ISTITUTO NAZIONALE DI FISICA NUCLEARE

Laboratori Nazionali di Frascati

INFN-12-16/LNF
19th September 2012

TEST SYSTEM FOR THE GENERAL INTERFACE BOARDS

A. Balla, L. Iafolla

Abstract

The GIB boards (General Interface Boards) and the dedicated crate were designed and realized by a team of the SEA (Servizio Elettronica e Automazione) group of the research division. The main task of these boards is to interface the front-end electronics (FE or transition boards) dedicated to different detectors of KLOE-2 experiment with the higher stages of the acquisition system. The GIBs are equipped with many types of interfaces: optical link, RS232, USB, Ethernet and a 184-pins connector for the crate; the link between the GIBs and the front-end boards is done through the backplane of the crate. About 90 boards were produced and some tests were necessary to discard the broken ones. The toughest is the test of the pins for the crate: so we developed a dedicated system and software in order to accomplish it in a fast and sure way. We wrote this note to show how this system works in the case of a future production of new GIB boards.

Published by ***SIDS–Pubblicazioni***
Laboratori Nazionali di Frascati

The GIB board and the Crate

The GIB is based on a Virtex-4 FPGA which is programmed with different HDLs depending on the front-end electronics it has to manage. The used Virtex-4 model contains a PowerPC processor and it can implement a powerful embedded system that can control all the interfaces of the board.

The crate was designed to couple one GIB to the corresponding FE board and to supply them the power: it can host up to 20 couple GIB-FE (see Fig. 1).

The special shape of the crate lets to easily connect the FE boards with the GIBs: the latest are plugged on the front side, the others are plugged on the back side and they are connected each other through the backplane. The signals on the connector are driven and read by the FPGA through some buffers (ex. Texas Instruments lvc16245a, lvdm1676....) whose outputs are both differential and single-ended.

The connection to the higher levels of the acquisition system or to a console can be done in different ways: RS232, Ethernet, Optical link, USB.

GIB test: the hardware configuration

The most demanding test is the one on the pins of the crate connector: it could require many days to test all the 184 I/O-pins of the 90 GIBs. A simple hardware configuration can help to accomplish this test in a very fast way. In order to be able to plug a previously tested GIB in the FE side we modified its back-plane connector and we removed a rail from the crate (see Fig. 2): this GIB was our "master GIB" which we used to test all the others. The boards to be tested were plugged one by one on the GIB side of the crate in the slot corresponding to the one of the "master GIB".

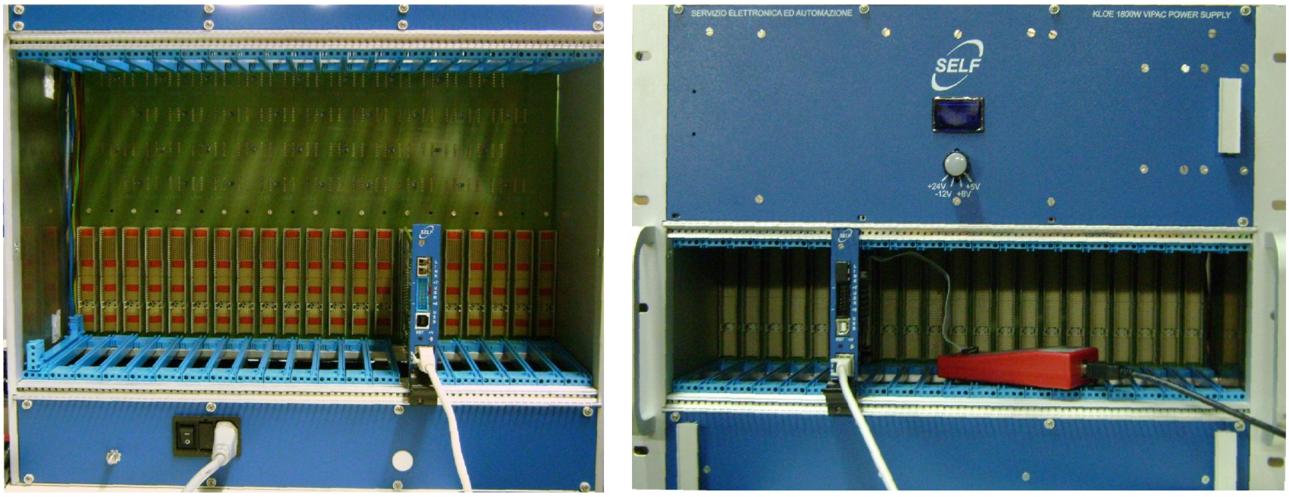
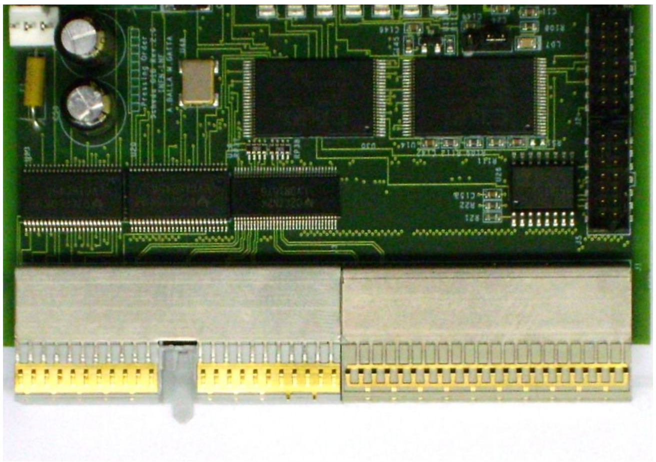

SERVIZIO ELETTRONICA ED AUTOMAZIONE
KLOE 1800W VIPAC POWER SUPPLY
SELF
+24V +5V
-12V +6V

Fig. 1. Back side (left) and front side (right) of the GIB dedicated crate. For the tests we plugged the “master GIB” on the backside and, one by one, the GIBs under test on the front side.

Unfortunately, plugging the GIB in this way does not connect each pin to the corresponding one of the other board; anyway, each differential couple is connected to a differential couple, each single-ended signal is connected to a single-ended signal, and the power supply pins are connected to the corresponding power supply lines: so, there is no danger of shortcuts.



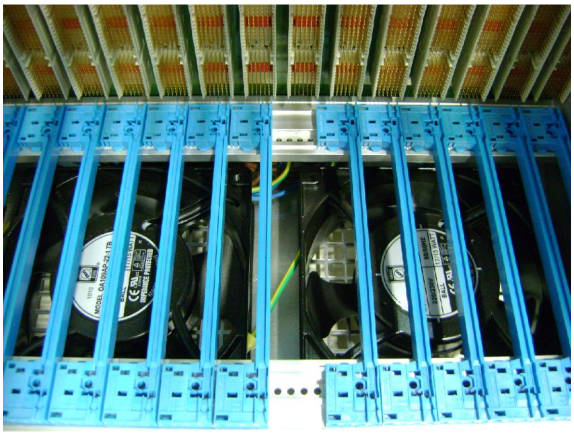


Fig. 2. Modified connector of the "master GIB" (left picture) and removed rail from the crate (right picture).

GIB test: FPGA programming

A dedicated embedded system to be implemented on the FPGA was designed in order to easily control the IO-pins.

Name	Address	Function
Dual Conf. Reg.	81420008	This register controls the tri-state buffers on the GIB
GPIO_tri-state(A)	83C1A004	This register controls the tri-state buffers in the FPGA for the pins IO_P/N(0 to 31)
GPIO_tri-state(B)	83C4D004	This register controls the tri-state buffers in the FPGA for the pins IO_P/N(32 to 63)
GPIO_tri-state(C)	83C4D00C	This register controls the tri-state buffers in the FPGA for the pins IO_P/N(64 to 71) and USER
GPIO_tri-state(D)	83C24004	This register controls the tri-state buffers in the FPGA for the pins IO_S(0 to 31)
GPIO_In-Out(A)	83C1A000	This register controls the input/output of the pins IO_P/N(0 to 31)
GPIO_In-Out (B)	83C4D000	This register controls the input/output of the pins IO_P/N(32 to 63)
GPIO_In-Out (C)	83C4D008	This register controls the input/output of the pins IO_P/N(64 to 71) and USER
GPIO_In-Out (D)	83C24000	This register controls the input/output of the pins IO_S(0 to 31)

Table 1. Address of the registers and their functions.

The project was obtained by modifying a previous one: what we need was just a set of registers to control the signals on the pins, some to control the tri-state buffers of the FPGA

IOBs (input-output block) and some to control the tri-state buffers which drive back-plane pins. This task can easily be accomplished with some GPIO (General Purpose Input/Output) IPs (Intellectual Property) that are provided by the XPS development software and can be implemented on the FPGA.

The Ethernet and RS232 interfaces can be used to pass the commands to the embedded system. We prefer Ethernet that let us to develop a software that can control 2 boards at the same time; the RS232 is useful for debugging in case the Ethernet connection doesn't work.

The final system is easy to control from the console with two commands.

1. To write: "w" & "register address" & "value";
2. to read: "r" & "register address".

For example, if you want to write "FFFFFFFF" on the register at the address "AAAAAAAA" you should write on the console: "wAAAAAAAAFFFFFFFF".

In Table 1 there is the address map; remember that each address and data is 32 bits long (8 hex). In the following sections a brief explanation of the function of each register is provided.

Dual Configuration Register

In this register only the 25 less significant bits are important; the others don't have any effects on the behavior of the board. These 25 bits are connected with the "INOUT" signals of GIB (see GIB schematics) which control the tri-state status of the buffers on the board.

The two most important configurations of this register are:

1) "01BC0000" for the input mode of all the buffers;
2) "0043FFFF" for the output mode of all the buffers.

GPIO Tri-State Registers

These registers control the status of the tri-state buffers inside the IOB (Input Output Block) of the FPGA. Each bit is associated to the corresponding bit of the GPIO_IN-OUT register.

The two most important configurations of these registers are:

1) "FFFFFFFF" for the output mode;
2) "00000000" for the input mode.

GPIO IN-OUT Registers

The GPIO IP has a simple architecture; the embedded system sees it just as a couple of registers with their addresses: the In-Out and the Tri-state registers. The Tri-state register is actually a register whose bits drive the buffers tri-state of the IOB: it can be set in "input mode" or in "output mode". The In-Out register behaves differently depending on the mode of the Tri-state.

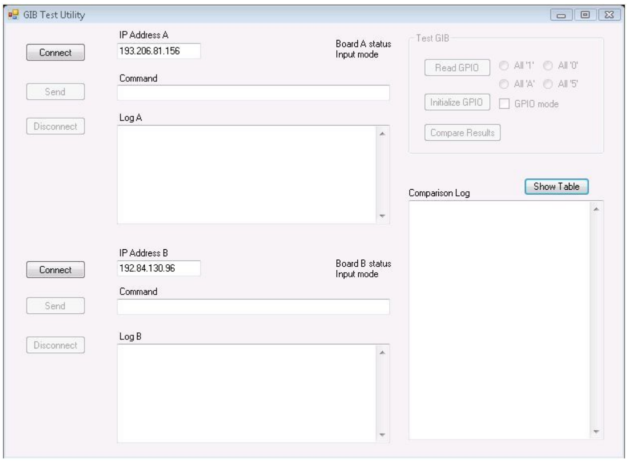
1. In "output mode" it behaves like a simple register whose bits drive the output pins: so you will read on it the same thing you wrote before;
2. in "input mode" you read the state of the input pins and, of course, you shouldn't write.

You can write the In-Out register while the Tri-state is in "input mode", but the value that you write will be available only when you switch back to the "output mode".

The association of the bits to the nets on the GIB is quite straight for the first two (A and B) and the last (D) In-Out register and just a little bit more complicated for the third (C) register: see the Table 3.

GIB test Utility

An application in Visual Basic was developed to make the tests in the fastest way: Fig. 3 shows a screenshot of the graphical interface.



IP Address A
193.206.81.156

Board A status
Input mode

IP Address B
192.84.130.96

Board B status
Input mode

Fig. 3. Graphical user interface of the application "GIB test Utility"

This software let to connect to the two GIBs under test and to write and read the signals from the pins very quickly. Below is the list of buttons and their functions.

Command	Function
Connect	Push this button to connect to the IP whose address is written in the text box on its right
IP address A/B (text-box)	Write in these text-boxes the addresses of the IP you want to connect to
Send	Enabled only after connection – Push this button to send the command written in the text-box on its right
Command (text-box)	Write in this box the command you want to send
Disconnect	Push this button to disconnect from the IP
Log A/B	In this text box will appear all the messages sent and received from the IP
Read GPIO	Enabled only if connected to both the boards – Push this button to read the status of all GPIO In-Out registers
Radio buttons "All 1/0/A/5"	Enabled only if connected to both the boards – Check one of these buttons to write on all the GPIO In-Out registers all 1/0/A/5
Initialize GPIO	Enabled only if connected to both the boards – Push this button to modify the tri-state buffers in such a way that all the pins of one board are in input mode and the pins of the other board are in output mode. You can read the status of the each board in the dedicated sign.
GPIO mode check button	Enabled only if connected to both the boards – Check this button to switch the input/output mode of the two boards.
Compare Results	Enabled only if connected to both the boards – Push this button after having pushed the "Read GPIO" button to compare the output of the board in "Output mode" with the output of a properly working board.
Comparison Log	In this text-box will appear the data read from the GIB in input mode to be compared with the expected ones..
Show Table	Open a window to access the table of signals mapping (see Fig. 4).

Table 2. Commands and functions of the application.

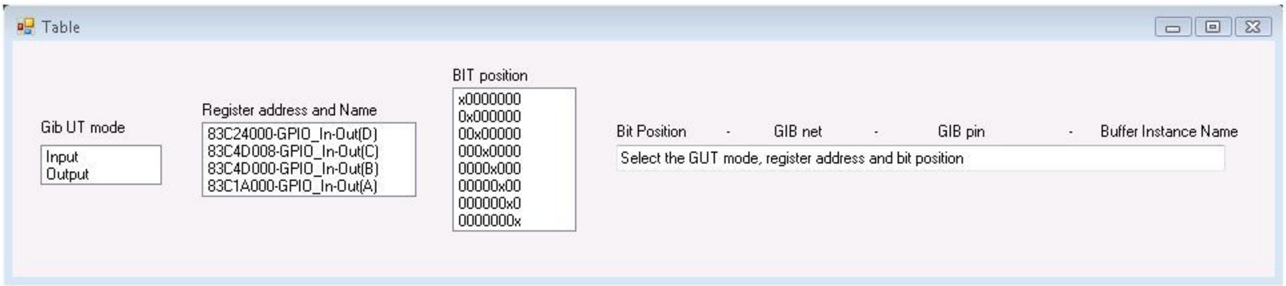
Follow these steps to execute a test:

1. plug the “GIB under test” in the crate at the slot corresponding to the "master GIB";
2. connect the FPGA programmer, the Ethernet cable and switch on the power;
3. program the FPGA with the dedicated firmware (HDL and software);
4. start the GIB test Utility;
5. choose the right IP addresses;
6. click "Connect" for both the GIBs (the "Test GIB" box will become active);
7. click "Initialize" to set the tri-state buffers (wait few seconds before moving to the next step);
8. check one of the radio button (wait few seconds before moving to the next step);
9. click "Read GPIO" (LOG A/B will show the outputs but only the one in "input mode" is important, wait few seconds before moving to the next step);
10. click "Compare results" to see if one bit has not the value it should have;
11. if the last write in the Comparison Log is "Test Ok", repeat the steps from the eighth for all the radio buttons and then repeat again the tests after switching the "GPIO mode";
12. if all tests succeed, the GIB has no fault and the procedure is finish, otherwise find which bit doesn't work and use "Show Table" window (see Fig. 4) to go up to the broken chip/net on the board.

Register Name	Address	Bit	GIB net	GIB pin	Buffer Instance Name
GPIO_In-Out(D)	83C1A000	31 downto 24	IO(103 downto 96)	IO_S(31 downto 24)	U20
		23 downto 16	IO(95 downto 88)	IO_S(23 downto 16)	
		15 downto 8	IO(87 downto 80)	IO_S(15 downto 8)	U23
		7 downto 0	IO(79 downto 72)	IO_S(7 downto 0)	
GPIO_In-Out(C)	83C4D008	11 downto 4	IO(71 downto 64)	IO_N/P(71 downto 64)	U17
		3 downto 0	User(3 downto 0)	User_N/P(3 downto 0)	U16
GPIO_In-Out(B)	83C4D000	31 downto 24	IO(63 downto 56)	IO_N/P(63 downto 56)	U18
		23 downto 16	IO(55 downto 48)	IO_N/P(55 downto 48)	
		15 downto 8	IO(47 downto 40)	IO_N/P(47 downto 40)	U19
		7 downto 0	IO(39 downto 32)	IO_N/P(39 downto 32)	
GPIO_In-Out(A)	83C1A000	31 downto 24	IO(31 downto 24)	IO_N/P(31 downto 24)	U21
		23 downto 16	IO(23 downto 16)	IO_N/P(23 downto 16)	
		15 downto 8	IO(15 downto 8)	IO_N/P(15 downto 8)	U22
		7 downto 0	IO(7 downto 0)	IO_N/P(7 downto 0)	

Table 3. Mapping between Registers addresses, bits and names of the nets and pins on the board.

Also Table 3 can be used to find the net/chip which doesn’t work: this bit/signal map must be used taking in account that signals of one board are not connected to the corresponding ones of the other board. The association is not straight and you should refer to the schematic of the GIB: use "show table" function and select the mode in which is working the Gib Under Test to avoid this.



Table

Gib UT mode
Input
Output

Register address and Name
83C24000-GPIO_In-Out(D)
83C4D008-GPIO_In-Out(C)
83C4D000-GPIO_In-Out(B)
83C1A000-GPIO_In-Out(A)

BIT position
x0000000
0x000000
00x00000
000x0000
0000x000
00000x00
000000x0
0000000x

Bit Position - GIB net - GIB pin - Buffer Instance Name
Select the GUT mode, register address and bit position

Fig. 4. This is a snapshot of the window that pops up by clicking “Show Table” in the main program. It is possible to go up to the net/pin and to the buffer that fails by selecting the mode of the GIB under test, the register and the position of the bit which is not coherent.

Optical link test

The optical links were also tested: we did not develop any system to do this it, we just used the existing ones for "Qcalt" or "Inner tracker" detectors. The test can be done by sending a defined pattern on the transmitter (Tx) and comparing the input of the receiver (Rx) with the same pattern: so we can short the Tx and the Rx of the same GIB board or we can use two different boards (the master, which remains always the same, and a slave) that can be plugged both on the same side of the crate. Using two different GIBs is a more exhaustive method because the clocks of the Rx and of the TX are not the same and the synchronization system is tested under more realistic conditions.

Tests results

This system was used to test almost all the GIB boards produced (about 90) and thanks to it all the work was accomplished in only 2-3 days; much less than it would have need using an oscilloscope or a multimeter. About the 5% of the GIBs didn't pass the IO-pin test; the problems were focused on 3 chips: U23 (3 times), U16 (3 times) and U20 (once). No faults were detected for the optical links.

The software will be shared in the section “Applicazioni” of this web page: http://www.lnf.infn.it/~balla/; the HDL files are shared in the following AFS area of the SEA group: /afs/lnf.infn.it/project/sea/data/CDS_SITE/Documenti/Xilinx/Ver_12.x.