TGN #93
Nancy J. Lai and Dave Toback
University of Chicago
July 3rd, 1995
Revised: April 1, 1998

Crate Sum Board Module Specifications

Upgrade for Run II

This document is designed for use by those individuals working on the design and implementation of the Crate Sum (CS) board on the Mentor Graphics system as well as those who debug the boards before, during, and after production. In particular, this document describes the purpose and functionality of all the top-level modules. It also discusses the lower level symbols. This document is NOT designed to stand alone and will make frequent reference to the primary Crate Sum document [1]. The DIRAC and PreFRED documents [2, 3] may also be useful. All signal names and pin ordering for external board connections can be found in these references.

The term "module" is used in this note to designate top-level diagram modules and the term "symbol" is used to designate the use of symbols within the top-level modules, following the terminology of the Mentor Graphics system. "External connection buses" are those buses which link the top level diagram modules. "Internal buses" are buses which are used to link symbols within a module. All bus, symbol, and module names are written in boldface. A list of selected bus signal names can be found at the end of this document.

The board is designed so that an almost complete set of diagnostics (as outlined in this document) can be run using only VME interactions. A test stand which allows front and back panel connections is needed to check connector continuity as well as allow full speed operational testing. The order of presentation of this note has been chosen in the hopes of best illustrating the functionality of each module.

There are five modules in the top-level diagram: **VME_Interface, CONTROL, P3, SUM**, and **FP**. See Figure 1. The **VME_Interface** module is where signals from the VRC [4] and TRACER [5] are read in via the P1 and P2 VME connectors. The **CONTROL** module is designed to interface with the TSI [6], decode VME addressing, distribute control signals (including the front panel LEDs) throughout CS, and control diagnostics. The **P3** module reads in energy and trigger information from DIRAC via the P3 connector through the J3 backplane [7]. The **SUM** module uses this information to perform the CS Level 1 energy and trigger summary through its **ET_SUM** and **TRIG_SUM** submodules, respectively. The energy and trigger summary as well as the original DIRAC E_T and trigger inputs (to Crate Sum) are stored in case of a readout after a L2 accept. The **FP** module controls the CS front panel LEDs and contains the output connectors which send the CS results to PreFRED [3] in the Global Level 1 Crate [8]. The diagnostic tools for CS are controlled by the **CONTROL** module and are implemented by transceivers in the **ET_SUM** and **TRIG_SUM** modules[1].

[1]The **ET_SUM**, **TRIG_SUM**, and **DIAG_DAQ** submodules of the top-level SUM module will henceforth be referred to as "modules" for convenience.

1 The VME_Interface Module

1.1 Purpose

This module reads in signals from the VRC and TRACER via the P1 and P2 VME connectors and processes them. It interfaces between CS and the DAQ during readout, downloading, and diagnostics, and between CS and the TRACER during run-time. All VME and DAQ interactions take place via the VRC and will be referred to collectively as VME interactions. Instructions from the TSI come via the TRACER and will be referred to as TSI interactions. The **VME_Interface** module performs the following functions:

1. Connects the J1 and J2 VME backplanes to the CS-P1 and CS-P2 connectors, respectively. The J0 backplane is not connected to CS via a P0 connector.

2. Routes the Geographical Address bits onto the **GA** bus.

3. Routes the VME Address bits onto the **VME_ADDRESS** bus.

4. Routes the VME Data bits onto the **VME_DATA** bus.

5. Routes the VME Control bits onto the **VME_CONTROL** bus.

6. Routes the TSI Control bits onto the **TRIG_CTRL** bus.

7. Provides simple VME address decoding such as board location, etc.

Note that this module is identical to that used in DIRAC and PreFRED [2, 3].

1.2 External Connection Buses

VME_DATA(31:0), **VME_ADDRESS(26:2)**, **TRIG_CTRL(38:0)**, **VME_CONTROL(7:0)**, **VME_CONFIG(9:0)**, **GA(4:0)**, **P0(94:0)**[2], **P1(159:0)**, and **P2(159:0)**.

2 The CONTROL Module

2.1 Purpose

This module is used to interface with the TSI, decode VME addressing, distribute control signals (including the front panel LEDs) throughout CS, and control diagnostics. The **CONTROL** module performs the following tasks:

1. Provides non-readout run-time control signals for use throughout CS by taking signals from **TRIG_CTRL** to generate latches on **CONTROL(??)**. This occurs in the **HEART** symbol, which will be described below.

[2]Note that the J0 backplane is not connected to CS via a P0 connector.

2. Provides readout control signals for use throughout CS during run-time by taking signals from **VME_CONTROL** and **VME_ADDRESS** to generate **CONTROL(??)**. This occurs in the BRAIN chip, which will be discussed below.
3. Provides VME address decoding and interfaces with all other VME interactions. This occurs within the BRAIN chip.
4. Controls the lights (LEDs) on the front panel. These signals are generated by the **HEART** symbol and the BRAIN chip and are taken directly from **TRIG_CTRL**.
5. Controls a full set of diagnostics on each of the modules within CS by using the signals from the **VME_CONTROL** and **VME_ADDRESS** buses.
6. Provides some error handling by taking in error signals from the **CONTROL** bus and outputting the error status result[3] onto **TRIG_CTRL** to be sent off the board via P2.
7. Provides an 8-bit counter which is implemented and stored in the **BUNCH_COUNTER** symbol.

This module is very similar to the equivalent module on DIRAC, which is mostly implemented in four large PLDs. The PLDs will need to be programmed differently to take into account the differences between the timing requirements and address maps of DIRAC and Crate Sum.

2.2 Components of the CONTROL Module

(This section will have to be updated when the **CONTROL** module is designed. The information here is from the old design, based on the old **CONTROL** module design for DIRAC, and may be obsolete.)

The **CONTROL** module contains the BRAIN, TIMING, and GEOGRAPHICAL ADDRESS chips, as well as the **HEART**, **BUNCH_COUNTER**, and **ERROR** symbols.

2.2.1 HEART

The **HEART** symbol provides the timing (run-time and non-run-time) VME control signals for use throughout CS by reading in control signals from **VME_ADDRESS**, **TRIG_CTRL**, and **VME_CONTROL** and using them to generate latches for use by the **SUM** and **FP** modules. Note that these delay times must be downloaded via interactions with **VME_ADDRESS** and **VME_DATA**. This symbol is implemented via an Altera PLD.

[3]Currently there is only one type of error message. In the future, more than one type of error message may be desired: for example, Fatal and non-Fatal errors, etc. In this case, the signals **CONTROL(??:??)** are logically OR'ed and the result passed out. In the current CS incarnation, there is no room allocated to handle additional error bits sent to the TRACER.

2.2.2 BRAIN

The BRAIN chip is an Altera PLD which performs the VME address decoding, handles the VME interactions, and is the main interface for diagnostics. It takes in signals from the **VME_DATA**, **VME_ADDRESS**, **TRIG_CTRL**, and **VME_CONTROL** buses and outputs its results on the **CONTROL** and **LIGHTS** buses. Furthermore, it performs the error handling for the CS board by taking in the error signal **CONTROL(??)** and outputting it on **TRIG_CTRL(??)**.

2.2.3 TIMING

These chips (timing taps) help generate the timing signals used throughout the Crate Sum board.

2.2.4 BUNCH_COUNTER

This symbol provides the 8-bit counter for CS and is stored in and read out from FIFOs and L2 buffers in a manner similar to that in the **ET_SUM** and **TRIG_SUM** modules.

2.2.5 ERROR

A description of the **ERROR** symbol will be placed here.

2.2.6 GEOGRAPHICAL ADDRESS

These chips decode the signals on the **GEO_ADDRESS** bus. Further details will be provided later.

2.3 External Connection Buses

VME_DATA(31:0), **VME_ADDRESS(26:2)**, **TRIG_CTRL(38:0)**, **VME_CONTROL(7:0)**, **VME_CONFIG(9:0)**, **CONTROL(130:0)**, **LIGHTS(7:0)**, and **GA(4:0)**.
See the "External Buses" section below for a listing of these buses and the signals that are passed.

3 The P3 Module

3.1 Purpose

This module provides the interface between CS and the custom VME J3 backplane [7]. In particular, the **P3** module:

1. Connects the J3 backplane to the CS-P3 connector.

2. Reads in signals from DIRAC via the J3 backplane and CS-P3 and places them on the **ET_RAW** and **TRIG_RAW** buses.

Note that latching of the input energy and trigger information from DIRAC is not done in this module. Rather, it is performed in the **SUM** module, specifically in the **DIAG_DAQ** module.

3.2 External Connection Buses

P3(263:0), **ET_RAW(63:0)**, and **TRIG_RAW(127:0)**.

4 The SUM Module

4.1 Purpose

Each Crate Sum stores and processes the E_T and trigger bit summaries from DIRAC, which receives information for all calorimeter towers from two different ϕ wedges. With four DIRAC boards per wedge, each Crate Sum board receives information from two sets of four DIRACs.

The **SUM** module is implemented by an Altera XX PLD and contains three modules: **ET_SUM**, **TRIG_SUM**, and **DIAG_DAQ**. The **ET_SUM** and **TRIG_SUM** modules are responsible for performing the CS energy and trigger summaries, respectively, as well as the storage and readout for each wedge. The **DIAG_DAQ** module stores the input DIRAC energy and trigger data for possible readout during a L2 accept signal.

4.2 External Connection Buses

ET_RAW(63:0), **TRIG_RAW(127:0)**, **CONTROL(130:0)**, **VME_DATA(31:0)**, **ET_SUM_MUX(9:0)**, and **TRIG_SUM_MUX(9:0)**.

4.3 The ET_SUM Module

4.3.1 Purpose

The **ET_SUM** module provides the CS energy summary, storage, and readout for each wedge during run-time. This module specifically does the following:

1. Latches the eight 8-bit E_T words which make up **ET_RAW(63:0)** on the assertion of **CONTROL(??)**. Note that there is a transceiver which can, via VME interactions and tri-state logic on the latch, read from and write to the output of this latch.

2. Sums the four latched 8-bit DIRAC energies for each of the two ϕ wedges to get one 10-bit result ($\sum^{\mathrm{Wedge}} E_T$). This occurs on the assertion of **CONTROL(??)**.

3. Sends the energy summary for each wedge to the **FP** module in two sequential steps of 66 ns each via **ET_SUM_MUX(9:0)**. This action occurs on the assertion of **CONTROL(??)**. The status bit in **CONTROL(??)** controls which wedge summary is sent out first.

4. Latches both wedge energy sums onto **ET_SUM(19:0)** on **CONTROL(??)**.

5. Latches **ET_SUM(19:0)** into the L1FIFO on **CONTROL(??)** and clocks out the L1FIFO on **CONTROL(??)**.

6. Clocks out the L1FIFO data into the specified L2 buffer within L2BUFF, the five L2 buffer chips, when instructed to by **CONTROL(??:??)**.

7. Reads out the specified L2 buffer onto **VME_DATA(31:0)** on **CONTROL(??)**.

4.3.2 External Connection Buses (of ET_SUM within SUM)

ET_RAW(63:0), **ET_SUM_MUX(9:0)**, **CONTROL(130:0)**, and **VME_DATA(31:0)**.

4.3.3 Internal Connection Buses

ET_SUM(19:0).

4.4 The TRIG_SUM Module

4.4.1 Purpose

This module is functionally very similar to **ET_SUM** and is implemented in very much the same manner and in the same PLD. The **TRIG_SUM** module performs the CS trigger summary, storage, and readout for each wedge during run-time. The L1FIFOs and L2Buffers are implemented in the same way as on **ET_SUM**. This module specifically performs the following functions:

1. Latches the eight 16-bit DIRAC trigger words which make up **TRIG_RAW(127:0)** on the assertion of **CONTROL(??)**. Note that there is transceiver which can, via VME interactions and tri-state logic on the latch, read from and write to the output of this latch.

2. Produces a 20-bit summary from these eight 16-bit DIRAC trigger words. This occurs on the assertion of **CONTROL(??)**.

3. Sends the trigger summary for each wedge to the **FP** module in two sequential steps of 66 ns each via **TRIG_SUM_MUX(9:0)**. This occurs on the assertion of **CONTROL(??)**. The 20-bit trigger summary consists of five 2-bit numbers and ten 1-bit numbers. The five 2-bit trigger result is sent in the first 66 ns stage, followed by the ten 1-bit numbers.

4. Latches the 20-bit trigger summary onto **TRIG_SUM(19:0)**.

5. Latches **TRIG_SUM(19:0)** into the L1FIFO on **CONTROL(??)** and clocks out the L1FIFO on **CONTROL(??)**.

6. Clocks out the L1FIFO data into the specified L2 buffer within L2BUFF on **CONTROL(??:??)**.

7. Reads out the specified L2 buffer onto **VME_DATA(31:0)** on the assertion of **CONTROL(??:??)**.

4.4.2 External Connection Buses (of TRIG_SUM within SUM)

TRIG_RAW(127:0), **TRIG_SUM_MUX(9:0)**, **CONTROL(130:0)**, and **VME_DATA(31:0)**.

4.4.3 Internal Connection Buses

TRIG_SUM(19:0).

4.5 The DIAG_DAQ Module

4.5.1 Purpose

This module stores the input DIRAC energy and trigger data and reads it out following a L2 accept during a trigger diagnostic readout. In particular, it:

1. Latches **ET_RAW(63:0)** and **TRIG_RAW(127:0)** into the L1FIFO on **CONTROL(??)** and clocks out the L1FIFO on **CONTROL(??)**.

2. Clocks out the L1FIFO data into the specified L2 buffer within L2BUFF on **CONTROL(??:??)**.

3. Reads out the specified L2 buffer onto **VME_DATA(31:0)** on the assertion of **CONTROL(??:??)**.

4.5.2 External Connection Buses (of DIAG_DAQ within SUM)

ET_RAW(63:0), **TRIG_RAW(127:0)**, **CONTROL(130:0)**, and **VME_DATA(31:0)**.

5 The FP Module

5.1 Purpose

This module reads the 10-bit multiplexed half-crate energy and trigger summary signals from **ET_SUM_MUX(9:0)** and **TRIG_SUM_MUX(9:0)**, respectively, and sends them differentially to the Global Level 1 Crate via front panel connectors. It also contains the front panel LED display. This module specifically does the following:

1. In two sequential 66 ns steps, takes half of the energy summary result from **ET_SUM_MUX(9:0)**, uses differential drivers to convert it to differential signals, and outputs it via CS-FR1. See Reference [1] for further details.

2. In two sequential 66 ns steps, takes half of the trigger summary result from **TRIG_SUM_MUX(9:0)**, uses differential drivers to convert it to differential signals, and outputs it via CS-FR2. The five double-bit trigger bits are sent in the first 66 ns and the ten single-bit trigger bits in the subsequent 66 ns. See Reference [1] for further details.

3. Takes in signals from **LIGHTS** to control the front panel LEDs.

5.2 External Connection Buses

LIGHTS(7:0), **ET_SUM_MUX(9:0)**, **TRIG_SUM_MUX(9:0)**, **CS_OUT(19:0)**, and **CONTROL(130:0)**.

6 Diagnostics

Most diagnostics are run using VME interactions. VME control signals **VME_CONTROL(??:??)** are asserted to tell CS that a VME address (**VME_ADDRESS(??:??)**) and data **VME_DATA(??:??)** will be sent. The **CONTROL** module uses these signals to send signals along the **CONTROL** bus. For simplicity, "VME interactions" will be taken to refer to those actions which clock, write data to buses, etc., and the VME mechanics behind these interactions will be ignored. Some anticipated standard diagnostics are listed below.

- To test DIRAC outputs: Read out the **ET_RAW(63:0)** and **TRIG_RAW(127:0)** information stored in **DIAG_DAQ** via VME. Compare to what is expected.

- To test the energy and trigger summers: Use DIRAC or VME to write to the input latches. Read out via VME to test whether the result matches what was written in. Repeat for *many* patterns.

- To test L1FIFO and L2BUFF: Use DIRAC or VME to provide inputs to L1FIFO (which are ordinarily the outputs of the summer). Read out via VME to test whether the result matches what was written in. Clock in the L1FIFO. Clock out the L1FIFO into a L2 buffer. Read out the result over VME and compare to what is expected. Repeat for all four L2 buffers. Repeat, but clock in many different events and clock them out, but only put some of them into the L2 buffer. Repeat again, but perform a reset in the middle.

- To test CS outputs: Use DIRAC or VME to provide outputs of SUMMER. Clock front panel drivers. Read out front panel connectors using the test stand and compare.

- To test CS at run-time speeds: Input data using DIRAC (teststand conditions), clock out using TRACER at full speed, read out front panel connectors using the test stand, and compare.

7 External Buses (between top-level modules)

GA(4:0)

Signal Name	Description
VME_ADDRESS(4:0)	Geographical Addresses

Table 1: Used only by the **CONTROL** module.

VME_ADDRESS(26:2)

Signal Name	Description
VME_ADDRESS(26:2)	Internal Addresses

Table 2: Used by the VME_Interface section for DAQ interactions around the CS board.

VME_DATA(31:0)

Signal Name	Description
VME_DATA(31:0)	Data

Table 3: Used by the VME_Interface section for DAQ interactions around the CS board.

VME_CONFIG(9:0)

Signal Name	Description
VME_CONFIG(9:0)	?

Table 4: The purpose of this bus will be stated here later.

VME_CONTROL(7:0)

Signal Name	Description
VME_CONTROL(0)	MODSEL*
VME_CONTROL(1)	DS*
VME_CONTROL(2)	D0EN*
VME_CONTROL(3)	D1EN*
VME_CONTROL(4)	ACK*
VME_CONTROL(5)	ERR*
VME_CONTROL(6)	WRITE
VME_CONTROL(7)	SYSCLK

Table 5: These lines between the **VME_Interface** and **CONTROL** modules to pass control signals from VME via the VRC.

TRIG_CTRL(38:0)

Signal Name	Description
TRIG_CTRL(0)	CLK
TRIG_CTRL(1)	CLK*
TRIG_CTRL(3)	B0*
TRIG_CTRL(4)	L1A*
TRIG_CTRL(5)	L1R*
TRIG_CTRL(6)	L2B0*
TRIG_CTRL(7)	L2B1*
TRIG_CTRL(8)	HALT*
TRIG_CTRL(9)	RECOVER*
TRIG_CTRL(14)	ERROR*

Table 6: These lines between the **VME_Interface** and **CONTROL** modules to pass control signals from the TSI via the TRACER. Only the signals used by CS are listed above.

CONTROL(130:0)

Signal Name	Description
CONTROL()	Latch DIRAC In
CONTROL()	DIRAC LATCH Enable
CONTROL()	VME LATCH latch in (Wedge0)
CONTROL()	VME LATCH latch in (Wedge1)
CONTROL()	VME LATCH latch in (Trigger)
CONTROL()	VME LATCH Enable
CONTROL()	SUMMER shift in
CONTROL()	SUMMER enable
CONTROL()	VME SUMMER latch (Wedge0)
CONTROL()	VME SUMMER latch (Wedge1)
CONTROL()	VME SUMMER latch (Trigger)
CONTROL()	VME SUMMER enable
CONTROL()	L1FIFO Shift in (Heart Beat)
CONTROL()	L1FIFO Shift out (Decision)
CONTROL()	FIFO Master Reset
CONTROL()	L1FIFO empty*
CONTROL()	Counter FIFO empty*
CONTROL()	L2 Buffer Write
CONTROL()	L1BA0
CONTROL()	L1BA1
CONTROL()	L2BA0
CONTROL()	L2BA1
CONTROL()	Front Panel Output Enable (ET)
CONTROL()	Front Panel Output Enable (TRIG)

Table 7: Bus signals used to distribute commands throughout the CS board. This table may not be accurate and will be updated when the **CONTROL** module design is completed.

LIGHTS(7:0)

Signal Name	Description
LIGHTS(0)	Run
LIGHTS(1)	Load
LIGHTS(2)	Error
LIGHTS(3)	VME_Select
LIGHTS(4)	Halt*
LIGHTS(5)	Clock
LIGHTS(6)	L1_Decision
LIGHTS(7)	CS or $\overline{\text{CS}}$

Table 8: Bus signals used to control the front panel LEDs in the **FP** module. Note that "CS or $\overline{\text{CS}}$" indicates whether the board outputs the even-wedge or odd-wedge energy summary result first.

ET_RAW(63:0) AND TRIG_RAW(127:0)

Signal Name	Description
ET_RAW(0)	Least Significant Bit of E_T: DIRAC 0
ET_RAW(1)	Next-to-Least Significant Bit of E_T: DIRAC 0
⋮	⋮
ET_RAW(6)	Next-to-Most Significant Bit of E_T: DIRAC 0
ET_RAW(7)	Most Significant Bit of E_T: DIRAC 0
ET_RAW(8)	Least Significant Bit of E_T: DIRAC 1
ET_RAW(9)	Next-to-Least Significant Bit of E_T: DIRAC 1
⋮	⋮
ET_RAW(54)	Next-to-Most Significant Bit of E_T: DIRAC 6
ET_RAW(55)	Most Significant Bit of E_T: DIRAC 6
ET_RAW(56)	Least Significant Bit of E_T: DIRAC 7
ET_RAW(57)	Next-to-Least Significant Bit of E_T: DIRAC 7
⋮	⋮
ET_RAW(62)	Next-to-Most Significant Bit of E_T: DIRAC 7
ET_RAW(63)	Most Significant Bit of E_T: DIRAC 7
TRIG_RAW(0)	Trigger bit 0 - DIRAC 0
TRIG_RAW(1)	Trigger bit 1 - DIRAC 0
⋮	⋮
TRIG_RAW(14)	Trigger bit 14 - DIRAC 0
TRIG_RAW(15)	Trigger bit 15 - DIRAC 0
⋮	⋮
⋮	⋮
TRIG_RAW(112)	Trigger bit 0 - DIRAC 7
TRIG_RAW(113)	Trigger bit 1 - DIRAC 7
⋮	⋮
TRIG_RAW(126)	Trigger bit 14 - DIRAC 7
TRIG_RAW(127)	Trigger bit 15 - DIRAC 7

Table 9: These signals are used to send the 8 DIRAC energy and trigger information to the **ET_SUM** and **TRIG_SUM** modules in **SUM**.

ET_SUM_MUX(9:0) AND TRIG_SUM_MUX(9:0)

Signal Name	Description
ET_SUM_MUX(9)	Most Significant Bit of $\overset{\text{Wedge}}{\Sigma}\mathrm{E_T}$
ET_SUM_MUX(8)	Next-to-Most Significant Bit of $\overset{\text{Wedge}}{\Sigma}\mathrm{E_T}$
⋮	⋮
ET_SUM_MUX(1)	Next-to-Least Significant Bit of $\overset{\text{Wedge}}{\Sigma}\mathrm{E_T}$
ET_SUM_MUX(0)	Least Significant Bit of $\overset{\text{Wedge}}{\Sigma}\mathrm{E_T}$
TRIG_SUM_MUX(0)	Trigger Bit 0 (Step 1) or Trigger Bit 10 (Step 2)
TRIG_SUM_MUX(1)	Trigger Bit 1 (Step 1) or Trigger Bit 11 (Step 2)
TRIG_SUM_MUX(2)	Trigger Bit 2 (Step 1) or Trigger Bit 12 (Step 2)
TRIG_SUM_MUX(3)	Trigger Bit 3 (Step 1) or Trigger Bit 13 (Step 2)
TRIG_SUM_MUX(4)	Trigger Bit 4 (Step 1) or Trigger Bit 14 (Step 2)
TRIG_SUM_MUX(5)	Trigger Bit 5 (Step 1) or Trigger Bit 15 (Step 2)
TRIG_SUM_MUX(6)	Trigger Bit 6 (Step 1) or Trigger Bit 16 (Step 2)
TRIG_SUM_MUX(7)	Trigger Bit 7 (Step 1) or Trigger Bit 17 (Step 2)
TRIG_SUM_MUX(8)	Trigger Bit 8 (Step 1) or Trigger Bit 18 (Step 2)
TRIG_SUM_MUX(9)	Trigger Bit 9 (Step 1) or Trigger Bit 19 (Step 2)

Table 10: These signals are used to transfer the energy and trigger bit summaries from the **ET_SUM** and **TRIG_SUM** modules to the **FP** module. Note that $\overset{\text{Wedge}}{\Sigma}\mathrm{E_T}$ is for either the odd or even wedge, depending on the particular type of Crate Sum board. The steps refer to the two 66 ns sequential steps in which the energy and trigger summaries are sent out.

References

[1] N. Lai and D. Toback, *Crate Sum: Upgrade for Run II*, CDF Note (?), (?) 1998.

[2] J. Wahl, *DIRAC - The FINAL Specification*, CDF Note 4225, June 1997.

[3] A. Amadon, M. Bogdan, and P. Wilson, *Specifications and Design of the Level 1 Calorimeter Trigger PreFRED Modules for the Run II Upgrade*, CDF Note 4340, Dec. 1997.

[4] VRC is the abbreviation for the **V**MEbus **R**eadout **C**ontroller.

[5] T. Shaw, *Specification for TRigger And Clock + Event Readout Module (TRACER)*, CDF Note (?), Aug. 1997.

[6] TSI is the acronym for the **T**rigger **S**upervisor **I**nterface.

[7] P. Wilson, *J3 Backplane and Transition Module (L1AUX) for the Level 1 Calorimeter Trigger Upgrade*, Trigger Note #92, Dec. 1997.

[8] A. Amadon *et al.*, *Conceptual Design of the Global L1 Trigger Decision Crate for the Run II Upgrade*, CDF Note 4339, (?) 1998.

Figure 1: Crate Sum Top-Level Diagram