June 13, 1995

Trigger ⇨ Front-End Electronics Interface Specification
Trigger/Clock Distribution Tree
Version 1.0a

V. Lindenstruth, P. Barale, R. Bellwied, F. Bieser, B. Bonner, M. Botlo, J. Cannon, H. Crawford, L. Greiner, J. Hoffmann, R. Jared, S. Klein, M. LeVine, T. Noggle, J. Schambach, D. Underwood

Sign-Off Page

The STAR management and group leaders agree to the definition of the interface between the STAR trigger system and the front-end systems as defined in this document. Further changes to this trigger/clock distribution tree will require the written agreement of all involved parties.

Jay Marx

Richard Jared

Henry Crawford

Rene Bellwied

Fred Bieser

Spencer Klein

Billy Bonner

Dave Underwood

Daniel Cebra

Change Log

??, 95	First version accepted

1 Introduction

This document describes the interfaces between the trigger system and the front-end electronics systems of the various detectors of STAR. The most important interface is the trigger and clock distribution tree. This tree delivers the actual triggers and clears to all front-end systems. It also distributes clocks such as the RHIC clock.

2 Requirements

2.1 Clock Distribution

Requirement:

Distribute the RHIC strobe as the central heart beat to all detector systems. Allow to compensate for different propagation delays up to 100 ns in steps of 1 ns. The timing jitter with respect to the RHIC strobe input to the trigger distribution tree is limited to 1ns RMS.

Justification:

The RHIC strobe (110 ns clock period) defines the interaction time and is kind of the heart beat of the whole detector. It will be used as timing reference for any state transitions in the detector system.

There are three parameters relevant for the trigger/clock distribution tree:

1) phase: average phase difference due to different propagation delays and cable lengths at the end points of the trigger/clock distribution tree, the readout boards.
2) coherent jitter: the phase of all trigger/clock distribution receivers varies in phase with respect to the RHIC strobe on an event-by-event basis.
3) incoherent jitter: the phase of all trigger /clock distribution receivers varies with respect to each other within an event. This number defines the timing resolution possible with signals derived from the trigger/clock distribution tree.

The maximum acceptable phase and jitter is defined by the detector requirements:

	phase	**coherent jitter**	**incoherent jitter**
TPC:	5 ns	5 ns (RMS)	1 ns (RMS)
SVT:	1 ns	1 ns (RMS)	1 ns (RMS)
EMC:	1 ns	1 ns (RMS)	1 ns (RMS)
ToF:	1 ns	1 ns (RMS)	50 ps (RMS)
xTPC:	5 ns	5 ns (RMS)	5 ns (RMS)
TRIG:	1 ns	1 ns (RMS)	25 ps (RMS)

Note:

The trigger/clock distribution tree logic will support both multi drop and point to point cabling scenarios. However in the case of a multi drop scenario it may not be possible for example due to cable delays to adjust the relative phase of the various trigger/clock receivers down to 1 ns.

2.2 Trigger Command Word

Requirement:

Distribute a global 4 bit trigger command word as defined below.

Justification:

Each readout board needs to know if for a given RHIC strobe certain actions are required.

Generally the number of bits needed to be communicated per trigger or abort should be kept as small as possible because it corresponds to either transmission speed or the number of signals of the trigger distribution cable. Both affect the cost of the system. Sixteen different trigger commands are sufficient. This allows to encode them in a 4 bit binary word. To define a good timing reference these trigger command words must be sampled at the rising edge of the RHIC clock.

Note:

There are three classes of Trigger Command Words:

① Commands that are not readout related

② Commands that are readout related and require an event to be produced in the front-end

③ Commands that are readout related and accompanied by a valid trigger token but that do not result in any readout.

The first group are commands that have to be distributed synchronously to the RHIC strobe and are not related to readout. These commands are broadcast to the detectors. The trigger token and DAQ command word that will be distributed together with these trigger command words will be forced to zero. These commands will be distributed independent of the busy status of the detector.

The second group requires Trigger to produce and reserve a unique trigger token different from zero that is distributed together with the Trigger Command. Depending on the Trigger Command it also has to inform DAQ about the event as specified in the Trigger-DAQ interface.

Group ① Trigger Commands:

no-trigger 0: This is the default symbol that will be transferred in the case of an idle link.

clear 1: This command clears redundancy counters on the readout boards.

The readout boards will have redundant counters counting the RHIC strobe. The value of these counters will be shipped together with the raw data allowing cross checks that each system received exactly the same number of clock pulses.

master-reset 2: This command will perform a general reset of the whole front-end logic.

spare 3: reserved

<u>Group ② Trigger Commands:</u>

Note: EMC means in the following tables the Electromagnetic Calorimeter and the Shower-Max detector

trigger$_n$ 4+n: There are four trigger commands available allowing four different kinds of readout in the front-end.

trigger0 4: This is the default physics readout and has to be supported by all detector systems.

TPC: physics readout
SVT: physics readout
EMC: Electron Trigger (EMC/TPC coincidence)
TOF: physics readout
XTPC: complete physics readout

trigger1 5:

TPC: reserved - executed as trigger0
SVT: reserved - executed as trigger0
EMC: Isospin readout (assumes EMC/TOF coincidence)
TOF: reserved - executed as trigger0
XTPC: reserved - executed as trigger0

trigger2 6:

TPC: reserved - executed as trigger0
SVT: reserved - executed as trigger0
EMC: reserved - executed as trigger0
TOF: reserved - executed as trigger0
XTPC: reserved - executed as trigger0

trigger3 7:

TPC: executed as trigger0 but copy event also into local buffer
SVT: executed as trigger0 but copy event also into local buffer
EMC: reserved - executed as trigger0
TOF: reserved - executed as trigger0
XTPC: reserved - executed as trigger0

Group ② Trigger Commands (continued):

pulser$_n$ 8+n: There are four different test strobe commands available that will be used to fire test pulses or to fire a laser for calibration and monitoring purposes.

Note: The test pulse command produced by Trigger may be intercepted by the trigger/clock Driver module and result in a sequence of test pulse commands. This test pulse sequence will be protected by the detector dead time signal which will be asserted by the trigger/clock Driver module while performing that sequence. The trigger/clock Receiver modules can easily disendtangle a pulser command that shall just fire the specified pulser or laser and the final test pulse command that requires also the read-out since only the latter will be accompanied by a non zero trigger token.

pulser0 8:

TPC: electrical pulser
SVT: electrical pulser - calibration pulser
EMC: electrical pulser - charge injection to PMT and ShowerMax
TOF: reserved - ignored
XTPC: electrical pulser

pulser1 9:

TPC: laser
SVT: electronics test pulser
EMC: LED/laser trigger
TOF: laser
XTPC: laser

pulser2 10:

TPC: reserved - ignored
SVT: reserved - ignored
EMC: radioactive source
TOF: reserved - ignored
XTPC: reserved - ignored

pulser3 11:

TPC: buffered readout - reproduce event stored by trigger3 command
SVT: buffered readout - reproduce event stored by trigger3 command
EMC: reserved - ignored
TOF: reserved - ignored
XTPC: reserved - ignored

config 12: This is a house keeping trigger. It has to be supported by all detector systems and returns geographic information of the front-end boards allowing automatic configuration of the experiment.

Group ③ Trigger Commands:

abort 13: This command aborts and clears an active event. Which event must be cleared is defined with the accompanying unique trigger token.

L1accept 14: This command informs the front-end that there was a level one (L1) accept of the event identified by the trigger token. it is usefull for detectors that perform level two (L2) analysis on their data. The L1 accept would define the event to be available for this L2 analysis preventing the L2 processors from wasting time on events that will be cleared by the trigger L1 logic.

L1accept commands are allways sent in the same order as the corresponding triggers were issued.

L2accept 15: This command informs the front-end that the event identified with the trigger token was accepted by the trigger system and that DAQ was informed about the existence of that event. Fast detectors like EMC and TOF use this command to start the transmission of the event to DAQ. After receipt of a L2accept command aborts for that event are impossible.

L2accept commands may be issued in any order.

2.3 Trigger Token

Requirement:

Generate a unique 12-bit trigger token for every trigger and distribute it with each trigger, accept and abort command.

Justification:

The STAR detector system will consist of many independent detector systems, which will have their individual dead time. Any combination of detectors is possible for coincident triggers. The readout time of the various systems varies greatly. Therefore it is possible that during the conversion of one detector another detector is already live and ready to take more data. The STAR trigger system supports to fire triggers to fast detectors even if their previously coincident event with a slow detector is still being processed. Therefore it is very important to have a mechanism that allows to uniquely identify to which event and detector any data buffer belongs. In order to uniquely identify a data buffer with a given event the trigger system will distribute a unique trigger token together with the trigger command itself. The L1/L2accept and abort commands identify the event with the trigger token as well.

One way to make the trigger token unique is to use enough bits to make sure that the same trigger token will never be used again. For example a 64 bit bunch crossing number would be a good example. However cost considerations require the number of bits transferred per trigger to be as small as possible. Another approach is to make sure that the same trigger token is not used again until the event is assembled. In this scenario the number of bits required for the trigger token is defined by the maximum number of outstanding triggers that are allowed by the data acquisition system. An upper limit for this number is defined by the highest accepted DAQ trigger rate (1000/sec) and the maximum DAQ latency (2s). Correspondingly a 12 bit trigger token would be sufficient. The trigger system will reuse a trigger token only if the corresponding event was readout and assembled by DAQ and the appropriate trigger structures were cleared by DAQ in the Trigger-DAQ interface (refer to the Trigger-DAQ interface specification).

2.4 DAQ Command Word

Requirement:

Distribute for every trigger a global 4 bit DAQ command word to all DAQ front end systems of all coincident detectors of a given event.

Justification:

There are event attributes that require the same handling on the front-end electronics level but different handling of the raw data on the DAQ front-end level. For example normal events and baseline events or scale-down events, that have to bypass the L3 filter, would all require exactly the same handling on the readout boards. However their handling on the receiver boards would be significantly different. It would be of course possible to fetch this information that is available in the Trigger-DAQ interface and distribute it to all DAQ front-end systems for each event. But considering that there will be 144 receiver boards just for the TPC this seems to be a clumsy approach. A much cleaner approach would be to distribute a descriptive bit pattern to DAQ together with the data that allows to disentangle different handling procedures in the front-end.

Four bits or 16 different commands are sufficient. The same DAQ command word is distributed to all coincident detectors.

Note:

The DAQ Command Word will be produced together with the Trigger Command word in the trigger action word lookup table within the TCU.

2.5 Maximum Trigger distribution dead time

Requirement:

The maximum time to distribute a complete trigger command is 5 RHIC strobes.

Justification:

The time required to distribute a trigger or clear command affects the dead time and correspondingly the detector performance. It defines also the minimum time between two successive trigger pulse commands. There detector system in STAR that requires triggers or test strobes faster than 5 RHICH strobes back-to-back

Note:

Using the implementation as proposed in chapter 3 there would not be any dead time associated with the trigger/clock distribution. It would be possible to distribute the whole 20 bit information within one bunch crossing.

2.6 Multi Drop

Requirement:

Support Multi Drop - allow powering down one trigger/clock receiver in a multi drop chain without interfering with the rest of the tree.

Justification:

Some detectors (TPC) will implement the trigger/clock distribution as a multi drop bus. This saves a lot of cable and cable space. However it may be required to power down an individual readout board. This condition must not interfere with the trigger and clock distribution of the other still running readout boards which are connected to the same branch.

2.7 Physical Signal Distribution

Requirement:

The physical trigger/clock signals are:

- differential, copper
- maximum cable length 30 m
- maximum ground potential difference (common mode suppression) 1 V
- highest frequency distributed: five times RHICH clock

Justification:

Given the number of trigger/clock distribution tree end nodes of about 1000 and given the moderate distance and clock rate a fiber optics based system could not be justified. The AC requirements need to be specified to allow the design of the trigger/clock distribution tree.

2.8 Accept front-end electronics busy and DAQ busy

Requirement:

Accept one individual busy signal from each detectors front-end electronics and one corresponding DAQ front-end busy (no common dead time).

Support optional BUSY return from the end nodes of the Trigger/Clock distribution tree.

Justification:

STAR does not have a common dead time. There is dead time created by the front-end electronics during digitization of an event or while clearing an event. Another source of dead time is the DAQ front-end if all available elasticity buffers are allocated. The trigger control unit will use the logical OR of these signals of a given detector as the detector busy input and if asserted prohibit any further triggers to the given detector.

Some detectors like EMC may perform data compression at the readout module level. Therefore there may be a need for producing a BUSY signal to protect buffers from overflowing. All BUSY signals of the readout modules will be logically ORed in the Trigger/Clock distribution tree. In this scenario a slow controls interface is required to allow checking of the BUSY status of every board for maintenance.

2.9 Readout card buffer space requirement

Requirement:

Each detector that needs to run at higher rates than the TPC (100/sec; 10 ms digitization time) is required to buffer its TPC related data. Fast detectors like EMC and TOF may send their data only after a L2 accept to DAQ.

Justification:

This requirement allows the trigger system to issue new triggers to fast detectors before their TPC related data was completely read out. With that rule it is even possible for example to clear a TPC event including EMC data after another non coincident EMC event was completely read out.

The interface between the front-end and DAQ becomes very simple since an event will never be cleared after it was completely transmitted.

2.10 Level zero Trigger requests

Requirement:

Accept level zero (L0) trigger requests of other fast detectors and level sensitive special trigger requests. The latency between a L0 trigger request and the issue or reject of the given request has to be deterministic.

Justification:

Detectors like EMC and TOF produce their own L0 trigger information.

Special or maintenance triggers such as laser or pulser triggers have to be implemented so that they can be requested by a given detector.

2.11 Participation of *non trigger detectors* in L2

Requirement:

Accept data of detector systems other than the trigger detectors for inclusion into the L2 event selection.

Justification:

External detectors that requested for example a L0 trigger need to be able to abort that event. The trigger system is the only instance that can issue aborts. Consequently these aborts have to be requested using a standardized interface.

3 Implementation

3.1 L0/FPGA tree interface

The L0 event selection will be mainly based on the data of the trigger detectors. The raw data of the trigger detectors will be filtered with a tree of field programmable gate arrays. These FPGAs will be implemented in a way that allows to reconfigure them on a run by run basis. Correspondingly the L0 trigger selection algorithm will be extremely flexible. There will be about 1µs allowed (this number may be variable in units of bunch crossings but fixed for any run) for the signal propagation through the FPGA tree until the trigger is issued or rejected by the TCU. In order to allow other detector systems to contribute to the L0 trigger decision the same setup can be used. The trigger crew will provide the design of the FPGA boards. The actual programming of the FPGA tree is however the responsibility of the appropriate detector crew.

The input to the detector FPGA tree will also include the detector busy signals. This allows for example to produce the front-end busy that will be deterministic for most of the detectors in the FPGA tree. For example the TPC and SVT front-end busy is well defined to be 10.1 ms after each trigger and 1 µs after each clear.

3.2 Participation of *non trigger detectors* in L2

The L2 interface between *external* detectors like EMC and TOF and the trigger system will be implemented as a shared memory. The external detectors will leave their summary data at this memory for consideration by the L2 trigger CPU. The physical implementation of that shared memory is not defined yet.

Detectors requesting that functionality have to supply physically meaningful summary information to the trigger system. The maximum time allowed to produce this summary data is constant per run. This will be implemented using a shared memory where the summary data is deposited by the detectors and fetched by the trigger L2 CPUs.

3.3 Trigger/Clock distribution tree

Figure 1 shows a sketch of the trigger/clock distribution tree as it is currently envisioned. There will be root driver modules for each individual detector. All root modules will sit in a VME crate and receive the 96 trigger bits and RHIC strobe at the back plane. Each trigger root module will have a detector specific mezzanine board that allows to customize the detector specific clock signals that must be distributed.

The trigger distribution tree may be implemented in two different ways: point-to-point with length adjusted cables and multidrop. For example in the case of the TPC a multi drop cable per sector may be sufficient. However this imposes that there may be a clock phase difference between the readout boards of a sector of as much as 5ns. Detectors that need the clock phase to be determined better than that may have to use point-to-point links as indicated in Figure 1 for the case of the SVT.

There may be an optional second layer of boards, simple repeaters, that drive the cables to the individual readout boards. In case of the TPC for example there could be two of these repeaters on each end cap driving 12 individual sector cables. Any delay adjustments on these boards would have to be implemented with jumpers. There will be no way to configure these boards by software. The utility of these repeater boards will be determined by a trade-off between more cables and more logic. The disadvantage of these boards is that they would need extra power supplied from somewhere depending on their physical position on the detector. Currently it is not clear if these boards will be required at all. However the use of these repeaters is a pure implementation issue.

The number of trigger/clock distribution cables per detector and end cap of STAR is:
(Note: the readout granularity of EMC and ShowerMax is not finalized yet; the numbers quoted are estimates for the detector+endcap)

TPC:	24	(one cable per inner and outer sector using multi drop)
SVT:	12	(one cable per readout board)
EMC:	60+24	
ShowerMax:	60	
ToF:	60	(one cable per Receiver/Control Module)
xTPC:		currently not defined
TRIG:	0	(all trigger detectors digitize each bunch crossing

Note: Requirement 2.1 states for the maximum incoherent jitter of 50 ps RMS fort the clock distribution of the TOF system. This requirement is a lot harder than all other requirements taking into account that the Trigger detectors run in lock step with the RHIC clock and do not need distribution of any trigger or abort signals. The TOF time to pulse height converter stop signals requiring that low jitter will have to be implemented by the TOF crew and are considered outside the scope of the trigger/clock distribution tree.

However all TOF front-end boards need to know of any triggers involving TOF. Consequently there is still a requirement for a trigger/clock distribution to the TOF front-end systems.

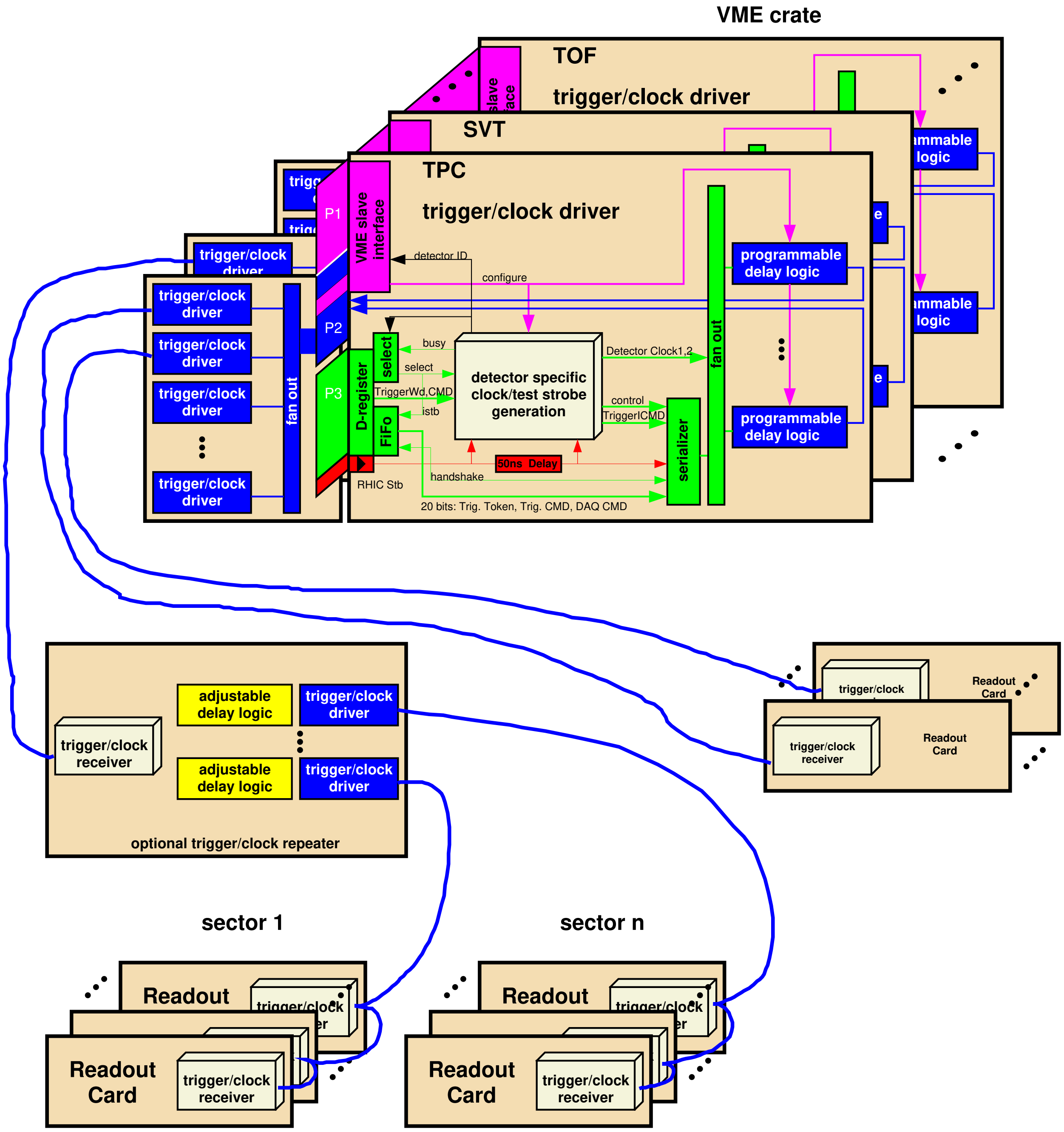

Figure 1: The major components of the trigger/clock distribution tree.

It is planned to distribute additional detector specific clock signals. The system design could be very simple if these clocks would be produced at one well defined place and distributed to the front-end through the trigger clock distribution tree. Two independent detector specific clock signals are sufficient. The detector crews requested the following:

detector	**clock signal 1**	**clock signal 2**
TPC:	SCA clock #1 (~10 MHz variable)	SCA clock #2 (~10 MHz variable)
SVT:	5•RHICclk (SCA clock)	3/4•RHICclk (ADC clock)
EMC:	4...5•RHICclk	spare
ToF:	4•RHICclk	spare
xTPC:	not defined	not defined

Note: The various requirements were only listed here in order to sustain the maximum clock rates stated in 2.7. This number defines the requirement for the transceivers of the trigger/clock distribution tree. The actual clock rate distributed through the trigger/clock distribution tree does not matter at all as long as it does not exceed this maximum. The table above indicates that roughly 50MHz are sufficient for all STAR detector systems.

3.3.1 Trigger/Clock receiver module

Since the trigger/clock receiver interfaces to the front-end electronics boards of the STAR detectors, it needs to be defined very well allowing the designs of all theses systems to proceed.

The stated electronic requirements of the distribution tree are met by the specifications of the LVDS (IEEE 1596.3) transceiver chips. Low Voltage Differential Signals (LVDS) are low power, low voltage swing high speed current signals. There are TTL⇨LVDS and LVDS⇨TTL chips available from National Semiconductor as quad receivers/transmitters (DS90C032/DS90C031). Their price is $4.50 per chip at quantity 100 and they support switching rates in excess of 77 MHz.

The trigger/clock distribution tree will have 10 signal pairs as follows:

1	RHIC strobe
2...5	4 bit data word
6	data clock (2.5•RHICclk; double edge transmission)
7	detector clock 1
8	detector clock 2
9	optional BUSY
10	optional STATUS

There are five 4-bit entities that have to be distributed for each trigger (trigger CMD, DAQ CMD, Trigger token (12 bits)). Running the 4-bit word clock at 5 times the RHIC clock rate which is well within the capabilities of the distribution tree it is possible to distribute the full trigger information within one RHIC strobe. Consequently the trigger/clock distribution tree would be dead time free. The two last optional signals (BUSY and STATUS) are returned as wired OR to the Trigger/Clock driver Module.

The actual 4 bit word sequence within one RHIC strobe period is sketched in Figure 2. The corresponding signals are underlined in light grey. The rising edge (dark) of the RHIC strobe de-

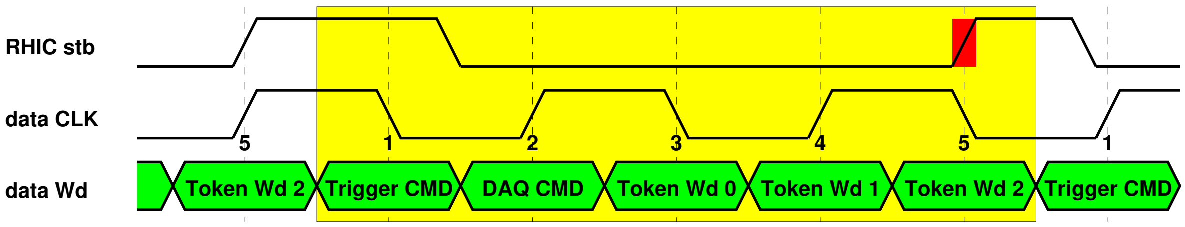

Figure 2: The word sequence within one RHIC strobe period.

fines the timing reference. The Trigger command is the most important word and needs to be available at the trigger time (rising edge of RHIC strobe). Consequently it is distributed first. The other words follow and are buffered in the trigger/clock receiver.

The actual logic of the trigger/clock receiver can be implemented in a single FPGA. The Intel/Altera Flexcore iFX740/EPX740 is especially suitable because it has high speed SRAM built in which can be used to buffer the trigger data word stream and guarantees a maximum propagation delay from any input to any output (excluding feed backs) of 10ns independent of the actual logic inside. This device is available at a price of about $20 for quantity 100.

Figure 3 shows a sketch how this will be implemented. The FPGA would essentially implement a FiFo to buffer the TriggerCMD, DAQ CMD and Trigger Token. This FiFo will have an 8 bit output allowing the data to be read out exactly in the header format as required by the fiber interface specification. The FiFo output has an individual output enable to allow the receiver module to be connected to a global data bus. The Trigger CMD will be always driven and guaranteed to have a stable value at the rising edge of the RHIC strobe.

The EPX740 is available in a 44 pin PLCC package. In this package it has 30 usable I/O signals. Referring to Figure 3 there are 8 inputs needed to read the outputs of the quad receivers. The interface to the readout board requires 14 I/O signals. This leaves 8 general purpose I/O signals available for special outputs such as divided clocks etc. The pin assignment is left open to accommodate any layout requirements.

Due to the small size of all components the size of the trigger/clock receiver logic does not exceed 2 square inches including the connector.

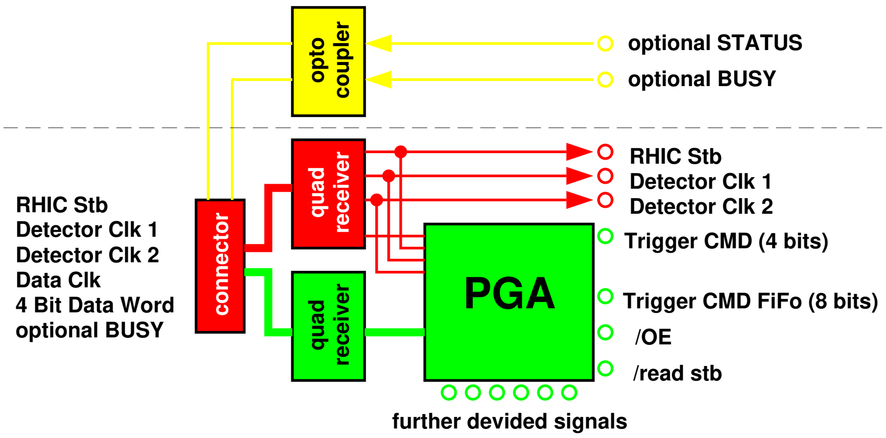

Figure 3: The components of the trigger/clock receiver.

3.3.2 Trigger/Clock Driver Module

The trigger/clock driver module is the root of the trigger/clock distribution tree. Its major components are sketched in figure 1. The trigger/clock driver module will be implemented as a 9U VME board. The use of this widely accepted standard bus allows easy configuration access using only standard components and interface chips. The trigger/clock Driver Module will be identical for all detector systems. However it will host a detector specific mezzanine board that implements all the detector specific logic required to produce the clock signals and special trigger command sequences.

Therefore there are four major components on the trigger/clock Driver Module: the VME slave interface, the trigger decode logic, the detector specific trigger/clock mezzanine board and the delay logic. The VME slave interface allows configuration of all parameters and CSR registers of the board including the trigger/clock mezzanine board. All resources will be memory mapped into the sub-address space of the trigger/clock Driver Module. The decode logic defines if a given trigger command includes the given detector. The most important component of the trigger/clock Driver Module is the detector specific mezzanine board. It generates the detector specific clock signals and the special command sequences. The output of the mezzanine board is serialized and forwarded to the fan-out modules. These 6U modules will be plugget in at the back side of the P2/P3 back plane. The signals of the P2 VSB bus will be used for passing through the trigger/clock signals. There are two sets of programmable delay line units. The actual delay used will be selected by writing to a CSR register.

The Trigger Back-Plane

The P3 bus of the trigger/clock Driver Module will be used to distribute the actual Trigger L0 TCU output signals (Trigger backplane). The logical signal groups distributed at the trigger back plane are:

❶ (1)	I	RHIC strobe
❷ (8)	O	individual detector BUSY signals (one bit per detector, open collector, low active)
❸ (8)	I	coincident detectors (one bit per detector)
❹ (4)	I	Trigger Command
❺ (4)	I	DAQ command
❻ (12)	I	Trigger Token
❼ (16)	I	Trigger Word

Note: items ❸ through ❺ form the 16 bit Trigger Action Word.

All signals are latched with the rising edge of the RHIC strobe. The setup time is 10 ns and the hold time is 5 ns.

The individual detector busy signals are required for special trigger generation. The time required to process a special trigger command is not deterministic, it may result in several special triggers to be sent to the front-end. Therefore has the special trigger sequencer on the trigger/clock detector specific mezzanine board to be able to assert the detector specific dead time signal.

The Trigger Action Word is split into three logical fields as indicated in the above table. Each detector (up to eight) has its own bit indicating if it is involved in the given trigger command. The trigger command and DAQ command are global for any given event.

The trigger token is distributed in case of a trigger command that results in a read-out.

The Trigger Word is the 16 bit field that caused the trigger to be fired for the actual bunch crossing.

There has to be a back-plane that distributes these trigger signals. Most elegant would be the use of an existing back-plane avoiding the need of the design of a private back-plane. It is possible to use VME P1 for this purpose. The P3 signal-pin assignment of the Trigger back plane is shown in Table 2.

Table 2: *VME-P3 pin assignment of the Trigger back plane*

pin	**signal (P3)**	**(P1)**	**pin**	**signal (P3)**	**(P1)**
A10	RHIC_STB	SYSCLK			
A01	TRG_EMC	D00	B30	BSY_EMC	IRQ1
A02	TRG_SVT	D01	B29	BSY_SVT	IRQ2
A03	TRG_TOF	D02	B28	BSY_TOF	IRQ3
A04	TRG_TPC	D03	B27	BSY_TPC	IRQ4
A05	TRG_XTPC	D04	B26	BSY_XTPC	IRQ5
A06	TRG_spare	D05	B25	BSY_spare	IRQ6
A07	TRG_spare	D06	B24	BSY_spare	IRQ7
A08	TRG_spare	D07	A20	BSY_spare	IACK
B18	TRG_CMD[0]	AM2	C16	DAQ_CMD[0]	AM0
B19	TRG_CMD[1]	AM3	C17	DAQ_CMD[1]	AM1
A24	TRG_CMD[2]	A07	C18	DAQ_CMD[2]	AM2
C24	TRG_CMD[3]	A14	C19	DAQ_CMD[3]	AM3
A30	TRG_TOKEN[0]	A01	C30	TRG_TOKEN[6]	A08
A29	TRG_TOKEN[1]	A02	C29	TRG_TOKEN[7]	A09
A28	TRG_TOKEN[2]	A03	C28	TRG_TOKEN[8]	A10
A27	TRG_TOKEN[3]	A04	C27	TRG_TOKEN[9]	A11
A26	TRG_TOKEN[4]	A05	C26	TRG_TOKEN[10]	A12
A25	TRG_TOKEN[5]	A06	C25	TRG_TOKEN[11]	A13
C23	TRG_WORD[0]	A15	C01	TRG_WORD[8]	D08
C22	TRG_WORD[1]	A16	C02	TRG_WORD[9]	D09
C21	TRG_WORD[2]	A17	C03	TRG_WORD[10]	D10
C20	TRG_WORD[3]	A18	C04	TRG_WORD[11]	D11
C19	TRG_WORD[4]	A19	C05	TRG_WORD[12]	D12
C18	TRG_WORD[5]	A20	C06	TRG_WORD[13]	D13
C17	TRG_WORD[6]	A21	C07	TRG_WORD[14]	D14
C16	TRG_WORD[7]	A22	C08	TRG_WORD[15]	D15

The Trigger/Clock Mezzanine

All detector specific clock and trigger signals and functions have to be produced on this board. The development of this mezzanine board is within the responsibility of each detector group. The physical board geometry and connectors are compliant with the IEEE P1386 Common Mezzanine Card Family (CMC) standard. The trigger/clock Mezzanine will be a double width, standard CMC board (149x149mm).

Note: The CMC standard was used only simplify the definition of the mezzanine board and connectors. The trigger/clock mezzanine board does not have to comply with all features of the P1386 standard. In fact the following deviations will be implemented in order to simplify the design:

- The BUSMODE# coding and V(I/O) pins are not used and will be ignored.
- The P13 connector will have the pin assignment of the P12 connector. The P12 and P14 connector will not be implemented at all. The P21 and P23 connectors will have only the power and ground pins connected as in Table 3. P22 and P24 will not be implemented, too. These changes to the P1386 standard allow to completely omit the second connector row of the standard and still ensure mechanical stability of the mezzanine board.

The signal groups on the CMC mezzanine connector are:

■ (6)	O	detector ID (3 bits), subdetector ID (3 bits)
trigger backplane inputs		
■ (3x1)	I	RHIC strobe, delayed RHIC strobe, detector select
■ (1)	O	detector BUSY
■ (4)	I	optional detector BUSY input from fan-out modules
■ (4)	I	optional detector STATUS input from fan-out modules
■ (4,16)	I	trigger CMD in, trigger word
trigger command serializer outputs		
■ (2)	O	distributed clocks (detector Clk1, detector Clk2)
■ (3)	O	serializer control (force trigger token and DAQ CMD to zero etc.)
■ (4)	O	trigger CMD out
configuration inputs		
■ (19)	I	mezzanine address bus
■ (8)	I/O	mezzanine data bus
■ (4)	I	CE#, OE#, WE#, RESET#

The six bit detectorID field contains three sudetectorID bits (D0...D2). The subdetectorID bits are required because there may be more than one trigger/clock driver module required for the given detector. The detector ID bits are statically set by the trigger/clock mezzanine board and will be used for the VME slave address decoding logic on the mother board. This allows kind of a geographic detector addressing within the VME framework.

The DAQ command word and trigger token bypass the trigger/clock mezzanine board in order to save connector pins.

Table 3: *CMC pin assignment of the trigger/clock mezzanine*

pin #	signal name	signal name	pin #	pin #	signal name	signal name	pin #
1	detectorID[0]	-12V	2	1	+12V	C0BUSY	2
3	GND	detectorID[1]	4	3	C2BUSY	C1BUSY	4
5	trig.CMDi[0]	trig.CMDi[1]	6	5	C3BUSY	GND	6
7	reserved,nc	+5V	8	7	GND	C0STATUS	8
9	trig.CMDi[2]	trig.CMDi[3]	10	9	C2STATUS	C1STATUS	10
11	GND	detector select	12	11	reserved,nc	reserved,nc	12
13	RHIC stb	GND	14	13	dly RHIC stb	reserved,nc	14
15	GND	detector busy	16	15	reserved,nc	reserved,nc	16
17	trig.CMDo[0]	+5V	18	17	CE#	GND	18
19	reserved,nc	trig.CMDo[1]	20	19	OE#	WE#	20
21	trig.CMDi[2]	trig.CMDo[3]	22	21	GND	CD[0]	22
23	control[0]	GND	24	23	CD[1]	reserved,nc	24
25	GND	control[1]	26	25	CD[2]	CD[3]	26
27	control[2]	detectorID[3]	28	27	reserved,nc	CD[4]	28
29	detectorID[2]	+5V	30	29	CD[5]	GND	30
31	reserved,nc	detectorID[5]	32	31	CD[6]	CD[7]	32
33	detectorID[4]	GND	34	33	GND	CA[0]	34
35	GND	detectorCLK1	36	35	CA[1]	reserved,nc	36
37	detectorCLK2	+5V	38	37	GND	CA[2]	38
39	GMD	RESET#	40	39	CA[3]	GND	40
41	trig.WD[0]	trig.WD[1]	42	41	reserved, nc	CA[4]	42
43	trig.WD[2]	GND	44	43	CA[5]	GND	44
45	reserved,nc	trig.WD[3]	46	45	CA[6]	CA[7]	46
47	trig.WD[4]	trig.WD[5]	48	47	CA[8]	CA[9]	48
49	trig.WD[6]	+5V	50	49	CA[10]	reserved,nc	50
51	GND	trig.WD[7]	52	51	CA[11]	CA[12]	52
53	trig.WD[8]	trig.WD[9]	54	53	reserved,nc	CA[13]	54
55	trig.WD[10]	GND	56	55	CA[14]	GND	56
57	reserved,nc	trig.WD[11]	58	57	CA[15]	CA[16]	58
59	trig.WD[12]	trig.WD[13]	60	59	GND	CA[17]	60
61	trig.WD[14]	+5V	62	61	CA[18]	reserved,nc	62
63	GND	trig.WD[15]	64	63	GND	C3STATUS	64

The 4-bit trigger command word is sent through the trigger/clock mezzanine in order to allow to intercept pulser triggers. This allows one pulser signal to result in a sequence of pulser trigger commands being sent to the front-end. During such a sequence the mezzanine board will assert the detector dead time. There are however trigger commands that may still be broadcast involving the busy detector like L1 and L2 accepts or aborts of earlier events. Therefore a trigger command FIFO is implemented on the Trigger/Clock Driver module (refer to figure 1), which absorbs and delays these commands during a pulser sequence. The control outputs instruct the sequencer to either transmit the trigger token driven by the mezzanine accompanied by a zero DAQ command word and zero trigger token or to distribute the latest trigger command with valid DAQ command and trigger token from the FIFO.

The control signals are defined as follows:

control[0]:	1:	transmit the trigger command defined by the mezzanine board force the DAQ command and trigger token to 0 (pulser mode)
	0:	transmit the latest trigger command read from the FIFO (default)
control[1]:		reserved
control[2]:		reserved

Table 3 shows the CMC P11 and P12 pin assignment. All signals are 5V TTL signals. There is no support for 3.3V logic. The connector P12 has only configuration signals connected. This allows trigger/clock mezzanine boards that have only static configuration parameters to ignore that connector.

The Trigger/Clock Fan-Out Back-Plane

The Trigger/Clock output of the programmable delay units will be fed through fan-out boards, which will be mounted from the back side of the 9U VME crate. The P2 VSB back-plane signals will be used for that purpose. All signals at this back-plane are 5V TTL signals. The chose of the VSB back-plane standard allows the use of this back-plane if more than one fan-out module is required per Trigger/Clock driver module. In this case the VSB back-plane would simply be mounted at the back side of the VME P2 back-plane. Depending on the depth of the VSB back-plane up to 8 fan-out boards could be driven by a single Trigger/Clock driver. The pin definition of the Trigger/Clock VSB back-plane is defined in Table 4.

The back side of the 9U VME board will have card guides supporting the 6U fan-out modules. These modules will sit behind the VME P2 and P3 back-plane because the VME power supply is typically mounted behind the P1 back-plane. The fan-out boards have to support hot insertion because all trigger/clock driver modules will be in one crate and a power cycle of that crate would affect the whole experiment. The connectors on the back side of the VME P2 or VSB back plane are of mail type. Bent pins may cause short cuts, which are dangerous especially if power pins are involved. Therefore the power of the VME P2 and VSB back-plane will be decupled from the P1 and P3 back-plane. The Trigger/Clock drivers are allowed to use only power from the P1 and P3 back-plane. The Trigger/Clock fan-out boards use only the P2 power pins. The P2 back-plane will have an individual fuse that will prevent the back-plane from burning upon a power shortcut due to bent connector pins. This is possible because the fan-out boards will not need a lot of power.

Table 4: *VME-P2 pin assignment of the Trigger/Clock fan-out*

pin	signal (P3)	(VSB)	pin	signal (P3)	(VSB)
A01	C0RHIC	AD00	C01	C1RHIC	AD01
A02	C0D0	AD02	C02	C1D0	AD03
A03	C0D1	AD04	C03	C1D1	AD05
A04	C0D2	AD06	C04	C1D2	AD07
A05	C0D3	AD08	C05	C1D3	AD09
A06	C0DCLK	AD10	C06	C1DCLK	AD11
A07	C0CLK1	AD12	C07	C1CLK1	AD13
A08	C0CLK2	AD14	C08	C1CLK2	AD15
C28	C0BUSY	ASACK0	C27	C1BUSY	ASACK1
A32	C0STATUS	BREQ	C31	C1STATUS	BUSY
A09	C2RHIC	AD16	C09	C3RHIC	AD17
A10	C2D0	AD18	C10	C3D0	AD19
A11	C2D1	AD20	C11	C3D1	AD21
A12	C2D2	AD22	C12	C3D2	AD23
A13	C2D3	AD24	C13	C3D3	AD25
A14	C2DCLK	AD26	C14	C3CLK	AD27
A15	C2CLK1	AD28	C15	C3CLK1	AD29
A16	C2CLK2	AD30	C16	C3CLK2	AD31
C25	C2BUSY	ACK	A24	C3BUSY	ERR
C29	C2STATUS	CACHE	C30	C3STATUS	WAIT

There has to be a commercial module in the crate that allows initialization of the Trigger/Clock drivers. This module may use both the P1 and P2 power pins. Therefore there will be two slots in the Trigger/Clock crate where the P1 and P2 power pins are connected. This crate will have correspondingly a 2 slot and a 19 slot P2 back-plane. These back-planes will be connected except for the power pins forming a virtual 21-slot VME P1/P2 back-plane. Such back-planes bridges with decoupled power pins are commercially available.

Configuration - Slow Control Interface

All configuration resources of the trigger/clock distribution tree are VME memory mapped. Therefore configuration and initialization can be performed by any processor in VME. It is assumed that the Slow-Control and Experiment-Control systems are able to call C-routines on a VME CPU. The arguments of the various initialization and maintenance routines have to be supplied and maintained by the calling shell. The following C-functions will be supplied:

- int reset (int detectorID)
- int configure (int detectorID, struct *base_config, struct *mezzanine_config)
- int read_config (int detectorID, struct *base_config, struct *mezzanine_config)

All the listed functions return NULL on success and the error code in case of an error.

The data format of the passed and returned data structures has to be defined platform independently to allow to run on any and not yet specified system. Therefore the IEEE standard number 1596.3-1993 *shared data formats* will be used to define these data types platform independently.

Note: These are routines that will call the appropriate functions for the mezzanine boards. The mezzanine boards, however, are detector specific and require detector specific handling. Therefore the mezzanine specific routines have to be supplied by the group developing the mezzanine board itself. The shell routines listed above will decode the detector ID and call the following functions, where XXX stands for the appropriate detector acronym (capitalized):

- int XXX_reset ()
- int XXX_configure (struct *mezzanine_config)
- int XXX_read_config (struct *mezzanine_config)