Specification for a FERA-VXI interface CENBG version 2.1

MM Aleonard, JP Goudour, JL Pedroza

December 16, 1999

This document presents the specification of a **F**ERA to **V**XI **I**nterface (**FVI**). This unit will be useful in experiments including FERA ADCs in the EUROBALL equipement because it supplies a standard data path for the ADC's data and FERA commands depending of the EUROBALL MASTER TRIGGER and of the ANCILLARY TRIGGER.This document is the result of a meeting at CENBG between MM Aleonard, JP Goudour and the author. Any comment and suggestion can be sent at:

pedroza@cenbg.in2p3.fr

Contents

Introduction

At the Euroball EDA meeting held on 12/13th Oct. 1995 at TSL Uppsala it was decided to build a VXI/FERA card with the main points :

- the FERA readout is a VXI size D module with VXI interface and readout using a GIR.
- must control and read all 4 FERA types: Lecroy, Silena, Ortec, Gannelec.
- must interface FERA with the EUROBALL trigger system to synchronise the ex ternal detectors.
- must generate an event number to tag each event.
- must be able to generate variable length gates with "sufficient" precision for FERA QDCs.
- must derive FERA gate timing from either front panel input or VXI triggers (user selects).
- must check for a EUROBALL validation before FERA readout.
- must generate EUROBALL data format.
- must have watchdog timeout for error recovery.

This specification is organized in 3 chapters. Chapter one describes the main functions of the ancillary trigger and its constraints conpared with the Euroball master trigger constraints, in particular for the timing and the dead time. Depending of these constraints, the main functions of the **FVI** are described. The chapter two shows the requirement model of the **FVI**. Then the chapter three describes the architectural model of the **FVI**

Chapter 1

The ancillary trigger functionnalities

1.1 The triggering of FERA ADCs

The figure 1.1 shows the actions taken by the ancillary trigger to start the coding of the FERA ADcs.

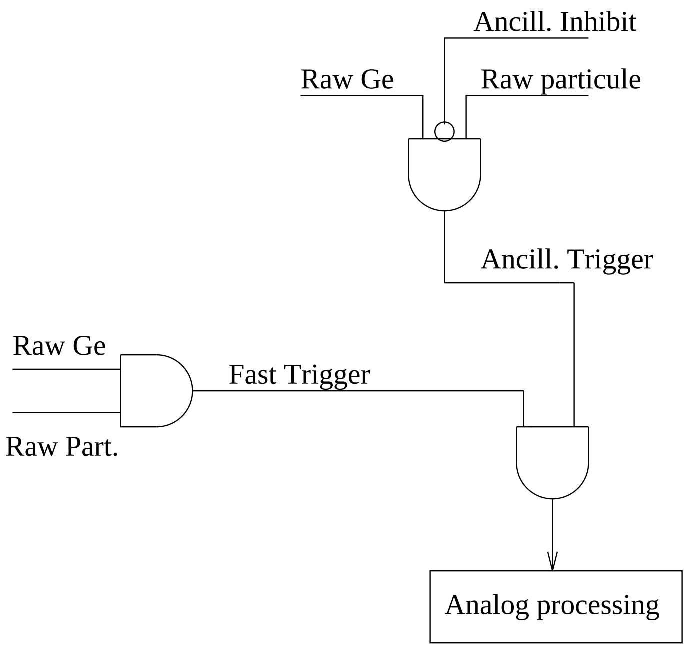

Figure 1.1: block diagram of the ancillary trigger

To start a FERA coding, the ancillary trigger is in agreement with the followings rules:

1. **must** detect a Raw Particule from ancillary detectors.
2. **must** check if a Fast Trigger from Master Trigger is received.
3. **can** wait for a Fast trigger if the previous rule is false.
4. **must** open the analog gates and strechers for the FERA ADCs.

During the analog processing, the ancillary trigger is in agreement with the followings rules:

1. **must** check a pileup and assert a reject signal if true.
2. **must** receive an inhibit signal to discard new Raw Particule or Fast Trigger.

On completion of the coding, the Ancillary trigger is in agreement with the followings rules:

1. **must** Clear its internal logic after FERA readout.
2. **must** know if a Fast Trigger from master trigger is occured.
3. **must** know if a Valid from master trigger is occured.

At any time, the ancillary trigger must also know if:

1. the Euroball is working or not via the **GO*/STOP** line.
2. the FERA readout is completed or aborted via TIMEOUT.

the timing constraints for the ancillary trigger and the master trigger are described in the next section.

1.2 Ancillary trigger timing versus Master Trigger timing

The figure 1.2 shows the timing of the EUROBALL ft and ancillary trigger to get an ancillary tigger signal. It shows also the dead time between the start of FERA coding and the end of the FERA readout.

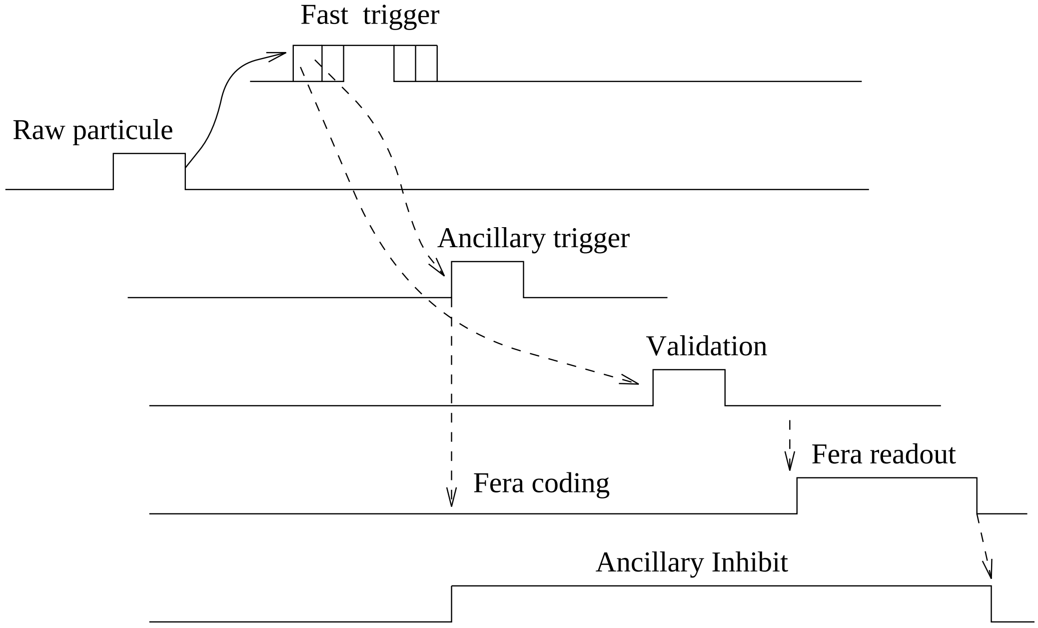

Figure 1.2: timing of ancillary trigger generation

The delay to start the FERA readout depends of the time required by the FERA ADCs for the coding. This time is between two values, depending of the ADCs types:
- 13μSec for Lecroy.
- 25μSec for Ganelec.

In the EUROBALL master trigger, Validation may not be delayed longer than 10μSec. In order to do the Fera coding, the FERA readout and the VXI readout with concordance of the timing, the **FVI** must have the fonctionnalities which are exposed in the next chapter.

Chapter 2

The FVI requierement model

2.1 Context

The figure 2.1 shows the context diagram of the **FVI**. The edges of the model are the EUROBALL master trigger, the ANCILLARY trigger, the VXI interface (GIR) and the FERA ADCs.

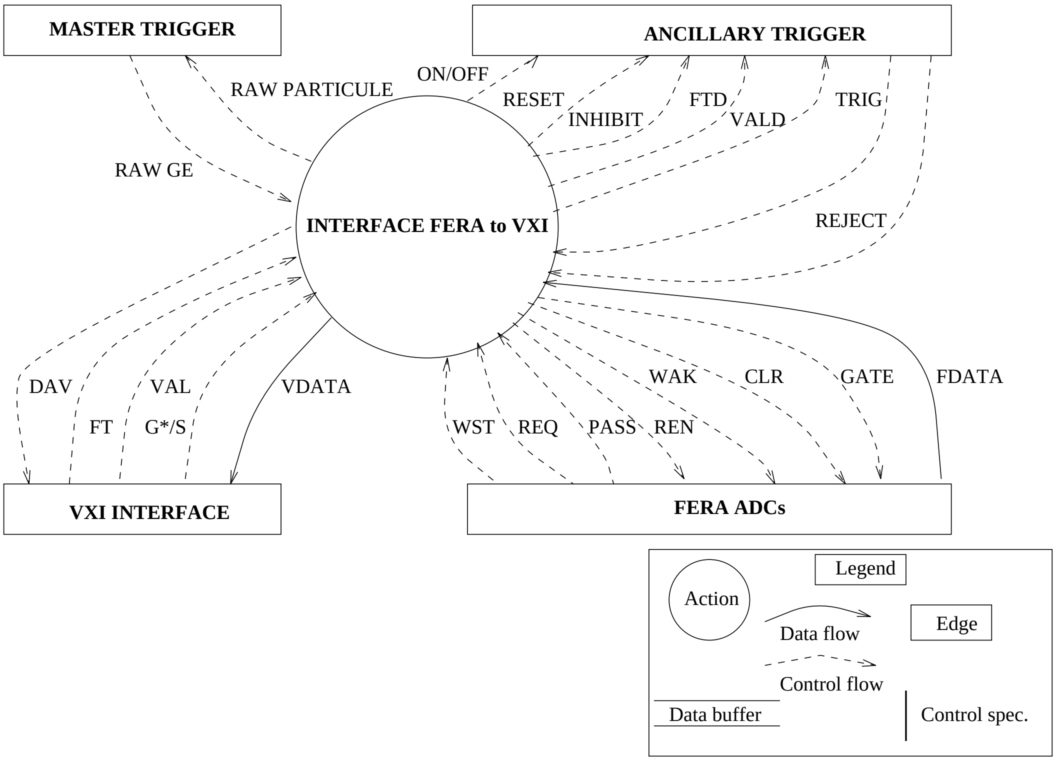

Figure 2.1: FVI context diagram (main functions)

2.2 Data and Control dictionary

Here are explained the Data and Controls symbols for all the context, that is for the I/O of the **FVI**. The levels and timing are also decribed.

Name	Function	Logic	Timing
RAW PARTICULE	Fast decision from Anc. Trigger	?	?
RAW GE	Raw Ge from Master Trigger	?	?
RESET	Clear Anc. Trigger	NIM	?
INHIBIT	Discard FT and Val	NIM	set by TRIG Cleared by last PASS
TRIG	Trigger from Ancillary	NIM	?
REJECT	Anc. pileup rejection	NIM	Duration of FERA coding
ON/OFF	Anc. trigger input	NIM	-
WST	Fera write strobe	ECL	min 40 nS
WAK	Fera write ackn.	ECL	min 30 nS
CLR	Fera Clear	ECL	min width 5 nS
REQ	Fera Request	ECL	-
REN	Fera Ren	ECL	maintained during entire readout
PASS	Fera branch last Pass	ECL	-
GATE	Fera gate	ECL	set by TRIG delayed from 0 to 20μSec Step 10 nSec width from 10 nSec to 20 μSec Step 10 nSec
FTD	Fast Trigger delayed	NIM	idem GATE
VALD	Validation delayed	NIM	idem GATE
FDATA	Fera Data from ADCs	ECL	-
FT	Fast Trigger EUROBALL	TTL	?
VAL	EUROBALL Validation	TTL	?
DAV	DATA available	TTL	-
G*/S	Go*/Stop signal	TTL	-
VDATA	Available data for VXI interf.	TTL	-

2.3 Main functions

To perform the FERA coding and the data readout through the data path, the **FVI** provides the followings functions:

1. handshake with the triggers (Master trigger and ancillary trigger)
2. handshake with the general EUROBALL Go/Stop
3. FERA branch management
4. Gates generator
5. FERA readout
6. VXI readout
7. Timeout management
8. internals tests

The nexts sections describe each functionnality.

2.4 Handshake with the triggers

When the ancillary trigger detects a raw particule (**RAW PARTICULE**), it send it to the Master trigger and start the ancillary trigger signal (**TRIG**) in coincidence with the **Raw Ge** signal from the Master trigger. Depending of somes delays, the Fast Trigger sent by the master trigger (**FT**) can be received either before the ancillary trigger (**TRIG**) or after. Depending of the experiment conditions, the master trigger can or not send a validation (**VAL**). When **TRIG** is asserted, the **FVI** must start the FERA coding (**GATE**).
During all the FERA coding and FERA readout, the **FVI** asserts **INHIBIT**.
2 timeouts check the arrival of **VAL** and **FT**. These timeout can be turned ON or OFF. If one timeout occurs, the **FVI** clears all the FERA ADCs
When the **FVI** receives **FT** it sends **FTD** to the ancillary trigger with adjustable delay and width.
When the **FVI** receives **VAL** it sends **VALD** to the ancillary trigger with adjustable delay and width.
When the ancillary trigger detects a pileup during the FERA coding, it asserts **REJECT**, then the **FVI** must clear all the FERA ADCs. If **REJECT** is asserted after the FERA coding, the event is good and the readout is normally performed.
On completion of the coding and FERA readout sequence (normal or ended by a timeout, the **FVI** sets **RESET**.

2.5 Handshake with the general EUROBALL Go/Stop

The signal **ON/OFF** sent to the ancillary trigger is the general **GO*/STOP**. When the VXI signal **GO*/STOP** from EUROBALL is low, then **ON/OFF** is high. When the VXI signal **GO*/STOP** from EUROBALL is high, then **ON/OFF** is low.

2.6 FERA branch management

The **FVI** is a D size VXI module. the front panel supplies the FERA commands and data, the NIM I/O and somes probes and flags as descibed in the nexts sections. The size does not allows more than two FERA branches per module but it will be possible to connect together several modules, that is to control more than 2 FERA branches. There is no **FVI** "Master" or "Slave", each module is identical but the ancillary trigger must:

1. handshake with only ONE **FVI**.
2. receive **INHIBIT** from every **FVI**
3. wait FERA coding and FERA readout completion from every **FVI** before restart its own logic.

2.7 Gates generator

The gates generator has a 100 MHz clock fequency, 50 ppm. The **FVI** provides two gates (one per branch). Each gate is set when **TRIG** is true with programmable delay and width. The programmable delay range is 0 to 20 μSec, with 10 nSec steps. The programmable width range is 0 to 20 μSec, with 10 nSec steps. On normal condition, the FERA readout sequence is started when the **GATE** is done. If a timeout occurs (as described is the next sections), then each **GATE** is cleared and set for the next cycle.

2.8 FERA readout

On completion of the coding sequence the **REQ** signal is received from each FERA branch. On receipt of **REQ** on one FERA branch, the **FVI** must assert the **REN** with an adjustable delay between 0 and 50 μSec, 100 nSec steps. A timeout checks the arrival of **REQ** for each FERA branch. If a timeout occurs, the **REN** is asserted and the readout sequence is started on the fera branch.
When the FERA **WST** is received from each branch, the **FVI** must :

1. convert the data to the **EUROBALL** format.
2. store the converted data into a FIFO.

3. send **WAK** to the FERA branch.

this sequence is repeated until the (last) **PASS** from every FERA branch is received. There is one FIFO per FERA branch. When **PASS** is received from the two branches, all the FERA ADCs connected to the **FVI** are read and the **FVI** must :

1. Wait for **GIR** readout completion (else because the **GIR** FIFO is empty or the **FVI** data FIFOs are empties; this choice is done by the logic which drives the **GIR** readout). Then the **FVI** can deassert its local **INHIBIT** and the **INHIBIT** sent to the ancillary trigger if the bit **CDT** is not set in the control register (see section registers). If the **CDT** bit is set, the **INHIBIT** sent to the ancillary trigger will remain set until every **INHIBIT** from every **FVI** will be cleared. When **INHIBIT** sent to the ancillary trigger is cleared a new coding sequence can be launched when **TRIG** is received.

2. If during the FERA readout one FIFO is **HALF-FULL**, the **INHIBIT** remains true to prevent data loss until The FIFO is **EMPTY**.

2.9 Data conversion

The FERA data from each FERA branch are translated to the acquisition data format (ie **EUROBALL**) through a Lookup table. To translate the FERA data to the **EUROBALL** data format, the software must:

1. write the VSN in the **FVI VSN select registers** for each group of ADC's (Lecroy, Silena etc...). The **FVI** provides 4 registers for each FERA branch

2. write the LUT with the new data format for each VSN group

2.10 VXI readout

The **GIR** manages the data transfer from the **FVI** FIFOs to the **GIR** FIFO. The **FVI** FIFOs size is 4Kbytes and 32 bit data format. The **GIR** manage also the VXI readout, that is the data transfer from the **GIR** FIFO through the VXI bus.
To take part in the current event data flow through the VXI bus, the **FVI** must:

1. Tell the **GIR** that the current event (started by **TRIG**) will be available by setting **VALACK** true on the falling edge of **TRIG**.

2. Latch the EVENT NUMBER from the four bits given by the **GIR EVENT-IN** output to the **GIR EVENT-OUT** input on the falling edge of **VAL**.

To transfer the data from the **FVI** FIFOs to the **GIR** FIFO the **FVI** must :

1. Tell the **GIR** that somes data will be available in the **FVI** FIFOs by setting **START-READOUT** true on the falling edge of **TRIG**. The **GIR** asserts **REN-IN** and waits for a data available in the first FIFO of the **FVI**.

2. When this data is available, the **FVI** sets **DATA-READY**. Then, the **GIR** writes this data in its FIFO and asserts **DATA-ACK**. At last, the **FVI** deasserts **DATA-READY**. This handshake is repeated until all the data from the first **FVI** FIFO are transfered to the **GIR** FIFO. Then the **FVI** desasserts **START-READOUT** and asserts **END-READOUT** which is connected to the **START-READOUT** of the second **FVI** FIFO.

3. On completion of the same handshake **DATA-READY/DATA-ACK** for the second FIFO, the **FVI** sets **END-READOUT** which is sent to the **GIR**.

4. If a **REJECT** or a timeout occure as explained in the next section, **START-READOUT** is cleared and **END-READOUT** is set by the **FVI**.

5. On completion of the readout, the **GIR** asserts **LTRESET** which clears **VALACK**.

2.11 Timeout management

The timing control needs some timeout for the error recovery. The timeout must check :

1. **FT** arriving if **TRIG** is set the first.

2. VAL arriving

3. REQ arriving

Four programmables timers are implemented : TIMER0 for the FT waiting, TIMER1 for the VAL waiting, TIMER2 and TIMER3 for the REQ waiting on each FERA branch. Each timer generate a timeout if the condition which is expected is not reached. These timeout are:

1. For TIMER0, **TIMEOUT0**:
   This timer is started by **TRIG** cleared by **FT** or the timeout.
   The range is 0-5 μSec and the step 500 nSec.
   This timer must be enabled or desabled by software to allow waiting for FT or not.
   On timeout, the **FVI** must :
   Clear the **GATE** and the FERA ADCs on each FERA branch if true.
   Clear the **INHIBIT** signal if true.
   Clear the **REN** signal if true.
   Assert **RESET** to the ancillary trigger.
   Clear the **FTD** signal if true.

2. For TIMER1, **TIMEOUT1**:
   This timer is started by **TRIG** cleared by **VAL** or the timeout.
   The range is 0-15 μSec and the step 500 nSec.
   This timer must be enabled or desabled by software to allow waiting of VAL or not.
   On timeout, the **FVI** must :

   Clear the **GATE** and the FERA ADCs on each FERA branch.
   Clear the **INHIBIT** signal if true.
   Clear the **REN** signal if true.
   Assert **RESET** to the ancillary trigger.
   Clear the **VALD** signal if true.

3. For TIMER2 and TIMER3, **TIMEOUT2**,**TIMEOUT3**:
   These timers are started by **GATE** on each FERA branch cleared by **REQ** or the timeout on each FERA branch.
   The range is 0-30 μSec and the step 500 nSec.
   On timeout, the **FVI** asserts **REN** to start the FERA readout.

2.12 Internal tests

The **FVI** provides somes internal tests to check the fonctionnalities with and without the EUROBALL and FERA equipments. To perform these tests, the **FVI** must have some bits in the Control and status registers. The test procedure is set or inhibed by a bit in the Control Register. On test the **FVI** , when the test bit is true can :

1. For the EUROBALL trigger:

   Simulate **FT**.
   Simulate **VAL**.

2. For the ancillary trigger:

   Simulate **TRIG**.
   Simulate **REJECT**.

3. For the FERA branches:

   Simulate **REQ** for branch 1 and branch 2.
   Write data in each FIFO through the VXI bus.

When the test mode is ON, FT, VAL, TRIG, REJECT, REQ1, REQ2 and data from FERA are without effect on the **FVI** behavior.

2.13 Data and Control flows

The functions described in the previous sections can be arranged in four main actions:

1. Interface TRIGGERS
2. Manage FERA
3. Manage VXI
4. Test

the figure 2.2 shows the data and control flows diagram. This diagram summarize all the signals described on the previous sections and shows their interactions.

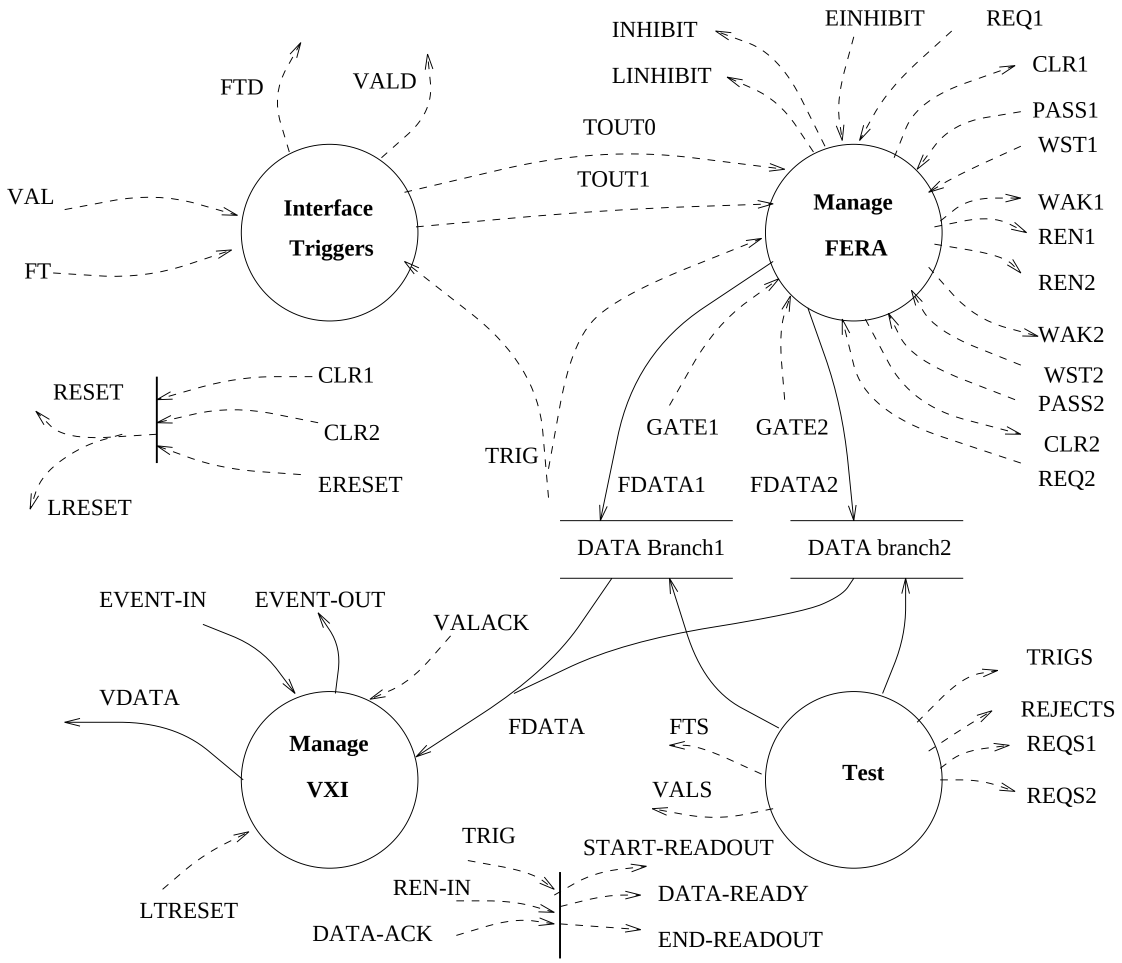

Figure 2.2: FERA/VXI DFD-DFC

The data and controls dictionary for the figure 2.2 is :

Name	Function
LRESET	Local RESET, chained to the next module
RESET	RESET for the ancillary trigger , same as LRESET
ERESET	LRESET, chained from a previous module The output of LRESET is the input of ERESET from the previous module
TOUT0	Timeout on FT
TOUT1	Timeout on VAL
LINHIBIT	Local INHIBIT, chained to the next module
EINHIBIT	LINHIBIT chained from a previous module The output of LINHIBIT is the input of EINHIBIT from the previous module
INHIBIT	INHIBIT for the ancillary trigger
GATE1	FERA GATE for branch1
GATE2	FERA GATE for branch2
CLR1	FERA CLEAR for branch1
CLR2	FERA CLEAR for branch2
REQ1	FERA REQUEST for branch1
REQ2	FERA REQUEST for branch2
REN1	FERA REN for branch1
REN2	FERA REN for branch2
WST1	FERA WST for branch1
WST2	FERA WST for branch2
WAK1	FERA WAK for branch1
WAK2	FERA WAK for branch2

Name	Function
PASS1	FERA LAST PASS for branch1
PASS2	FERA LAST PASS for branch2
FDATA1	FERA DATA from branch1
FDATA2	FERA DATA from branch2
FDATA	FERA DATA to the GIR
VDATA	VXI DATA from the GIR
EVENT-IN	EVENT NUMBER from GIR
EVENT-OUT	EVENT NUMBER to GIR
LTRESET	GIR end of readout
VALACK	FVI participation for the current event
START-READOUT	FVI data presumption
REN-IN	GIR ready for readout
DATA-READY	FVI data available
DATA-ACK	GIR data acknowledge
END-READOUT	FVI readout done
FTS	FAST TRIGGER in test mode
VALS	EUROBALL validation in test mode
TRIGS	ancillary TRIGGER in test mode
REJECTS	ancillary REJECT in test mode
REQS1	FERA REQ for branch1 in test mode
REQS2	FERA REQ for branch2 in test mode

The figure 2.3 shows the data and control flows diagram for the management of the 2 FERA branches. This action is made up of the FERA CODING, the FERA READOUT and the FERA CLEAR for each FERA branch.

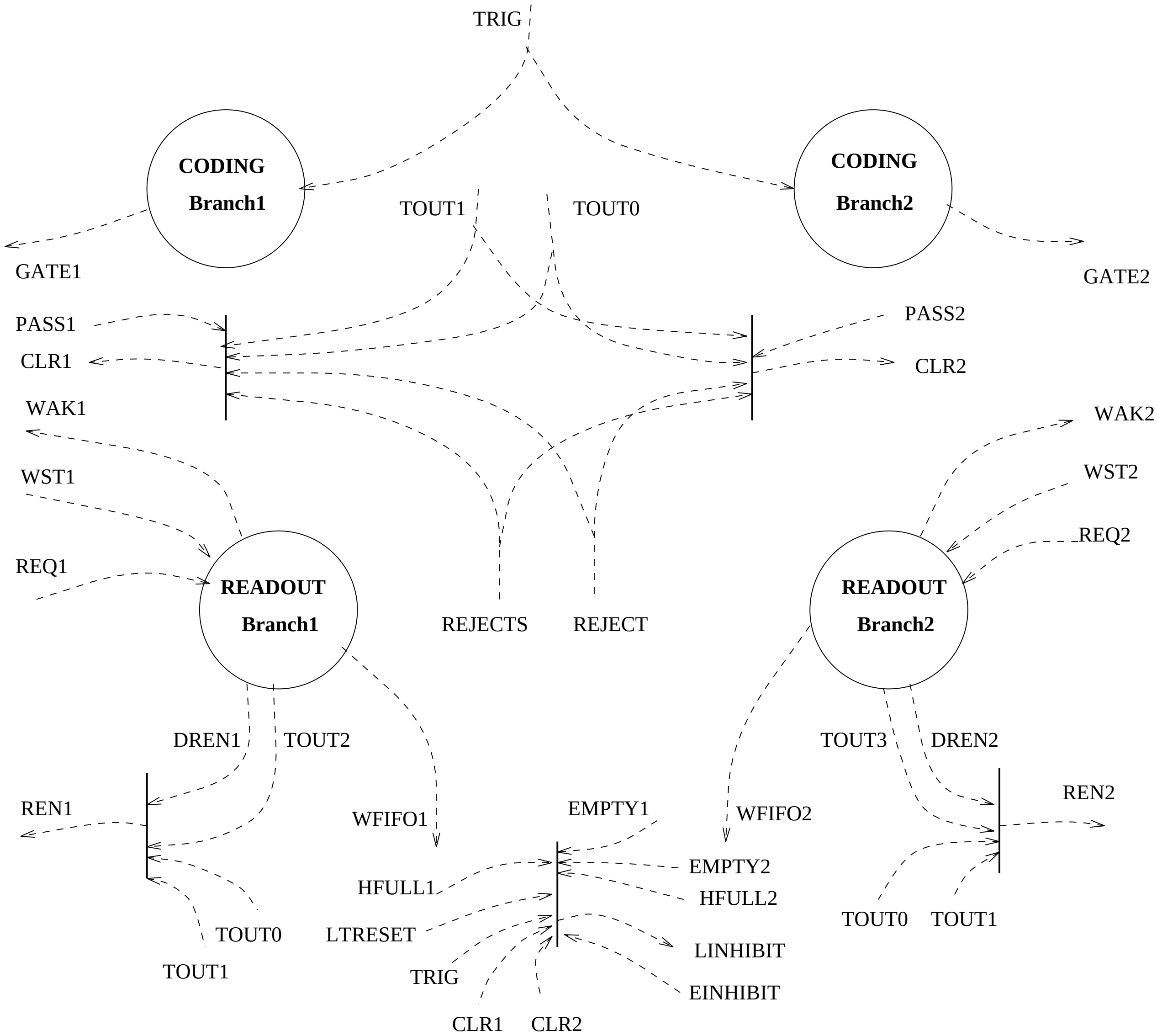

Figure 2.3: FERA MANAGEMENT on **FVI**

The data and controls dictionary for the figure 2.3 is :

Name	Function
DREN1	Delayed normal REN for branch1
DREN2	Delayed normal REN for branch2
WFIFO1	write strobe for data FIFO branch1
WFIFO2	write strobe for data FIFO branch2
TOUT2	TIMEOUT on REQ1
TOUT3	TIMEOUT on REQ2
HFULL1	FIFO branch1 HALF-FULL
HFULL2	FIFO branch2 HALF-FULL
EMPTY1	FIFO branch1 EMPTY
EMPTY2	FIFO branch2 EMPTY
REJECT	pileup from ancillary trigger

2.14 Controls and timings

This section summarize the controls specification and specify the timing constraints. It identify the relationships among the events which control the **FVI**. The model used is the State Transition Diagram (STD). The figure 2.4 shows the description of the symbols used in a STD. The Timings give the mains phases in the **FVI**.

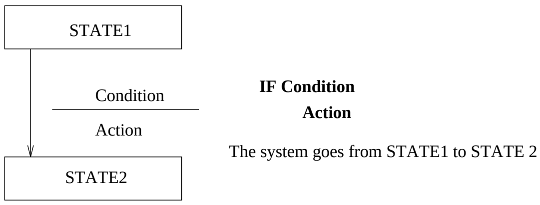

Figure 2.4: STD symbols

The figure 2.5 shows the timing for **FT** or **VAL** waiting, for **REQ** waiting and the timeout actions

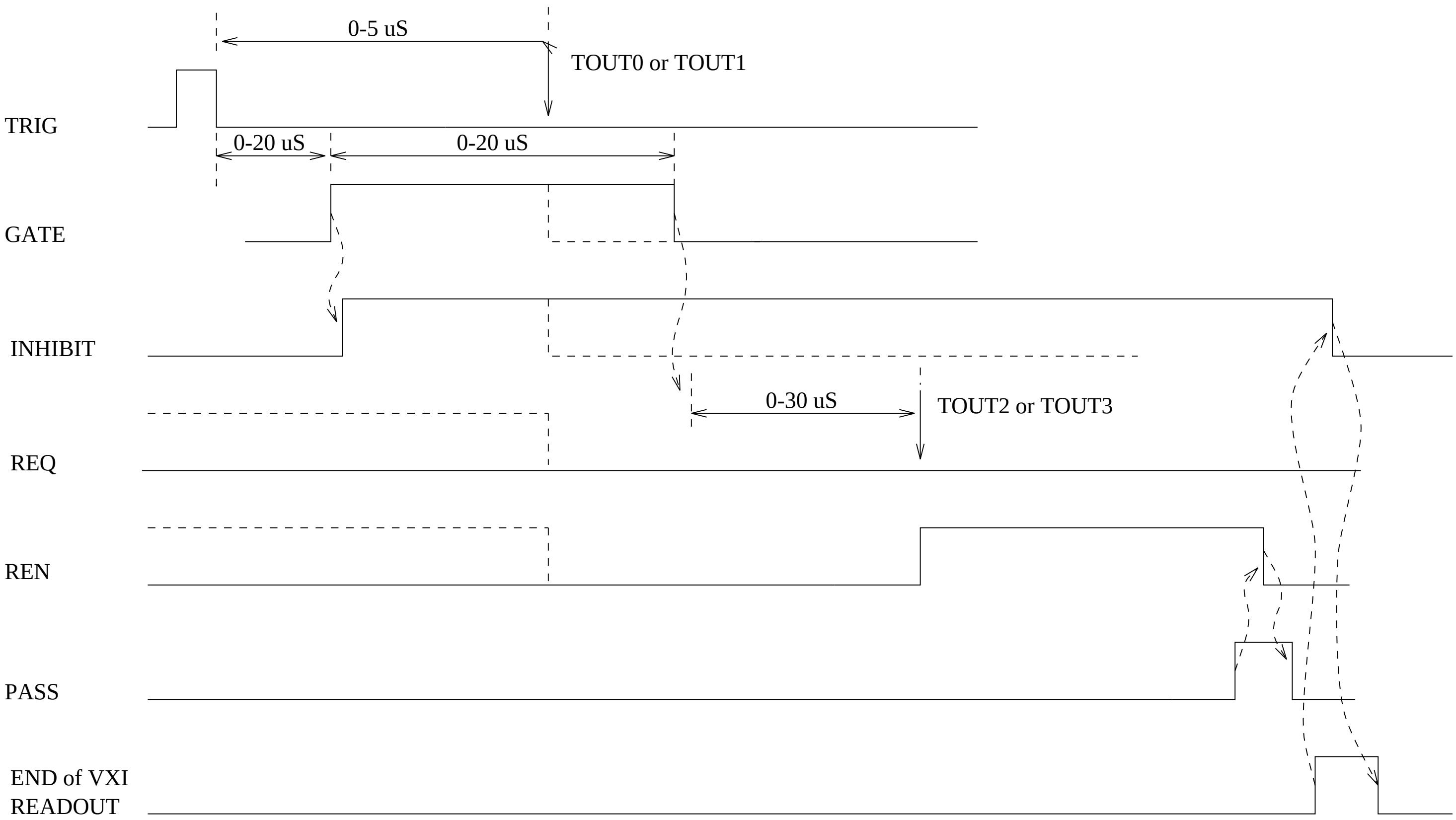

Figure 2.5: TIMEOUTs for FT, VAL, REQ

The figure 2.6 shows the STD for **RESET**.

The figure 2.7shows the STD for **CLR1** and **CLR2**.
The figure 2.8 shows the STD for **INHIBIT**.

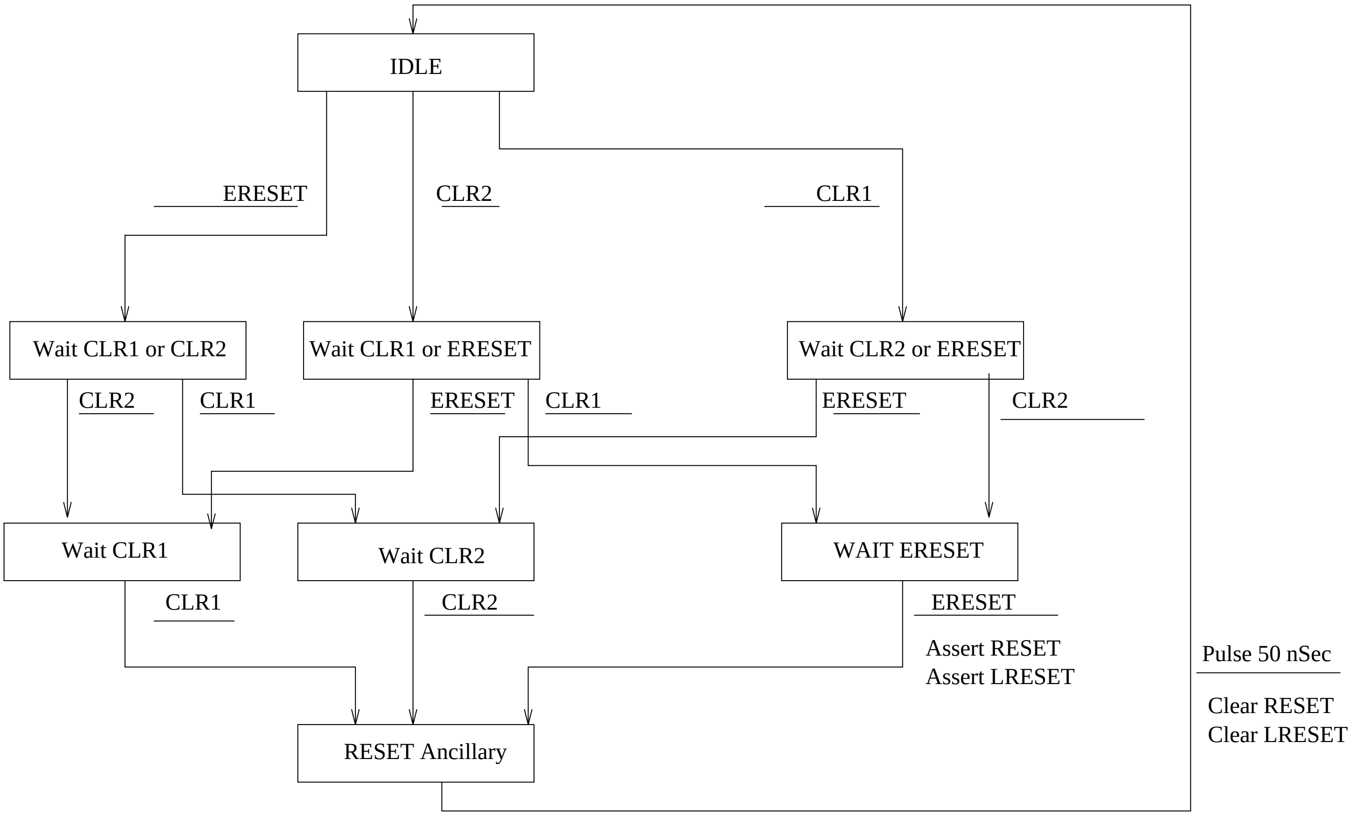

Figure 2.6: RESET

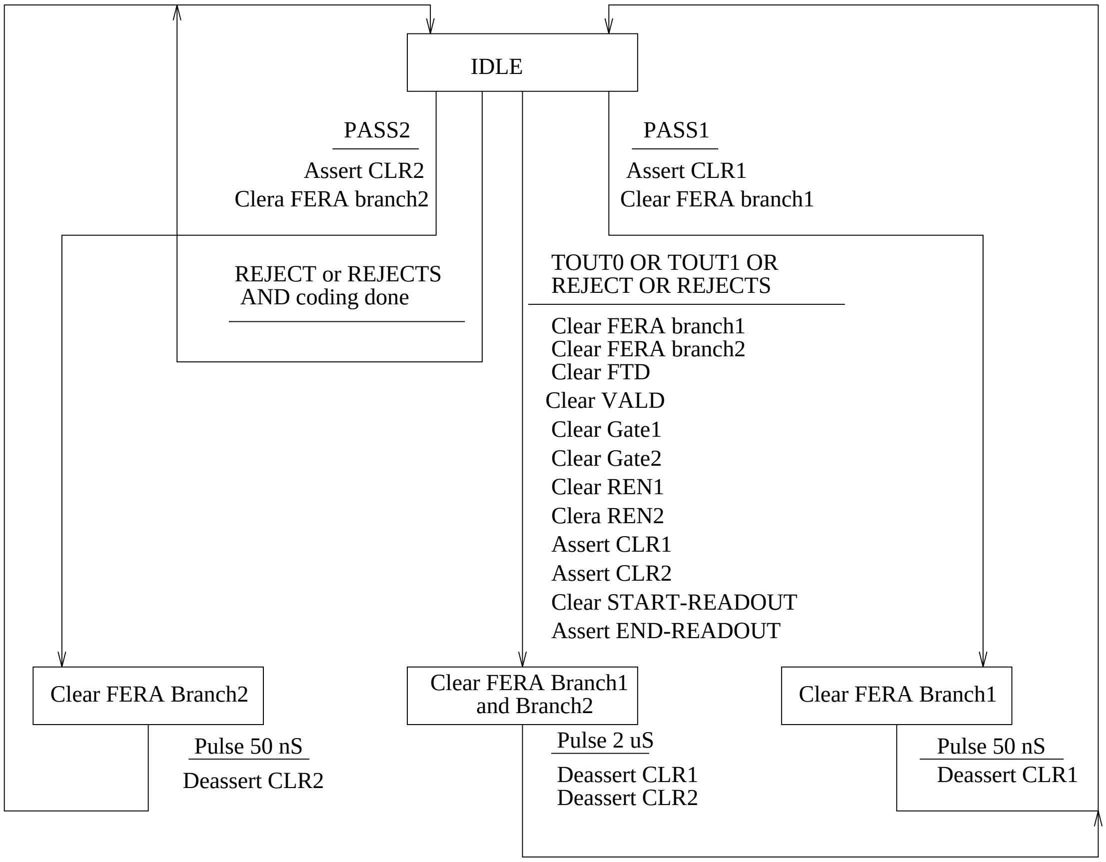

Figure 2.7: CLR1 and CLR2

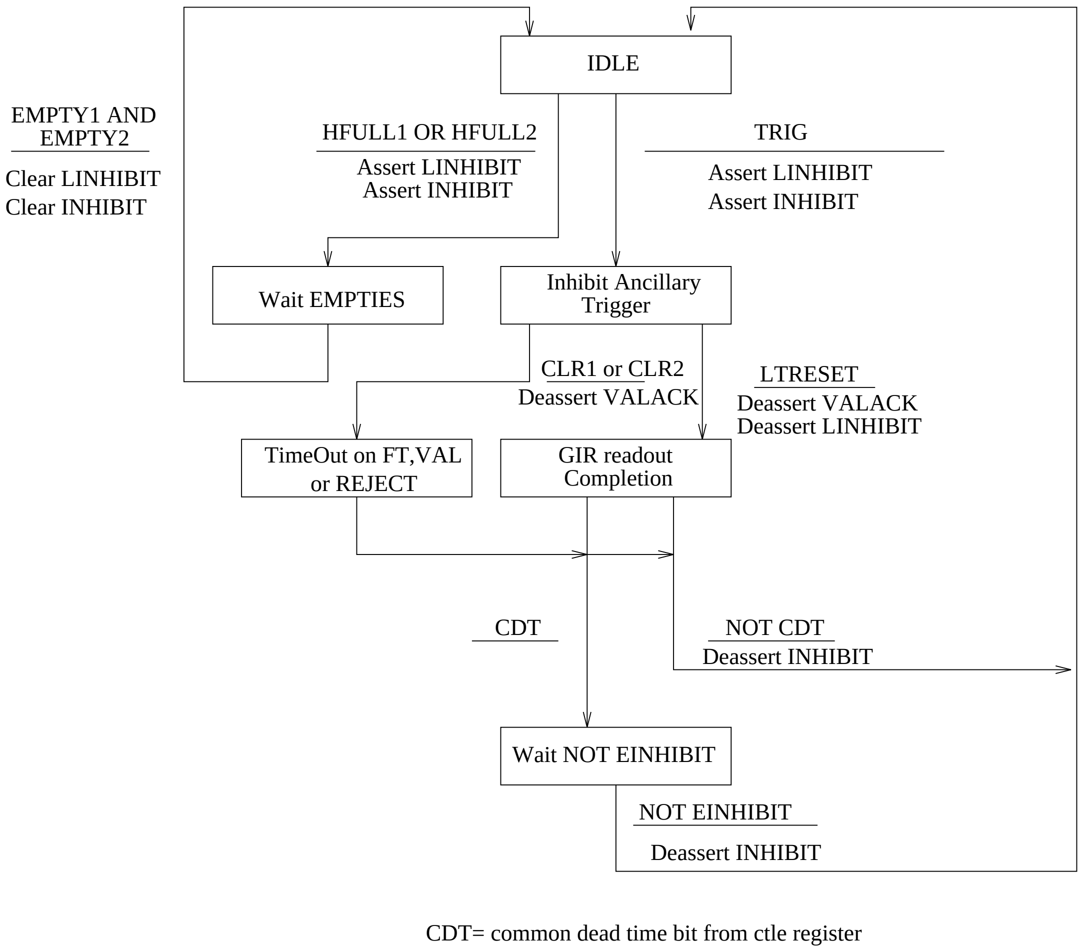

Figure 2.8: INHIBIT

Chapter 3

Architectural model

As described in the chapter two, the **FVI** manage 2 FERA branches. The VXI interface is managed by the **GIR**.

3.1 Front panel equipement

The **FVI** module is a size D VXI card, with somes connections in front panel.

1. LEMO connectors for FTD, VALD, TRIG, REJECT, INHIBIT, RESET and ON/OFF.
2. FERA data and command connectors for every branch.

3.2 VXI logical inspection lines

The 2 VXI lines for logical inspections are used.

1. WST1, REN1, GATE1.
2. WST2, REN2, GATE2.
3. RESET, INHIBIT, REJECT, TRIG.

3.3 Front panel indicators

Somes front panel indicators are implemented.

1. Given by GIR: ALARM, EVENT ERROR, DONE GIR LCA.
2. Given by the FVI : TEST, CODING, STOP.

3.4 Voltage inspection

The **FVI** provides inspection voltage lines to check the levels of all the voltages used by the module.

1. VCC
2. VEE
3. GND
4. VBB

3.5 Counters

The **FVI** must have Three counters. All the counters have a synchronized clear which is set by one bit of the **Control Register** (bit=1). The deep of each counter is 32 bits. At any time, it will be possible to read the counters without loss of values. The counters are used to count :

1. TRIG
2. FT
3. VAL

3.6 Model reference

The **FVI** hold a 32 byte PROM which contains the references of the card (Manufacturer ID, dacrd type ...)

3.7 Temperature sensor

A tempearture sensor checks the temperarute of the module and send the status to the GIR.

3.8 Registers description

The registers used in the **FVI** module are implemented on the **FVI** card and on the **GIR** card. This specification describes the registers used to manage the FVI and not the registers used in the **GIR** management.The global addresses of all the registers and the **GIR** management registers are described in the document **"GIR technical document for FVI"**. Three majors areas hold the registers:

1. the channel 1 area for the FERA branch 1
2. the channel 2 area for the FERA branch 2
3. the Common area

3.8.1 Channel 1 area

Name	Function	Access	Size(bytes)
LUT1	lookup table	r/w	32k
TIMEOUT2	Timeout on REQUEST	r/w	2
REN1DELAY	Delay between REQUEST and REN	r/w	2
GATE1DELAY	Delay between TRIG and GATE	r/w	2
GATE1WIDTH	Gate width	r/w	2
FERACTL1	FERA control	r/w	2
FERAFIFO1	FERA FIFO	r/w	4
VSN1SEL1	VSN select group LECROY	r/w	2
VSN1SEL2	VSN select group GANELEC	r/w	2
VSN1SEL3	VSN select group SILENA	r/w	2
VSN1SEL4	VSN select group ORTEC	r/w	2

The 3 most significant bit of the VSN are used to find the FERA ADCs types.

3.8.2 Channel 2 area

Name	Function	Access	Size(bytes)
LUT2	lookup table	r/w	32k
TIMEOUT3	Timeout on REQUEST	r/w	2
REN2DELAY	Delay between REQUEST and REN	r/w	2
GATE2DELAY	Delay between TRIG and GATE	r/w	2
GATE2WIDTH	Gate width	r/w	2
FERACTL2	FERA control	r/w	2
FERAFIFO2	FERA FIFO	r/w	4
VSN2SEL1	VSN select group LECROY	r/w	2
VSN2SEL2	VSN select group GANELEC	r/w	2
VSN2SEL3	VSN select group SILENA	r/w	2
VSN2SEL4	VSN select group ORTEC	r/w	2

The 3 most significant bit of the VSN are used to find the FERA ADCs types.

3.8.3 Commun area

Name	Function	Access	Size(bytes)
TIMEOUT0	Timeout on Fast Trigger	r/w	2
FTDDELAY	Delay between Fast Trigger and FTD	r/w	2
FTDWIDTH	FTD width	r/w	2
FTCOUNT	Fast Trigger counter	r/w	4
TIMEOUT1	Timeout on Validation	r/w	2
VALDDELAY	Dealy between VALIDATION and VALD	r/w	2
VALDWIDTH	VALD width	r/w	2
VALCOUNT	Validation counter	r/w	4
TRIGCOUNT	Ancillary TRIG counter	r/w	4
DIGCSEL1	Logical inspection channel select 1	r/w	2
DIGPSEL1	Logical inspection parameter select 1	r/w	2
DIGCSEL2	Logical inspection channel select 2	r/w	2
DIGPSEL2	Logical inspection parameter select 2	r/w	2
VOLTCSEL	Voltage inspection channel select	r/w	2
VOLTPSEL	Voltage inspection parameter select	r/w	2
FVICR	FVI card configuration registers	r/w	32
MCR	Module control register	r/w	2
MSR	Module status register	ronly	2
FTP	Simulate fast trigger pulse 250ns	wonly	2
VALP	Simulate validation pulse 250ns	wonly	2
TRIGP	Simulate ancillary trigger pulse 250ns	wonly	2
CFTC	Clear fast trigger counter pulse 250ns	wonly	2
CVALC	Clear validation counter pulse 250ns	wonly	2
CTRC	Clear ancillary TRIG counter pulse 250ns	wonly	2

3.8.4 Module Control Register

The 16 bits **MCR** must have the following bits to :

1. RESET the module, that is clear the FERA branches, clear the FIFOs, clear **REN**, assert **LRESET** and **RESET**.
2. VALIDATE or INHIBIT TIMER0 and TIMER1 to discard the fonctionnalities of the EUROBALL MASTER TRIGGER.
3. VALIDATE or INHIBIT the TEST MODE.
4. VALIDATE or INHIBIT the TEST functions to simulate **REJECT**and **REQ** on FERA branches 1 and 2.

5. Insert or separate the **EINHIBIT** for the INHIBIT action on the ancillary trigger.
   (a) In the first case , there is a common dead time between every **FERA** crate, **INHIBIT** is sent to the ancillary trigger.
   (b) In the second case, only a local INHIBIT inhibs the **FVI**. **INHIBIT** is not sent to the ancillary trigger.
   (c) When **INHIBIT** is true **TRIG**, **FT** and **VAL** actions have not effect in the **FVI**, except for the counters (see counters).

Bit number	Value = 0	Value = 1
0	Normal mode	Test mode
1	Fast Trigger timeout disabled	Fast Trigger timeout enabled
2	Validation timeout disabled	Validation timeout enabled
3	Reject Off	Reject On if bit 0 set
4	FVI card Prom write disabled	FVI card Prom write enabled
5	REQ Off for FERA branch 1	REQ On for FERA branch 1 if bit 0 set
6	REQ Off for FERA branch 2	REQ On for FERA branch 2 if bit 0 set
7	No common dead time	Common dead time
8	Reset FVI and Ancillary trig Off	Reset FVI and Ancillary trig On
9	Reset GIR Off	RESET GIR On
10	No last card	Last card(use terminators)
11	Not used	not used
-	Not used	Not used
15	Not used	Not used

3.8.5 Module Status Register

The 16 bit **Status Register** must have the following bits to :

1. Know if the **FVI** is on **NORMAL** or **TEST** mode.
2. Know if each FIFO is **EMPTY** or not.
3. Know if each FIFO is at least **HALF-FULL** or not.
4. Know if the **FVI** has asserted **INHIBIT**.
5. Know if the **REJECT** is asserted or not.

Bit number	Value = 0	Value = 1
0	Normal mode	Test mode
1	FVI FIFO branch 1 not Empty	FVI FIFO branch 1 Empty
2	FVI FIFO branch 2 not Empty	FVI FIFO branch 2 Empty
3	FVI FIFO branch 1 not Half Full	FVI FIFO branch 1 Half Full
4	FVI FIFO branch 2 not Half Full	FVI FIFO branch 2 Half Full
5	FVI FIFO branch 1 not Full	FVI FIFO branch 1 Full
6	FVI FIFO branch 2 not Full	FVI FIFO branch 2 Full
7	Inhibit is not asserted	Inhibit is asserted
8	Reject is not asserted	Reject is asserted
9	Reset GIR Off	RESET GIR On
10	No last card	Last card(use terminators)
11	Not used	not used
-	Not used	Not used
15	Not used	Not used

The bit 7 and 8 of the status register are cleared after a read cycle. They are set again if the condition is true.

3.8.6 FERA Control Registers

FERACTL1 for FERA branch1 and **FERACTL2** for FERA branch 2 allow to clear all the ADCs and the selection of the FERA readout mode if the FERA ADCs provide a sequential or random readout.The **FERACTL1** register bit 2 allows to hold the data in the 2 **FERA** fifos for the branch 1 and the branch 2. When set to 1 the bit 3 of **FERACTL1** and **FERACTL2** set **REN** to **PASS** for the **FERA** branches 1 and 2.These bit avoid the use of a wire between **REN** and **PASS** if a **FERA** branch is not used.

Bit number	Value = 0	Value = 1
0	No Clear action on ADCs	Clear action on ADCs
1	Sequential readout of ADCs	Random readout of ADCs
2	NO hold data in FERA fifo	Hold FERA data in FERA fifo
3	FERA branch On	FERA branch OFF
4	Not used	Not used
-	-	-
15	Not used	Not used

3.8.7 FERA FIFOs

1. **FERAFIFO1** and **FERAFIFO2** are accessible in read and write mode if the **MCR** bit 0 is set.

2. **FERAFIFO1** and **FERAFIFO2** are not accessible in read and write mode if the **MCR** bit 0 is cleared. Only the **FERA** readout and the **GIR** readout can write

and read the FIFOs.

3.8.8 Logical inspection lines

A digital inspection line is disabled when the value of **DIGCSEL1** or **DIGCSEL2** is set to 0.
A digital inspection line is enabled when the value of **DIGCSEL1** or **DIGCSEL2** is set to 1.
The range values of **DIGPSEL1** or **DIGPSEL2** are :

Parameter	Data value(D3-D0)
WST1	0000
REN1	0001
GATE1	0010
WST2	0011
REN2	0100
GATE2	0101
RESET	0110
INHIBIT	0111
REJECT	1000
Ancillary TRIG	1001
spare	1010
-	-
spare	1111

3.8.9 Voltage inspection line

The voltage inspection line is disabled when the value of **VOLTCSEL** is set to 0.
The voltage inspection line is enabled when the value of **VOLTCSEL** is set to 1.
The range values of **VOLTPSEL** are :

Parameter	Data value(D3-D0)
VCC	0000
GND	0001
VEE	0010
VBB	0011
spare	0100
-	-
spare	1111

3.8.10 FVI configuration registers

The 8 bit PROM hold the Model reference for the **FVI**

Description	Size(byte)
Manufacturer ID	2
Card type	2
Serial number	2
Revision level	2
Spare	24

3.9 Modules interconnection

Since each **FVI** is identical and only One **FVI** must handshake with the Ancillary trigger, **LRESET** must be chained to **ERESET** and **LINHIBIT** must be chained to **EINHIBIT**. The **VXI LBUS30** and **LBUS31** are allocated for the **FVI** interconnection.

3.10 FVI overview

The figure 3.1 shows the global implementation of the **FVI**.

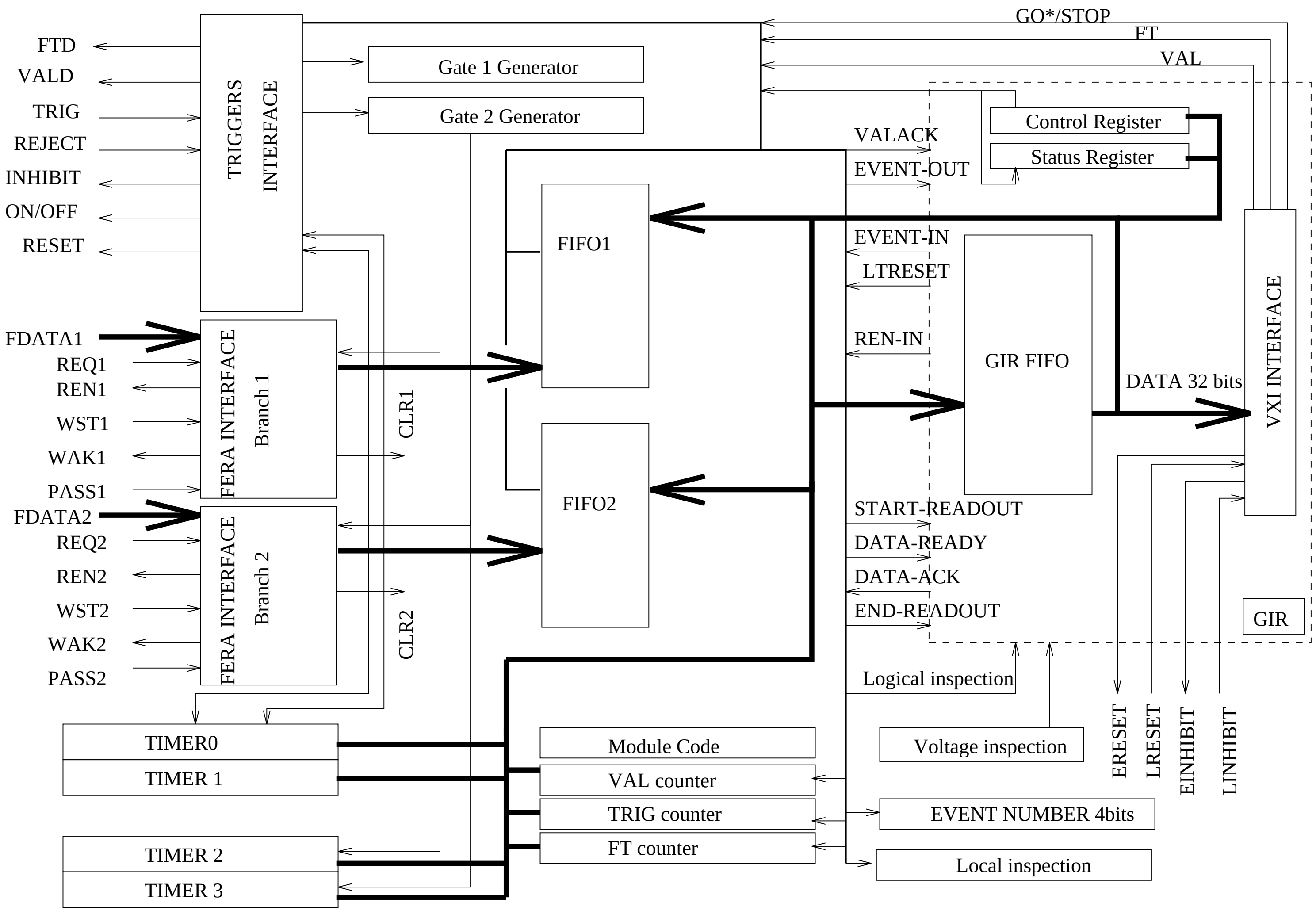

Figure 3.1: **FVI** overview