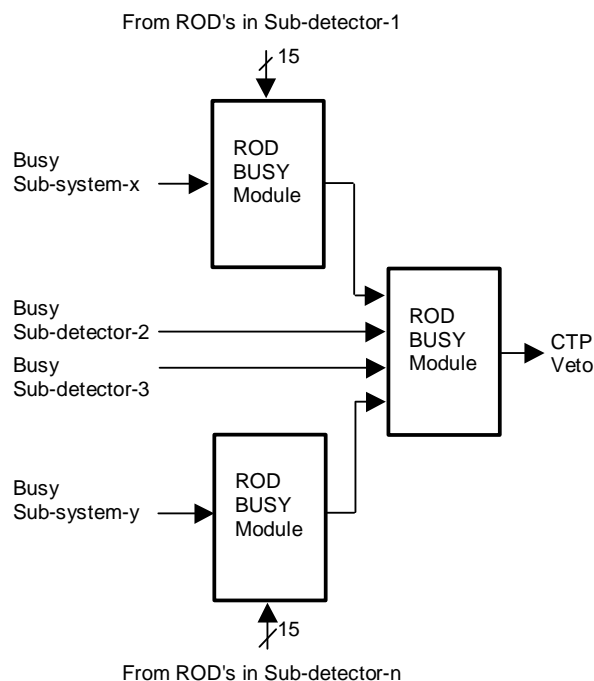


Figure 1. The ROD -Busy tree structure

Each ROD produces a Busy signal, which is sent to a ROD-Busy module together with Busy signals from other ROD's in the same sub-system. The ROD-Busy module sums the incoming Busy signals to produce one Busy signal of the particular sub-system. In turn the sub-system Busy signal is summed with other sub-system Busy signals in another Busy module to form a sub-detector Busy signal. Finally all sub-detector Busy signals are gathered to form a Busy input to the CTP.

## THE ATLAS ROD-BUSY MODULE FEATURES

### Basic Operation

The ROD-Busy module [5] has been designed to perform the following functionality:

- Collect and make a logical OR of up to 16 Busy input signals.
- Monitor the state of any input Busy signal.
- Mask off any input Busy signal in the case a ROD is generating a false Busy state.
- Measure the integrated duration any Busy input is asserted for a given time period.
- Store a history of the integrated Busy duration for each input.
- Generate an interrupt if any Busy input is asserted for longer than a pre-set time limit.
- Generate a Busy output serving as an input for a subsequent ROD-Busy module in the tree structure or as a veto for the CTP.

### Software Controlled Mode

In this mode of operation are the resetting and enabling of the integrating duration counters, as well as the resetting, writing and reading of the history FIFO buffers done entirely under program control. The FIFO empty and full status flags for each FIFO are available to the VMEbus.

### Circular Buffer Mode

In this mode of operation is the transfer of data from the Busy duration counters to the FIFO buffers controlled by a timed sequencer. Bits may be set in a register in order to allow a circular buffer operation, i.e. a word is read out from the FIFO for each word written when the FIFO full flag is present. The maximum time between two consecutive data transfers, from counter to FIFO, is 6.55 ms. This time may be adjusted in a 16 bit VME register.

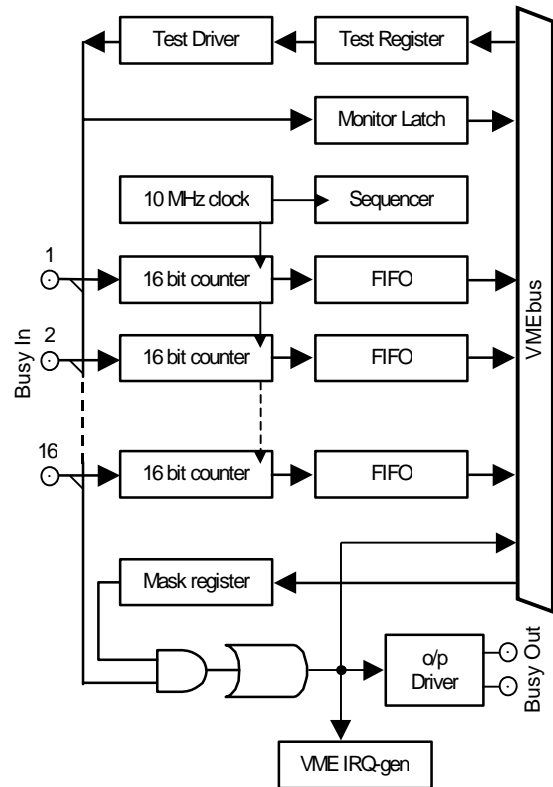


Figure 2. ROD -Busy module block diagram

### Additional Features

- Each input path may be tested by setting bits in a VME test register.
- A status bit reflects the state of the Busy Out.
- A bit may be set in a control register in order to turn on a global Busy signal on all Busy Outputs.
- The Busy Time-Out service requester may be controlled by software functions, i.e. enable, disable, set and clear of the service request.
- The VMEbus interrupter may be tested with a software function.
- The module may be globally reset by a software function.

### Modular VHDL blocks

The code for the different functional blocks has been written in VHDL and may be obtained on request in the case a designer wants to implement the Busy module functions directly in a ROD module.

The following VHDL entities are made available:

1. Input monitoring, masking, stimulating and summing.
2. Quad 16-bit up-counter.
3. FIFO read/write sequencer.
4. VME slave and interrupter interface.
5. Busy time-out service requester to drive interrupter.

## MODULE DESIGN DESCRIPTION

### *Input signal receivers and test drivers*

The inputs are terminated with a Thévenin network resulting in a 50Ω resistive input impedance and calculated to give a + 0.8 V idle voltage. A Busy TRUE input corresponds to a 0 V level and a Busy FALSE to a + 0.8 V level. The input voltage threshold is set to + 0.4 V and the ultra fast input comparators have an internal hysteresis circuit producing clean input signals even when receiving data over long lines. All inputs may be monitored by reading a 16 bit input status VME register. Each input may be tested by being pulled down by an internal open-collector driver connected in turn to a 16 bit VME test register.

### *Output signal drivers*

The four Busy Out outputs are driven by FAST TTL open-collector drivers. The outputs have the following characteristics and usage:

0. Pulled up to + 5 V by 10 kΩ and should be used to drive a following Busy Input or the CTP Busy Input.
  1. Same as 0.
  2. Pulled up to + 5 V by 510 Ω and should be used for monitoring purposes, i.e. oscilloscope etc.
  3. Same as 2.

### *Busy Input masking and summing*

The cleaned-up input signals drive the Busy Summing circuit and the Busy Duration counters. The input signals to the Summing circuit may be masked off in order to isolate faulty ROD units. The Summing circuit produces a global Busy signal, which is fed to the four Busy Out outputs. A control bit may be set to produce a global Busy Out for system test purposes. This block is implemented in a FPGA named *ip\_reg\_structure*.

### *Duration Counting*

The 16 bit duration counters increment at a speed of 10 MHz as long as there are Busy In signals on the inputs. There are global counter enable and reset functions generated by either accessing VME control bits or by the Buffer

Sequencer. The sixteen counters are implemented in four FPGA's named *quad\_count\_struct*.

### *Duration Count Buffering and Read-Out*

The 512 word deep FIFO's buffer the Duration Counter data until read out by the VMEbus. There are global FIFO write cycles and reset functions generated by either accessing VME control bits or by the Buffer Sequencer. The FIFO read cycles are either done by the VMEbus or by the Buffer Sequencer. Control bits enable the FIFO's to be configured as circular buffer, i.e. they maintain always the history of the 512 last entered Duration Count figures. If not configured as circular buffers the FIFO's will only contain the first 512 entered Duration Count figures.

### *Duration Counter/Buffer Sequencer*

The sequencer, when enabled, handles the control of the Duration counters and the FIFO's. A 16 bit down counter with a VME programmable shadow register, clocked by the 10 MHz clock, is used to set the rate for transferring the duration counts to the FIFO's. This block is implemented in a FPGA named *fifo\_sequencer*.

### *Global Busy Time-Out Service Requester*

The Time-Out circuit monitors the duration of the global busy signal and generates a service request if a certain time limit is reached. Two sets of 16 bit counters, magnitude comparators and VME programmable registers are used for this monitoring circuitry. An Interval counter/comparator/register circuit sets the frequency at which the two counters are reset. The Limit counter/comparator/register circuit, where the counter increments during the time the Busy is true, generates a service request, if the preprogrammed level is attained before being reset by the Interval circuit. Both counters are incremented at 10 MHz. The Time-Out service request may be programmed to trigger a VMEbus interrupt. This block is implemented in a FPGA named *sreq\_timer\_struct*.

### *VMEbus Data Bus Interface*

The VME bus slave interface is of conventional type and accepts only 16 bit word data cycles (D16). The addressing can either be standard or short (A24 or A16). Address pipelining and address only cycles are accepted. Four hexa-decimal switches are used

for setting the base address of the module. This block is implemented in a FPGA named *vme\_if*.

### *VMEbus Interrupt generator*

A VMEbus interrupt can be generated when a Time-Out service request occurs. A control register in which the VME Interrupt Request level is programmed and the interrupter is enabled controls the interrupt generator. Another register contains the Status/ID information in an 8 bit format (D<7..0>). This block is also implemented in the FPGA named *vme\_if*.

### *Module Configuration EEPROM*

Manufacturer/module identification and serial number, as well as module revision number should be stored in this non-volatile memory chip. There are spare locations for storing supplementary information. A strap must be installed in order to program this memory chip.

### *ISP Module Firmware programming*

All ALTERA® FPGA chips, except for the VMEbus interface chip, are programmed with an In-System Programming scheme, using a "Byte-Blaster" adapter connected to a PC, where the ALTERA MAX-PLUS® programming software is installed.

### *System Clock generation and distribution*

An internal 10 MHz system clock generator is implemented and a clock driver fan-out chip is used to drive the seven impedance matched clock lines, each terminated with a series RC network.

## **STATUS AND DOCUMENTATION**

The design of the ROD-Busy module is finished and two prototypes has been debugged and tested which are now available for evaluation. The prototypes are packaged as 6U/4TE form factor VME modules. A conversion kit for implementation on larger VME boards is also foreseen.

A technical manual has been produced for the ROD Busy Module, which can be retrieved from the CERN EDMS system, together with all the engineering data.[6]

## **CONCLUSION**

The implementation of the ROD-Busy modules and their associated tree structured signal gathering scheme makes it possible to

efficiently control the dead-time in the ATLAS experiment and to easily detect faulty ROD modules introducing excessive dead-time.

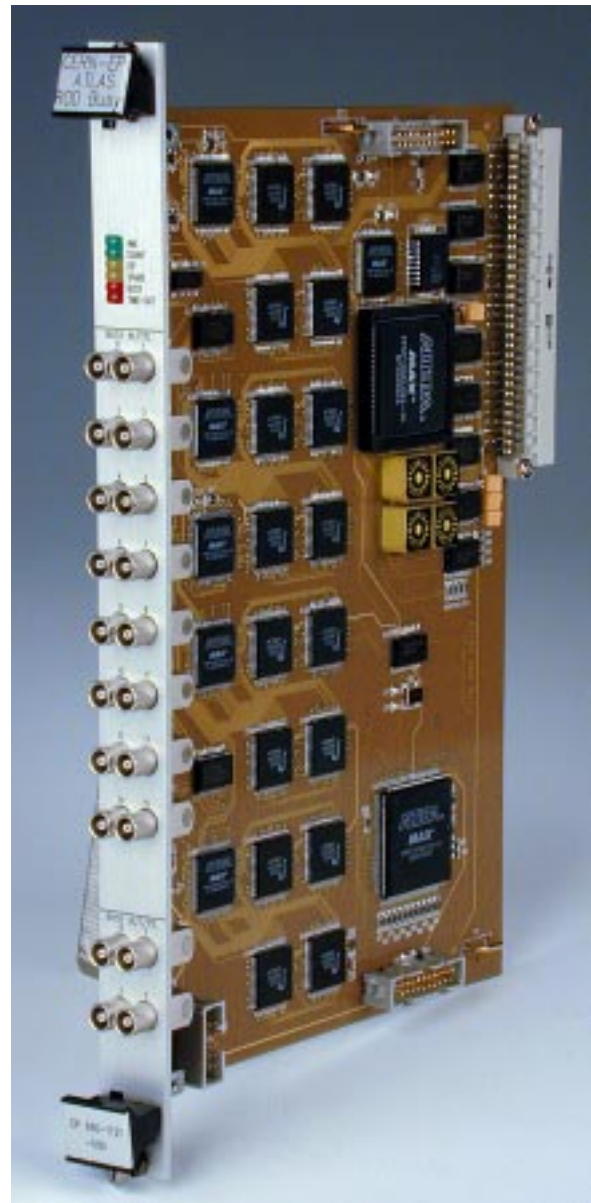


Figure 3. ATLAS ROD Busy Module

## **TTCvi PRODUCTION STATUS**

The CERN EP/ESS Group will in the future handle the out-sourcing of the fabrication of the TTCvi modules and then make them available to users. During the fourth quarter of 2001 will a new batch of modules be produced, which should be available when tested by the EP/ESS group in the beginning of 2002. The EP/ESS group will also be responsible for the service (maintenance, support and spares). Teams who already have TTCvi MkII version modules will be able to subscribe to this service, by paying a yearly fee.

## REFERENCES

- [1] P. Gällnö: "Timing, Trigger and Control Distribution and Dead-Time Control in Atlas", LEB 2000 Workshop, Kraków, Poland, Sept. 2000  
[http://lebworkshop.home.cern.ch/lebworkshop/LEB00\\_Book/daq/gallno.pdf](http://lebworkshop.home.cern.ch/lebworkshop/LEB00_Book/daq/gallno.pdf)
- [2] Ph. Farthouat; "TTC & Dead-time handling" 2<sup>nd</sup> ATLAS ROD Workshop, University of Geneva, Oct. 2000  
[http://dpnc.unige.ch/atlas/rod00/transp/P\\_Farthouat2.pdf](http://dpnc.unige.ch/atlas/rod00/transp/P_Farthouat2.pdf)
- [3] R. Spiwojs: "Dead-time Generation in the Level-1 Central Trigger Processor", ATLAS Internal Note
- [4] ATLAS Level-1 TDR Chapter 20  
<http://atlasinfo.cern.ch/Atlas/GROUPS/DAQTRIG/TDR/tdr.html>
- [5] P. Gällnö: "The ATLAS ROD Busy Module" 1<sup>st</sup> ATLAS ROD Workshop, University of Geneva, Nov. 1998  
<http://mclaren.home.cern.ch/mclaren/atlas/conferences/ROD/programme.htm>
- [6] P. Gällnö: "ATLAS ROD Busy Module - Technical description and users manual", EDMS Item no: CERN-0000003935