A development board based on the GR740 processor.

GR-CPCI-GR740

USER MANUAL

RELEASED AUGUST 2023

GR-CPCI-GR740
Quick Start Guide

Frontgrade Gaisler AB | Kungsgatan 12 | SE-411 19 | Göteborg | Sweden | frontgrade.com/gaisler

Table of Contents

1. Introduction

1.1. Overview

This document is a quick start guide for the GR-CPCI-GR740 CompactPCI Development Board.

The purpose of this document is to get users quickly started using the board.

For a complete description of the board please refer to the GR-CPCI-GR740 Development Board User's Manual.

The GR740 system-on-chip is described in the GR740 Data sheet and User's Manual.

This quick start guide does not contain as many technical details and is instead how-to oriented. However, to make the most of the guide the user should have glanced through the aforementioned documents and should ideally also be familiar with the GRMON debug monitor.

Information in this document applies to GR-CPCI-GR740 Revision 1.2 or later. Please contact support@gaisler.com for technical questions and for document versions applicable to earlier board revisions. The GR-CPCI-GR740 data package and this document (including possibly newer revisions) are available from the GR-CPCI-GR740 product page at https://www.gaisler.com.

1.2. Handling

ATTENTION : OBSERVE PRECAUTIONS FOR
HANDLING ELECTROSTATIC SENSITIVE DEVICES

This unit contains sensitive electronic components which can be damaged by Electrostatic Discharges (ESD). When handling or installing the unit observe appropriate precautions and ESD safe practices.

When not in use, store the unit in an electrostatic protective container or bag.

When connecting/disconnecting cables, ensure that the unit is in an un-powered state.

This equipment has SpW ports that use Low Voltage Differential Signalling (LVDS) which has limited common mode voltage protection. Please refer to the user’s manual for instructions on how to ensure that the grounds of equipment are connected together when using SpaceWire.

1.3. References

Table 1.1. References

RD-1	GR-CPCI-GR740 Development Board User's Manual
RD-2	GR740 Data sheet and User's Manual [https://www.gaisler.com/doc/gr740/GR740-UM-DS-2-3.pdf]
RD-4	GRMON User's Manual [https://www.gaisler.com/doc/grmon3.pdf]
RD-6	RTEMS homepage [https://www.rtems.org]
RD-7	LEON/ERC32 RTEMS Cross Compilation System (RCC) [https://www.gaisler.com/index.php/products/operating-systems/rtems]
RD-8	RCC User's manual [https://gaisler.com/anonftp/rcc/doc]
RD-9	Frontgrade Gaisler RTEMS driver documentation [https://gaisler.com/anonftp/rcc/doc]
RD-10	Bare C Cross-Compilation System [https://www.gaisler.com/index.php/products/operating-systems/bcc]

RD-11	BCC User's Manual [https://www.gaisler.com/doc/bcc2.pdf]
RD-12	VxWorks 7 SPARC architectural port and BSP [https://www.gaisler.com/index.php/products/operating-systems/vxworks-7]
RD-13	MKPROM2 User Manual [https://gaisler.com/doc/mkprom.pdf]

The referenced documents can be downloaded from https://www.gaisler.com.

2. Board Configuration

2.1. Overview

The primary sources of information are the GR-CPCI-GR740 Development Board User's Manual and the GR740 Data sheet and User's Manual. Before start using the GR-CPCI-GR740, clock sources have to be installed, bootstrap signals need to be set correctly and the desired interfaces have to be enabled. The GR740 shares some of the PROM/IO and SDRAM pins due to a limited number of pins. For that reason, the pin multiplexing has to be set up depending on the desired interfaces and memory configuration.

2.2. Default configuration

This guide provides a default configuration that uses flash PROM as boot memory and 96-bit SDRAM. The debug links are a) JTAG over FTDI and b) Ethernet. As general I/O, UART0 and UART1 are used. The complete default configuration can be found in Appendix A. If this is your first time using the GR-CPCI-GR740, please use this configuration as a starting point.

Default configuration

To achieve the default configuration please follow the instructions on each box note like this one.

2.3. Clocking

The default configuration of the board uses 50 MHz oscillators in the X2 (MEMCLK) and X4 (SYSCLK) sockets. If the SpaceWire interfaces are used then a 50 MHz oscillator must be present in the X1 (SPWCLK) socket. Also, an external clock source is required for the PCI interface, which can operate at 33 or 66 MHz, either from the PCI backplane or from an oscillator in socket X3 (PCICLK). Please refer to GR-CPCI-GR740 Development Board User's Manual for instructions on how to configure the board to function in a PCI system.

The internal frequencies for system, memory and spacewire depend on the oscillators installed on the board and on the PLL bypass[0:2] settings set on the front panel (FP-S3 switches 3, 4 and 5), as shown in Table 2.1 and Table 2.2 (Default configuration highlighted in grey). The default configuration with all clocks installed uses all PLLs enabled, as shown in the table. When changing settings, JP24 also needs to be set as shown in the table (see section *FPGA for PCI Arbiter & Versaclock Controller* in GR-CPCI-GR740 Development Board User's Manual for details). The clksel=0 options are intended to allow generating both system and memory clock with a single PLL but it multiplies the clock by 1 in default PLL configuration and is therefore not usable. It is recommended to keep clksel at the default (1) setting.

Table 2.1. System and Memory clock configuration

Bypass[0]	Bypass[1]	clksel	JP24 1-2	JP24 3-4	System clock	SDRAM clock	PLLs enabled
0	0	1	open	open	5xSYS=250	2xMEM=100	Sys, Mem
1	0	1	open	open	1xSYS=50	2xMEM=100	Mem
0	1	1	closed	closed	5xSYS=250	1xMEM=50	Sys
1	1	1	closed	closed	1xSYS=50	1xMEM=50	-
1	don't care	0	closed	closed	1xSYS=50	1xSYS=50	-

Table 2.2. Spacewire clock configuration

Bypass[2]	SpW clock	PLLs enabled
0	8xSPW=400	SpW
1	1xSPW=50	-

Default configuration

The default configuration of the board uses 50 MHz oscillators installed on X1, X2 and X4 for SPWCLK, MEMCLK and SYSCLK respectively; and a 33 MHz oscillator in X3 for PCICLK.

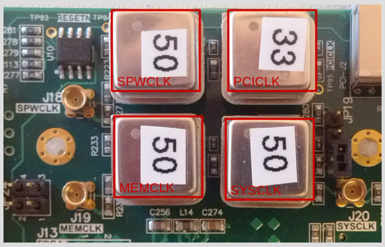

Figure 2.1. GR-CPCI-GR740 default configuration as delivered

PLLs are enabled, i.e. bypass signals are disabled, which means that DIP switches FP-S3-3, FP-S3-4, FP-S3-5 are in CLOSED position to enable PLLs (see Figure 2.2). The result is System clock at 250 MHz, SDRAM at 100 MHz, Spacewire at 400 MHz and PCI at 33 MHz.

By default, grmon(version 3.2.2 or later) sets up the srdam timing parameters such that the memory works both at 50MHz and 100MHz SDRAM clock. However, to have the correct performance, user need to set startup option -sdfreq <*mhz*>. Please refer section Section 4.2 for grmon command formats for different debug link connections.

2.4. Bootstrap Signals

Bootstrap signals configure the chip on reset and are listed in section *Bootstrap signals* of GR740 Data sheet and User's Manual. Some of these signals can be controlled on the GR-CPCI-GR740 via the DIP switches FP-S1, FP-S2, FP-S3 and S1. The bootstrap signals that are mapped to the general purpose I/O lines (DIP switch FP-S1 and FP-S2) control settings such as the reset address for the Ethernet debug communications link (EDCL), routing of EDCL traffic, boot-PROM width and PROM EDAC enable. The bootstrap signals controlled via DIP switch FP-S3 and S1 have a larger impact on the full system behaviour:

- The BREAK signal (S1) controls if the first processor will start execution after system reset. If it is on the right switch position, then it is not activated, meaning that the execution will start after system reset. On the left switch position, break is activated and the execution will not start.
- The DSU enable signal (FP-S3-1) controls if the design's Debug Support Unit is enabled and also if the debug communication links are active. DIP switch S3-1 must be set to OPEN, i.e. DSU enabled, to connect to the board using the GRMON3 debug monitor.
- The MEM_CLKSEL signal (FP-S3-2) selects the clocks used for the SDRAM memory interface and the on-chip buses. If the DIP switch S3-2 is OPEN the source for the memory clock is the MEM_EXTCLOCK clock input, otherwise the memory clock and the system clock have the same source.
- The BYPASS_PLL signals (FP-S3-3, FP-S3-4, FP-S3-5) are explained in Section 2.3.
- The PLL_IGNLOCK signal (FP-S3-6) is explained in Section 2.3.
- The ETHCLK signal (FP-S3-7) is explained in Section 2.6.2.
- The WDEN signal (FP-S3-8) controls if a watchdog timeout in the GR740 will trigger a board reset. This switch should be set to OPEN in order to disconnect the watchdog from the reset circuit.

Some bootstrap signals are automatically set up by installing the configuration plug on J21, J22 or J23. See Section 2.5:

- The MEM_IFWIDTH signal selects the width of the primary memory interface.
- The PCI_MODE signal selects if the top-half of the SDRAM interface should be used for the PCI controller (HIGH) or Ethernet port 1 (LOW) when the SDRAM is not in full width.

The GPIO bootstrap signals are as follows:

- The GPIO[0:5] signals (FP-S1-1 to FP-S1-6) control EDCL IP address. All pulled up disable EDCL on both ETH0 and ETH1.
- The GPIO[6:7] signals (FP-S1-7 and FP-S1-8) control Spacewire router distributed interrupt configuration.
- The GPIO[8:9] signals (FP-S2-1 and FP-S2-2) control EDCL routing. Pulled up route EDCL traffic to the debug bus.

- The GPIO[10] signal (FP-S2-3) controls PROM width. Pulled down selects 8-bit, otherwise 16-bit.
- The GPIO[11] signal (FP-S2-4) controls Spacewire router clock gating settings. Pulled up disables the Spacewire router on the clock gating unit.
- The GPIO[12:13] signals (FP-S2-5 and FP-S2-6) set Spacewire router's instance ID.
- The GPIO[14] signal (FP-S2-7) controls PROM EDAC. Pulled down disables PROM EDAC, otherwise enabled.
- The GPIO[15] signal (FP-S2-8) controls PROM/IO pin multiplexing (after reset). Pulled down enables PROM interface, otherwise alternative functions are enabled. Note that this can be changed later by software (see Section 2.5).

Default configuration

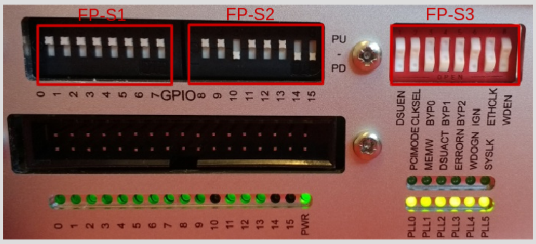
FP-S1
FP-S2
FP-S3
PU
-
PD
0 1 2 3 4 5 6 7 GPIO 8 9 10 11 12 13 14 15
OPEN
DSUEN
PCIMODE CLKSEL
MEMW BYP0
DSUACT BYP1
ERRORN BYP2
WDOGN IGN
SYSLK ETHCLK
WDEN
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 PWR
PLL0 PLL1 PLL2 PLL3 PLL4 PLL5

Figure 2.2. GR-CPCI-GR740 default configuration as delivered

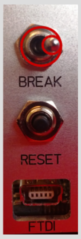

Figure 2.3. GR-CPCI-GR740 default configuration as delivered

The default configuration of the board uses:

FP-S1-1 to FP-S1-6 pulled up to disable EDCL on ETH0 and ETH1.

FP-S2-1 and FP-S2-2 pulled up to route EDCL traffic through DSU bus. FP-S2-3 pulled down to set PROM width to 8-bit. FP-S2-4 pulled up to disable the Spacewire router in the clock gating unit. FP-S2-7 and FP-S2-8 pulled down to disable PROM EDAC and enable PROM mode.

FP-S3-1 Open to enable DSU. FP-S3-2 Open to have external memory clock. FP-S3-3, FP-S3-4, FP-S3-5 Closed to enable PLLs and FP-S3-6 Open to ignore PLL lock. FP-S3-7 to Closed to enable 100Mbit ethernet mode. FP-S3-8 Open to disconnect watchdog from the reset.

S1 switch on the right position to disable BREAK signal.

2.5. Pin multiplexing

2.5.1. PROM/IO, UART, CAN, MIL-STD-1553B, Spacewire, general purpose I/O

The GR740 shares some of the PROM/IO pins due to a limited number of pins. There are three configurations available: a) PROM/IO; b) UART, CAN, MIL-STD-1553B and Spacewire debug interfaces; and c) general-purpose I/O. See section *Pin multiplexing* of the GR740 Data sheet and User's Manual.

There are two things to take into consideration when configuring the pin multiplexing: The configuration on the GR-CPCI-GR740 board and the configuration on the GR740 device.

First, to configure the GR-CPCI-GR740 board pin multiplexing, we need to set JP11 jumpers in the wanted configuration. Each JP11 jumper chooses each individual pin configuration as shown in Table 2.3(Default configuration highlighted in grey). Set all JP11 jumpers on position *AB* if full PROM/IO mode (a) is used or position *CD* for peripheral mode (i.e. c and d). See section *PROMIO / Interface configuration* of the GR-CPCI-GR740 Development Board User's Manual.

Second, to configure the GR740 set GPIO[15] to LOW for full PROM/IO mode (a) or HIGH for peripheral mode (c). GPIO[15] controls the reset value of the FTMEN register in the GR740, which belongs to the GRGPRBANK core (mapped in memory at 0xFFA0B000). FTMEN configures each pin between configurations a) and b) (another register, called ALTEN is used for configuration c)). Table 2.3 shows the configuration for each bit. GPIO[15] LOW puts all bits to 1 and HIGH to 0. See section *PROMIO / Interface configuration* of the GR-CPCI-GR740 Development Board User's Manual and *Register Bank For I/O and PLL configuration registers* of the GR740 Data sheet and User's Manual.

It is also possible to create custom muxing configurations in between these three configurations, such as using the 8-bit PROM/IO along with the UART0 and UART1, which is the chosen default configuration. In such a configuration, the two most significant address pins and the two most significant data pins of the PROM/IO are used for the UARTs. To set up such custom muxing configurations, three steps are required: First, set the corresponding JP11 jumpers in the board (see Table 2.3). In our case, means that all JP11 jumpers are in position *AB*, or PROMIO mode, except jumpers 1,2, 13 and 14 in position *CD* (UARTs pins). Note that the signals PROMIO_DATA[0:7] (JP11 13 and 14) are used only in the 16-bit PROM mode. The 8-bit PROM interface uses the signals PROMIO_DATA[8:15]. This means that the PROM/IO can only be used in 8-bit mode, up to 16 MiB instead of 64 MiB, when using the UART0 and UART1 along with the PROM, because the MSB address pins and the MSB data pins of the PROM/IO are used for the UARTs. Second, GPIO[15] needs to be LOW to set FTMEN and start in PROM mode. Third, software needs to change the configuration of the individual bits in FTMEN from a) PROM to b) peripheral mode. This means changing bits 8, 9, 20 and 21 to 0, that is writing 0xFFCFF to FTMEN(0xFFA0B000). This can be done on the initialization boot code, as shown in Chapter 5 See section *PROMIO / Interface configuration* of the GR-CPCI-GR740 Development Board User's Manual and *Register Bank For I/O and PLL configuration registers* of the GR740 Data sheet and User's Manual.

Table 2.3. Jumper JP11 and FTMEN Register configurations

JP11 jumper	FTMEN bit	JP11 jumper at AB / FTMEN bit at 1	JP11 jumper at CD / FTMEN bit at 0
1	21	PROMIO_ADDR27	UART_TXD0
2	20	PROMIO_ADDR26	UART_TXD1
3	19	PROMIO_ADDR25	1553TXA
4	18	PROMIO_ADDR24	1553TXNA
5	17	PROMIO_ADDR23	1553RXENA
6	16	PROMIO_ADDR22	1553TXB
7	15	PROMIO_ADDR21	1553TXNB
8	14	PROMIO_ADDR20	1553RXENB
9	13	PROMIO_ADDR19	SPWDCL_DBG_TXD
10	12	PROMIO_ADDR18	SPWDCL_DBG_TXS
11	11	PROMIO_ADDR17	UART_RTSN0
12	10	PROMIO_ADDR16	UART_RTSN1
13	9	PROMIO_DATA7[1]	UART_RXD0
14	8	PROMIO_DATA6[1]	UART_RXD1
15	7	PROMIO_DATA5[1]	CAN_RX0
16	6	PROMIO_DATA4[1]	CAN_RX1
17	5	PROMIO_DATA3[1]	1553RXA
18	4	PROMIO_DATA2[1]	1553RXNA
19	3	PROMIO_DATA1[1]	1553RXB
20	2	PROMIO_DATA0[1]	1553RXNB
21	1	PROMIO_CEN1	CAN_TX0

JP11 jumper	FTMEN bit	JP11 jumper at AB / FTMEN bit at 1	JP11 jumper at CD / FTMEN bit at 0
22	0	PROMIO_IOSN	CAN_TX1

[1] PROMIO_DATA bits [0:7] are only used in 16-bit PROM mode. PROMIO_DATA[8:15] are used for 8-bit and 16-bit PROM mode.

Default configuration

The default configuration of the board consist of a custom multiplexing configuration using PROM/IO and UARTs 0 and 1. To set the default configuration, set all JP11 jumpers on position AB except JP11-1, JP11-2, JP11-13 and JP11-14 in position CD (see Figure 2.4). FP-S2-8 (GPIO[15]) is Closed (see Figure 2.2) to start in PROM/IO mode and the software has to set up the register FTMEN(0xFFA0B000) to 0x000FFCFF to activate the UARTs pins. For instance, this can be done on the initialization boot code, as shown in Chapter 5 or in GRMON3 as shown in Chapter 4. See section *Pin multiplexing control* of the GR740 Data sheet and User's Manual.

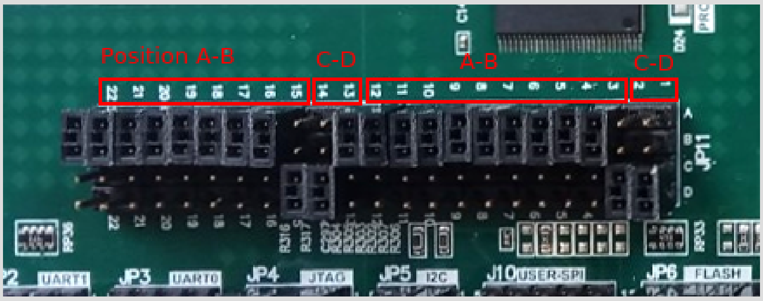

Figure 2.4. GR-CPCI-GR740 default configuration as delivered

2.5.2. SDRAM, PCI, Ethernet port 1

The top half of the SDRAM interface shares pins with PCI and Ethernet port 1. There are three configurations, a) 96-bit memory interface; b) 48-bit memory interface and PCI; and c) 48-bit memory interface and Ethernet port 1. These configurations are selected by installing the configuration plug on the bottom side of the GR-CPCI-GR740 in J21 (b), J22 (c) or J23 (a) depending on the desired configuration and cannot be reprogrammed in software. Please note that for configuration a) 96-bit interface, 2 SODIMM modules are required. See section *SDRAM Memory Interface configuration* of the GR-CPCI-GR740 Development Board User's Manual.

Default configuration

The default configuration of the board uses the full 96-bit SDRAM width, in which the configuration plug is installed on the bottom side of the GR-CPCI-GR740 in J23.

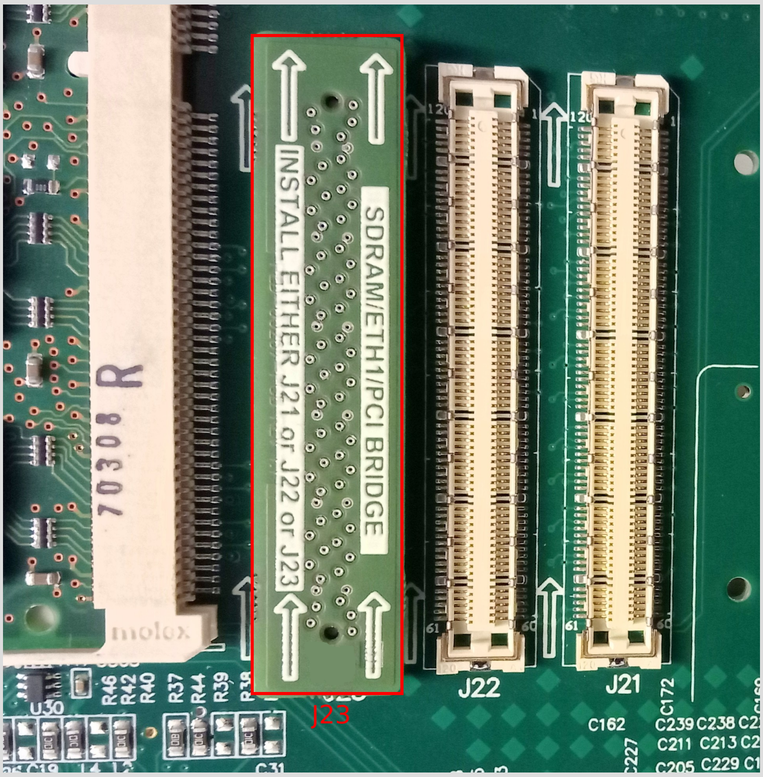

Figure 2.5. GR-CPCI-GR740 default configuration as delivered

2.6. Interfaces

This section describes how to set up the main different interfaces of the GR-CPCI-GR740. If you are not interested on a specific interface not used on the default configuration, skip that part.

2.6.1. JTAG FTDI

A FTDI FT423HL chip provides a JTAG to USB conversion supported by GRMON3 (see Chapter 4). Jumpers JP4 have to be installed to use JTAG to USB. Please see section *FTDI Serial to USB Interface* of the GR-CPCI-GR740 Development Board User's Manual.

Default configuration

The default configuration uses JTAG to USB conversion to connect to the board using GRMON3 by having all JP4 jumpers installed.

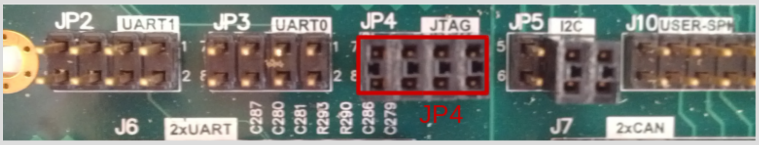

Figure 2.6. GR-CPCI-GR740 default configuration as delivered

2.6.2. Ethernet

There are two ethernet ports available in the GR-CPCI-GR740, port 0 and port 1. To use port 1, make sure that the pin multiplexing is configured to use ETH1, as explained in Section 2.5. FP-S3-7 chooses between gigabit mode and 100 Mbps. To use the Ethernet interfaces for the EDCL Debug link (see Chapter 4), it is necessary to appropriately set the GPIO signals at power-up/reset. GPIO[0..5] (FP-S1-1 to FP-S1-6) set the least significant address bits of the IP address of each port and also disable EDCL if all of them are Open. GPIO[8] and GPIO[9] (FP-S2-1 and FP-S2-2) set the routing of the ethernet traffic on the Debug AHB bus, required to use EDCL, for port 0 and 1 respectively. Please see section *Ethernet Interface* of the GR-CPCI-GR740 Development Board User's Manual.

Default configuration

The default configuration uses only ethernet port 0, prepared to use EDCL by setting FP-S2-1 and FP-S2-2 Open (see Figure 2.2). But disabled by default to avoid conflicts when connecting the board by setting FP-S1-1 to FP-S1-6 to Open (see Figure 2.2). Please note that user should either change FP-S1-1 to FP-S1-6 to get a valid EDCL IP address at startup or set up any IP by using GRMON3 in order to use EDCL (see Chapter 4). Also FP-S3-7 should be adjusted according to the desired link speed.

2.6.3. PROM Parallel flash

Make sure that the pin multiplexing is configured to use PROM/IO, as explained in Section 2.5. The JS28F640J3 flash provides 8 MiB of non-volatile memory. This device can be configured for 8-bit or 16-bit mode. Please note that for 16-bit mode, the pin multiplexing has to be on full PROM/IO mode (a). GPIO[15] should also reflect this configuration. There is a hardware write-protection of the flash by installing jumper JP6a. The flash can be disconnected by removing jumper JP6b. Please see section *PROMIO / Parallel flash* of the GR-CPCI-GR740 Development Board User's Manual.

Default configuration

The default configuration uses 8-bit PROM with EDAC disabled. Jumpers JP6 are all connected except JP6a to remove write-protection (see Figure 2.7). DIP switches FP-S2-3 and FP-S2-7 Closed to enable 8-bit PROM with EDAC disabled (see Figure 2.2).

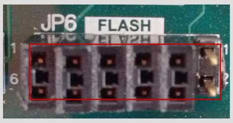

Figure 2.7. GR-CPCI-GR740 default configuration as delivered

2.6.4. UARTs

Make sure that the pin multiplexing is configured to use UARTs, as explained in Section 2.5. The UARTs of the GR740 can either be connected to RS232 transceivers and the DSUB-9 connectors on the accessory PCB, or, they can be connected to the FTDI USB to serial converter. This configuration is chosen with jumpers JP2 and JP3 on the board. Please see section *Serial Interface (RS232)* of the GR-CPCI-GR740 Development Board User's Manual.

Default configuration

The default configuration of the board uses the DSUB-9 connectors by having all JP2 and JP3 jumpers disconnected.

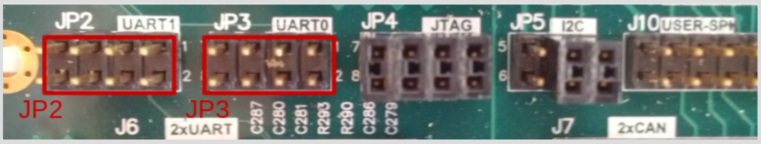

Figure 2.8. GR-CPCI-GR740 default configuration as delivered

2.6.5. CAN

Make sure that the pin multiplexing is configured to use CAN, as explained in Section 2.5. The CAN interfaces of the GR740 can be connected to the DSUB-9 connectors on the accessory PCB. Make sure that the correct bus termination is chosen with jumpers JP1 and JP2 on the accessory PCB. Please see section *CAN 2.0 Interfaces* of the GR-CPCI-GR740 Development Board User's Manual.

2.6.6. MIL-STD-1553B

Make sure that the pin multiplexing is configured to use MIL-STD-1553B, as explained in Section 2.5. The MIL-STD-1553B interfaces of the GR740 can be connected to the DSUB-9 connector on the accessory PCB. Make sure that the correct bus termination is chosen with jumpers JP3 on the accessory PCB. Please see section *MIL-STD-1553 Interface* of the GR-CPCI-GR740 Development Board User's Manual.

2.6.7. Spacewire

There are nine Spacewire (SPW) interfaces provided by the GR740 that can be connected on the front panel with standard MDM9S connectors. One of them is a SPW Debug interface (DSU),which can be used by installing jumpers on positions 1-2 and 3-4 of JP7.

Make sure that the Spacewire router is not clock-gated off by the clock gating unit in order to use the Spacewire ports. The default configuration of the board boots with the Spacewire router clock-gated off. Set GPIO[11] or FP-S2-4 to low to change this and enable the Spacewire router by default.

To be able to boot via SPW debug interface, pin multiplexing also needs to be configured properly. Pull up the front panel switch FP-S2-8 to set the GPIO[15] signal as HIGH. This will configure the FTMEN register reset value such that alternative function is routed to the shared pins. Install jumpers on C-D on JP11 positions 9 and 10 to enable the SPWDCL TX signals.

Please see section *Spacewire (LVDS) Interfaces* of the GR-CPCI-GR740 Development Board User's Manual.

2.6.8. PCI

Make sure that the pin multiplexing is configured to use PCI, as explained in Section 2.5. The PCI interface of the GR740, can be connected directly to backplane CPCI connector or via a PCI-PCI Bridge (Texas Instruments, PCI2060). There are four different types of configuration: 1) Host with PCI bridge; 2) Peripheral with PCI bridge; 3) Host without PCI bridge; and 4) Peripheral without PCI bridge. All these configurations are automatically set up by connecting the different plug-on PCBs provided for each type of PCI configuration. The plug-on PCB's 24, 25, 26 and 27 have to be plugged on the J24, J25, J26 and J27 on the back of the GR-CPCI-GR740 respectively. Please see section *PCI Interface* of the GR-CPCI-GR740 Development Board User's Manual.

The complete default configuration can be found in Appendix A. The referenced documents can be downloaded from https://www.gaisler.com/gr740.

3. Comments on System-on-Chip Design

3.1. Overview

The goal of this section is to summarize differences with the GR740 device compared to other contemporary LEON systems and what these differences mean to users. The GR740 Data sheet and User's Manual has a section named Technical notes that contain additional information on this topic specific for the GR740.

3.2. Building Operating Systems for GR740

Operating systems must take into account that the RAM starts at 0x0 in the GR740 design. This is typically done by specifying special build options. Operating systems distributed by Frontgrade Gaisler have been extended to support linking to address 0x0 instead of the 0x40000000 address that is traditionally used in LEON systems. Please refer to the operating system documentation for additional information.

Software must also take into account that in the GR740 the peripheral units are connected through an AHB-to-AHB bridge. This has no impact for normal memory accesses but existing software may not support AMBA plug and play scanning over the AHB-to-AHB bridges. All recent versions of operating systems distributed by Frontgrade Gaisler will correctly detect the peripheral devices in GR740. Support for recursive plug and play scanning over bridges is present in BCC version 1.0.41 (software compiled by earlier versions of BCC need to have been built using the `-qambapp` flag), RTEMS version 4.10, VxWorks 6.3/6.5/6.7 release 1.0.3, MKPROM2 version 2.0.56, and later versions of these software releases available from Frontgrade Gaisler.

3.3. Building Applications for GR740

No special consideration needs to be taken for applications that run on top of an operating system.

Special flags must be specified when using the Bare-C Compiler (BCC) toolchain available from Frontgrade Gaisler. In order to link the application to address 0 the flags `-Wl,-msparcleon0` must be specified. In order to enable plug and play scanning over AHB-to-AHB bridges, the `-qambapp` must be specified. The functionality enabled by `-qambapp` is enabled by default starting with BCC version 1.0.41. With version 1.0.41 or higher of BCC the following call would compile a hello world application: **sparc-elf-gcc -Wall --Wl,-msparcleon0 hello.c -o hello**

3.4. Running binaries linked to address 0x40000000

Note that legacy software linked to address 0x40000000 may still be used on the GR740 under the right conditions. Either by having 2 GiB of memory installed or by ensuring that the memory will wrap at address 0x40000000.

The memory controllers will wrap at the end of RAM so by installing a smaller amount of memory (less than 2 GiB) it is possible to run applications from 0x40000000 as the memory area will wrap and it will access the same external memory positions as an access to address 0x0. For this workaround to properly work the following conditions must be met:

- The installed memory detected by GRMON3 must be of a size so that the memory will actually wrap at address 0x40000000. This will happen for memories that have bank sizes less than, or equal to, 500 MiB. For example, a 1 GiB single rank memory will not work as the first 1 GiB will be mapped in the range 0x00000000 - 0x3FFFFFFF, the area 0x40000000 - 0x7FFFFFFF will then cause the second chip select to be asserted.
- The stack pointer must be set so that it takes the 0x40000000 offset into account.
- The software may not assume that it can access the range 0x00000000 - 0x3FFFFFFF (this area will contain the PROM and memory mapped I/O areas on legacy LEON systems). Accesses to this range will modify the RAM.
- The software must support plug and play scanning over bridges, or not depend on finding peripherals through plug and play scanning.

3.5. Considerations when enabling the Level-2 cache

The Level-2 cache is disabled after system reset and should be enabled in order to improve system performance. It is important that the Level-2 cache contents is invalidated before the cache is enabled. Otherwise the power-on contents of the cache RAMs may be interpreted as valid data by the cache. Please note that L2 cache is automatically enabled by GRMON2/GRMON3 (see Chapter 4).

4. GRMON3 hardware debugger

4.1. Overview

GRMON3 is a debug monitor used to develop and debug GRLIB/LEON systems. The target system, including the processor and peripherals, is accessed on the AHB bus through a debug-link connected to the host computer. GRMON3 has GDB support which makes C/C++ level debugging possible by connecting GDB to the GRMON3's GDB socket. With GRMON3 one can for example:

- Inspect LEON and peripheral registers
- Upload applications to RAM with the **load** command.
- Program the FLASH with the **flash** command.
- Control execution flow by starting applications (**run**), continue execution (**cont**), single-stepping (**step**), inserting breakpoints/watchpoints (**bp**) etc.
- Inspect the current CPU state listing the back-trace, instruction trace and disassemble machine code.

The first step is to set up a debug link in order to connect to the board. The following section outlines which debug interfaces are available and how to use them on the GR-CPCI-GR740 CompactPCI Development Board. After that, a basic first inspection of the board is exemplified.

Note the Debug Support Unit and the Debug AHB bus must be enabled if GRMON3 is to be used to connect to the board. The DIP switch FP-S3-1 must be set to OPEN.

Note that the GR740 requires that GRMON3 version 2.0.71 is used. Earlier versions will not use the correct JTAG version and will not recognize all the clock-gated cores in the clock-gating unit.

Several of the SoC's peripherals may be clock gated off. GRMON3 will enable all clocks if started with the flag **-cginit**. Within GRMON3, the command **grcg enable all** will have the same effect.

GRMON3 is described on the homepage [https://www.gaisler.com/index.php/products/debug-tools] and in detail in [RD-4].

It is recommended to use version GRMON3 3 or later with GR740.

4.2. Debug-link alternatives

4.2.1. Connecting via the FTDI USB/JTAG interface

Please see Section 2.6.1 to configure FTDI interface.

Please see GRMON User's Manual for how to set up the required FTDI driver software. Then connect the PC and the board using a standard USB cable into the FTDI USB connector and issue the following command:

```
grmon -ftdi
```

For grmon version 3.2.2 and later, please use the following command:

```
grmon -ftdi -sdfreq <mhz>
```

4.2.2. Connecting via the Ethernet debug interfaces

The design has two Ethernet debug communication links (EDCL). These links have default addresses in the range 192.168.0.16 to 192.168.0.47. The GR-CPCI-GR740 should not be connected to an existing network where these addresses may be already occupied. The selection of address can be controlled via bootstrap signals where the first Ethernet debug link can be bootstrapped to an address in the range 192.168.0.16 - 192.168.0.31 and the second link to an address in the range 192.168.0.32 - 192.168.0.47 (see Section 2.6.2). The second link has to be enabled on the pin multiplexing (see Section 2.5.2).

If another address is wanted for the Ethernet debug link then one of the other debug links must be used to connect GRMON3 to the board. The EDCL IP address can then be changed using GRMON3's **edcl** command. This new address will persist until next system reset.

Note that the Ethernet debug link traffic can be routed either to the Master I/O AHB bus or to the Debug AHB bus. In order to control the LEON processors the debug link must be routed to the Debug AHB bus, otherwise GRMON3 will not be able to use the debug link to access the Debug Support Unit. For all uses except testing of

IOMMU functionality it is recommended that DIP switch FP-S2-1 and FP-S2-2 are set to OPEN to route debug Ethernet traffic via the Debug AHB bus.

After reset the first Ethernet debug communication link will attempt to configure the Ethernet PHY. In order for this to succeed, the Ethernet 0 port must be connected to a switch or other networking equipment. Once the PHY for Ethernet 0 has been configured then control over the shared MDIO bus will be given to Ethernet 1. This means that in order to use the Ethernet 1 debug communication link, Ethernet 0 must also be connected to a network.

With the Ethernet Debug Communication Link 0 address set to 192.168.0.16 the GRMON3 command to connect to the board is:

```
grmon -eth 192.168.0.16
```

For grmon version 3.2.2 and later, please use the following command:

```
grmon -eth 192.168.0.16 -sdfreq <mhz>
```

4.2.3. Connecting via SpaceWire RMAP interface

Please see Section 2.6.7 to configure SPW DSU interface.

GRMON3 has support for connecting to boards with SpaceWire interfaces as long as the SpaceWire has RMAP and automatic link start. An Ethernet to SpaceWire bridge (GRESB) is required to tunnel SpaceWire packets from the Ethernet network over to SpaceWire.

Please see the [RD-4] for information about connecting through a GRESB and optional parameters. Connect the GRESB SpW0 connector and the GR-CPCI-GR740's SPW DSU connector, then issue the following command:

```
grmon -gresb
```

4.3. First steps

The previous sections have described which debug-links are available and how to start using them with GRMON3. The subsections below assume that GRMON3, the host computer and the GR-CPCI-GR740 board have been set up so that GRMON3 can connect to the board.

When connecting to the board for the first time it is recommended to get to know the system by inspecting the current configuration and hardware present using GRMON3. With the **info sys** command more details about the system is printed and with **info reg** the register contents of the I/O registers can be inspected. Below is a list of items of particular interest:

- AMBA system frequency is printed out at connect, if the frequency is wrong then it might be due to noise in auto detection (small error). See `-freq` flag in the GRMON User's Manual [RD-4].
- Memory location and size configuration is found from the **info sys** output.
- The GR740 has a clock-gating unit which is able to disable/enable clocking and control reset signals. Clocks must be enabled for all cores that LEON software or GRMON3 will be using. The **grcg** command is described in the GRMON User's Manual [RD-4].
- If the Ethernet debug link is present, one can view and change the EDCL IP using the **edcl** command as described in the GRMON User's Manual [RD-4].

4.4. Connecting to the board

The transcript below shows a example session with GRMON3. GRMON3 is started with the `-u` flag in order to redirect UART output to the GRMON3 terminal.

```
cg@hwlin0:~$ grmon -ftdi -u
  GRMON LEON debug monitor v3.0.12-86-gc455eae 64-bit internal version

  Copyright (C) 2018 Frontgrade Gaisler - All rights reserved.
  For latest updates, go to http://www.gaisler.com/
  Comments or bug-reports to support@gaisler.com

  This internal version will expire on 09/05/2019

Parsing -u
Parsing -ftdi

Commands missing help:

JTAG chain (1): GR740
  Device ID:           0x740
  GRLIB build version: 4153
  Detected system:     GR740 rev1
  Detected frequency:  250 MHz

  Component                            Vendor
  JTAG Debug Link                      Frontgrade Gaisler
  GRSPW2 SpaceWire Serial Link         Frontgrade Gaisler
  EDCL master interface                Frontgrade Gaisler
  EDCL master interface                Frontgrade Gaisler
  LEON4 SPARC V8 Processor             Frontgrade Gaisler
  LEON4 SPARC V8 Processor             Frontgrade Gaisler
  LEON4 SPARC V8 Processor             Frontgrade Gaisler
  LEON4 SPARC V8 Processor             Frontgrade Gaisler
  IO Memory Management Unit            Frontgrade Gaisler
  AHB-to-AHB Bridge                    Frontgrade Gaisler
  L2-Cache Controller                  Frontgrade Gaisler
  AHB Memory Scrubber                  Frontgrade Gaisler
  IOMMU secondary master i/f           Frontgrade Gaisler
  AHB-to-AHB Bridge                    Frontgrade Gaisler
  LEON4 Debug Support Unit             Frontgrade Gaisler
  AHB/APB Bridge                       Frontgrade Gaisler
  AMBA Trace Buffer                    Frontgrade Gaisler
  AHB/APB Bridge                       Frontgrade Gaisler
  AHB/APB Bridge                       Frontgrade Gaisler
  Muxed FT DDR/SDRAM controller        Frontgrade Gaisler
  Memory controller with EDAC          Frontgrade Gaisler
  GRPCI2 PCI/AHB bridge                Frontgrade Gaisler
  GRSPW Router                         Frontgrade Gaisler
  LEON4 Statistics Unit                Frontgrade Gaisler
  GRPCI2 Trace buffer                  Frontgrade Gaisler
  Generic UART                         Frontgrade Gaisler
  Generic UART                         Frontgrade Gaisler
  General Purpose I/O port             Frontgrade Gaisler
  Multi-processor Interrupt Ctrl.      Frontgrade Gaisler
  Modular Timer Unit                   Frontgrade Gaisler
  Modular Timer Unit                   Frontgrade Gaisler
  Modular Timer Unit                   Frontgrade Gaisler
  Modular Timer Unit                   Frontgrade Gaisler
  Modular Timer Unit                   Frontgrade Gaisler
  GRSPW Router DMA interface           Frontgrade Gaisler
  GRSPW Router DMA interface           Frontgrade Gaisler
  GRSPW Router DMA interface           Frontgrade Gaisler
  GRSPW Router DMA interface           Frontgrade Gaisler
  GR Ethernet MAC                      Frontgrade Gaisler
  GR Ethernet MAC                      Frontgrade Gaisler
  CAN Controller with DMA              Frontgrade Gaisler
  CAN Controller with DMA              Frontgrade Gaisler
  SPI Controller                       Frontgrade Gaisler
  Clock gating unit                    Frontgrade Gaisler
  MIL-STD-1553B Interface              Frontgrade Gaisler
  AHB Status Register                  Frontgrade Gaisler
  AHB Status Register                  Frontgrade Gaisler
  General Purpose I/O port             Frontgrade Gaisler
  General Purpose Register             Frontgrade Gaisler
  Temperature sensor                   Frontgrade Gaisler
  General Purpose Register Bank        Frontgrade Gaisler
  CCSDS TDP / SpaceWire I/F            Frontgrade Gaisler
  LEON4 Statistics Unit                Frontgrade Gaisler
  64-bit PC133 SDRAM Controller        Frontgrade Gaisler

  Use command 'info sys' to print a detailed report of attached cores

grmon3> info sys
  ahbjtag0  Frontgrade Gaisler  JTAG Debug Link
            AHB Master 0
  grspw0    Frontgrade Gaisler  GRSPW2 SpaceWire Serial Link
            AHB Master 1
            APB: E4000000 - E4000100
            Number of ports: 1
  adev2     Frontgrade Gaisler  EDCL master interface
            AHB Master 2
  adev3     Frontgrade Gaisler  EDCL master interface
            AHB Master 3
  cpu0      Frontgrade Gaisler  LEON4 SPARC V8 Processor
            AHB Master 0
  cpu1      Frontgrade Gaisler  LEON4 SPARC V8 Processor
            AHB Master 1
  cpu2      Frontgrade Gaisler  LEON4 SPARC V8 Processor
            AHB Master 2
  cpu3      Frontgrade Gaisler  LEON4 SPARC V8 Processor
            AHB Master 3
  iommu0    Frontgrade Gaisler  IO Memory Management Unit
            AHB Master 4
            AHB: FF840000 - FF848000
            IRQ: 31
```

```
          Device index: 0
          Protection modes: APV and IOMMU
          msts: 11, grps: 8, accsz: 128 bits
          APV cache lines: 32, line size: 16 bytes
          cached area: 0x00000000 - 0x80000000
          IOMMU TLB entries: 32, entry size: 16 bytes
          translation mask: 0xff000000
          Core has multi-bus support
          Core has 4 ASMP register blocks
ahb2ahb0  Frontgrade Gaisler  AHB-to-AHB Bridge
          AHB Master 5
          AHB: 00000000 - 80000000
          AHB: 80000000 - C0000000
          AHB: C0000000 - E0000000
          AHB: F0000000 - 00000000
l2cache0  Frontgrade Gaisler  L2-Cache Controller
          AHB Master 0
          AHB: 00000000 - 80000000
          AHB: F0000000 - F0400000
          AHB: FFE00000 - FFF00000
          IRQ: 28
          L2C: 4-ways, cachesize: 2048 kbytes, mtrr: 16, FT, AHB SPLIT support
memscrub0 Frontgrade Gaisler  AHB Memory Scrubber
          AHB Master 1
          AHB: FFE01000 - FFE01100
          IRQ: 28
          burst length: 32 bytes
adev12    Frontgrade Gaisler  IOMMU secondary master i/f
          AHB Master 2
ahb2ahb1  Frontgrade Gaisler  AHB-to-AHB Bridge
          AHB Master 0
          AHB: 80000000 - C0000000
          AHB: C0000000 - D0000000
          AHB: D0000000 - E0000000
          AHB: FF800000 - FF900000
dsu0      Frontgrade Gaisler  LEON4 Debug Support Unit
          AHB: E0000000 - E4000000
          AHB trace: 256 lines, 128-bit bus
          CPU0:  win 8, nwp 4, itrace 512, V8 mul/div, srmmu, lddel 1, GRFPU
                 , FT
                 stack pointer 0x0ffffff0
                 icache 4 * 4 kB, 32 B/line, lru
                 dcache 4 * 4 kB, 32 B/line, lru, snoop tags
          CPU1:  win 8, nwp 4, itrace 512, V8 mul/div, srmmu, lddel 1, GRFPU
                 , FT
                 stack pointer 0x0ffffff0
                 icache 4 * 4 kB, 32 B/line, lru
                 dcache 4 * 4 kB, 32 B/line, lru, snoop tags
          CPU2:  win 8, nwp 4, itrace 512, V8 mul/div, srmmu, lddel 1, GRFPU
                 , FT
                 stack pointer 0x0ffffff0
                 icache 4 * 4 kB, 32 B/line, lru
                 dcache 4 * 4 kB, 32 B/line, lru, snoop tags
          CPU3:  win 8, nwp 4, itrace 512, V8 mul/div, srmmu, lddel 1, GRFPU
                 , FT
                 stack pointer 0x0ffffff0
                 icache 4 * 4 kB, 32 B/line, lru
                 dcache 4 * 4 kB, 32 B/line, lru, snoop tags
apbmst0   Frontgrade Gaisler  AHB/APB Bridge
          AHB: E4000000 - E4100000
ahbtrace0 Frontgrade Gaisler  AMBA Trace Buffer
          AHB: EFF00000 - EFF20000
          Trace buffer size: 128 lines
apbmst1   Frontgrade Gaisler  AHB/APB Bridge
          AHB: FF900000 - FFA00000
apbmst2   Frontgrade Gaisler  AHB/APB Bridge
          AHB: FFA00000 - FFB00000
ddrsdmux0 Frontgrade Gaisler  Muxed FT DDR/SDRAM controller
          AHB: 00000000 - 80000000
          AHB: FFE00000 - FFE00100
          Backend: sdctrl0
mctrl0    Frontgrade Gaisler  Memory controller with EDAC
          AHB: C0000000 - D0000000
          AHB: D0000000 - E0000000
          APB: FF903000 - FF903100
          8-bit prom @ 0xc0000000
pci0      Frontgrade Gaisler  GRPCI2 PCI/AHB bridge
          AHB: 80000000 - C0000000
          AHB: FF800000 - FF840000
          APB: FFA00000 - FFA00100
          IRQ: 11
          Device is disabled
spwrtr0   Frontgrade Gaisler  GRSPW Router
          AHB: FF880000 - FF882000
          IRQ: 31
          Instance id: 19
          SpW ports:  8  AMBA ports:  4  FIFO ports:  0
l4stat0   Frontgrade Gaisler  LEON4 Statistics Unit
          APB: E4000200 - E4000400
          Device is disabled
pcitrace0 Frontgrade Gaisler  GRPCI2 Trace buffer
          APB: E4040000 - E4080000
          Device is disabled
uart0     Frontgrade Gaisler  Generic UART
          APB: FF900000 - FF900100
          IRQ: 29
          Baudrate 38390, FIFO debug mode
uart1     Frontgrade Gaisler  Generic UART
          APB: FF901000 - FF901100
          IRQ: 30
          Baudrate 38390, FIFO debug mode
gpio0     Frontgrade Gaisler  General Purpose I/O port
          APB: FF902000 - FF902100
          IRQ: 16
irqmp0    Frontgrade Gaisler  Multi-processor Interrupt Ctrl.
          APB: FF904000 - FF908000
          EIRQ: 10
gptimer0  Frontgrade Gaisler  Modular Timer Unit
          APB: FF908000 - FF908100
          IRQ: 1
          16-bit scalar, 5 * 32-bit timers, divisor 250
gptimer1  Frontgrade Gaisler  Modular Timer Unit
          APB: FF909000 - FF909100
          IRQ: 6
          16-bit scalar, 4 * 32-bit timers, divisor 250
gptimer2  Frontgrade Gaisler  Modular Timer Unit
          APB: FF90A000 - FF90A100
          IRQ: 7
          16-bit scalar, 4 * 32-bit timers, divisor 250
gptimer3  Frontgrade Gaisler  Modular Timer Unit
          APB: FF90B000 - FF90B100
          IRQ: 8
          16-bit scalar, 4 * 32-bit timers, divisor 250
gptimer4  Frontgrade Gaisler  Modular Timer Unit
          APB: FF90C000 - FF90C100
          IRQ: 9
          16-bit scalar, 4 * 32-bit timers, divisor 250
grspw1    Frontgrade Gaisler  GRSPW Router DMA interface
          APB: FF90D000 - FF90E000
          IRQ: 20
          Number of ports: 1
grspw2    Frontgrade Gaisler  GRSPW Router DMA interface
          APB: FF90E000 - FF90F000
          IRQ: 21
          Number of ports: 1
grspw3    Frontgrade Gaisler  GRSPW Router DMA interface
          APB: FF90F000 - FF910000
          IRQ: 22
          Number of ports: 1
grspw4    Frontgrade Gaisler  GRSPW Router DMA interface
          APB: FF910000 - FF911000
          IRQ: 23
          Number of ports: 1
greth0    Frontgrade Gaisler  GR Ethernet MAC
          APB: FF940000 - FF940100
          IRQ: 24
          1000 Mbit capable
          edcl ip 192.168.0.31, buffer 2 kbyte
greth1    Frontgrade Gaisler  GR Ethernet MAC
          APB: FF980000 - FF980100
          IRQ: 25
          Device is disabled
grcan0    Frontgrade Gaisler  CAN Controller with DMA
          APB: FFA01000 - FFA01400
          IRQ: 16
          Device is disabled
grcan1    Frontgrade Gaisler  CAN Controller with DMA
          APB: FFA02000 - FFA02400
          IRQ: 16
          Device is disabled
spi0      Frontgrade Gaisler  SPI Controller
          APB: FFA03000 - FFA03100
          IRQ: 19
          Device is disabled
grcg0     Frontgrade Gaisler  Clock gating unit
          APB: FFA04000 - FFA04100
          GRMON did NOT enable clocks during initialization
gr1553b0  Frontgrade Gaisler  MIL-STD-1553B Interface
          APB: FFA05000 - FFA05100
          IRQ: 26
```

```
            Device is disabled
  ahbstat0  Frontgrade Gaisler  AHB Status Register
            APB: FFA06000 - FFA06100
            IRQ: 27
  ahbstat1  Frontgrade Gaisler  AHB Status Register
            APB: FFA07000 - FFA07100
            IRQ: 27
  gpio1     Frontgrade Gaisler  General Purpose I/O port
            APB: FFA08000 - FFA08100
            IRQ: 16
  gpreg0    Frontgrade Gaisler  General Purpose Register
            APB: FFA09000 - FFA09100
  adev49    Frontgrade Gaisler  Temperature sensor
            APB: FFA0A000 - FFA0A100
  gpreg1    Frontgrade Gaisler  General Purpose Register Bank
            APB: FFA0B000 - FFA0B100
  spwtdp0   Frontgrade Gaisler  CCSDS TDP / SpaceWire I/F
            APB: FFA0C000 - FFA0C200
            IRQ: 31
  l4stat1   Frontgrade Gaisler  LEON4 Statistics Unit
            APB: FFA0D000 - FFA0D200
            cpus: 4, counters: 16, i/f index: 1
  sdctrl0   Frontgrade Gaisler  64-bit PC133 SDRAM Controller
            AHB: 00000000 - 80000000
            AHB: FFE00000 - FFE00100
            64-bit sdram: 2 * 128 Mbyte @ 0x00000000,
            col 9, cas 3, ref 7.8 us

grmon3> info sys dsu0
  dsu0      Frontgrade Gaisler  LEON4 Debug Support Unit
            AHB: E0000000 - E4000000
            AHB trace: 256 lines, 128-bit bus
            CPU0:  win 8, nwp 4, itrace 512, V8 mul/div, srmmu, lddel 1, GRFPU
                   , FT
                   stack pointer 0x0ffffff0
                   icache 4 * 4 kB, 32 B/line, lru
                   dcache 4 * 4 kB, 32 B/line, lru, snoop tags
            CPU1:  win 8, nwp 4, itrace 512, V8 mul/div, srmmu, lddel 1, GRFPU
                   , FT
                   stack pointer 0x0ffffff0
                   icache 4 * 4 kB, 32 B/line, lru
                   dcache 4 * 4 kB, 32 B/line, lru, snoop tags
            CPU2:  win 8, nwp 4, itrace 512, V8 mul/div, srmmu, lddel 1, GRFPU
                   , FT
                   stack pointer 0x0ffffff0
                   icache 4 * 4 kB, 32 B/line, lru
                   dcache 4 * 4 kB, 32 B/line, lru, snoop tags
            CPU3:  win 8, nwp 4, itrace 512, V8 mul/div, srmmu, lddel 1, GRFPU
                   , FT
                   stack pointer 0x0ffffff0
                   icache 4 * 4 kB, 32 B/line, lru
                   dcache 4 * 4 kB, 32 B/line, lru, snoop tags

grmon3> l2cache invalidate
  invalidate all cache lines

grmon3>  l2cache enable

grmon3> load hello
  00000000 .text                     24.0kB /  24.0kB   [===============>] 100%
  00006000 .data                      2.8kB /   2.8kB   [===============>] 100%
  Total size: 26.84kB (788.19kbit/s)
  Entry point 0x00000000
  Image /home/cg/hello loaded

grmon3> run
Hello world!

  CPU 0:  Program exited normally.
  CPU 1:  Power down mode
  CPU 2:  Power down mode
  CPU 3:  Power down mode

grmon3> hist
      TIME     ADDRESS   INSTRUCTIONS/AHB SIGNALS      RESULT/DATA
     10548312  00001940  restore                       [00000000]
     10548313  00001944  retl                          [00001944]
     10548314  00001948  nop                           [00000000]
     10548317  0000106C  call  0x00004C70              [0000106C]
     10548318  00001070  nop                           [00000000]
     10548319  00004C70  mov  1, %g1                   [00000001]
     10548320  00004C74  ta  0x0                       [  TRAP  ]
     10548331  00000800  AHB read   mst=0  size=4      [91D02000 01000000 01000000 01000000]
     10548332  00000810  AHB read   mst=0  size=4      [91D02000 01000000 01000000 01000000]
     10548337  00000800  ta  0x0                       [  TRAP  ]
```

```
grmon3> inst
      TIME     ADDRESS   INSTRUCTION                   RESULT      SYMBOL
     10548308  0000453C  jmp  %l1                      [0000453C]  -
     10548309  00004540  rett  %l2                     [00001944]  -
     10548312  00001940  restore                       [00000000]  main+0x1c
     10548313  00001944  retl                          [00001944]  main+0x20
     10548314  00001948  nop                           [00000000]  main+0x24
     10548317  0000106C  call  0x00004C70              [0000106C]  -
     10548318  00001070  nop                           [00000000]  -
     10548319  00004C70  mov  1, %g1                   [00000001]  _exit+0x0
     10548320  00004C74  ta  0x0                       [  TRAP  ]  _exit+0x4
     10548337  00000800  ta  0x0                       [  TRAP  ]  -

grmon3>  grcg clkinfo
  GRCLKGATE GR740 info:
  Unlock register:       0x00000000
  Clock enable register: 0x00000585
  Reset register:        0x0000027a

  GR740 decode of values:
  +------+----------+---------------------------+----------+---------+-------+
  | Gate | Core(s)  | Description               | Unlocked | Enabled | Reset |
  +------+----------+---------------------------+----------+---------+-------+
  |   0  | GRETH    | Ethernet MAC 0            |    0     |    1    |   0   |
  |   1  | GRETH    | Ethernet MAC 1            |    0     |    0    |   1   |
  |   2  | SPWRTR   | SpaceWire router          |    0     |    1    |   0   |
  |   3  | PCI      | PCI (GRPCI, PCIDMA)       |    0     |    0    |   1   |
  |   4  | GR1553B  | MIL-STD-1553B             |    0     |    0    |   1   |
  |   5  | GRCAN    | CAN core 0 & 1            |    0     |    0    |   1   |
  |   6  | L4STAT   | LEON4 Statistics          |    0     |    0    |   1   |
  |   7  | APBUART  | UART 0                    |    0     |    1    |   0   |
  |   8  | APBUART  | UART 1                    |    0     |    1    |   0   |
  |   9  | SPICTRL  | SPI Controller            |    0     |    0    |   1   |
  |  10  | MCTRL    | PROM/IO                   |    0     |    1    |   0   |
  +------+----------+---------------------------+----------+---------+-------+

grmon3> grcg enable 4

grmon3>  grcg clkinfo
  GRCLKGATE GR740 info:
  Unlock register:       0x00000000
  Clock enable register: 0x00000595
  Reset register:        0x0000026a

  GR740 decode of values:
  +------+----------+---------------------------+----------+---------+-------+
  | Gate | Core(s)  | Description               | Unlocked | Enabled | Reset |
  +------+----------+---------------------------+----------+---------+-------+
  |   0  | GRETH    | Ethernet MAC 0            |    0     |    1    |   0   |
  |   1  | GRETH    | Ethernet MAC 1            |    0     |    0    |   1   |
  |   2  | SPWRTR   | SpaceWire router          |    0     |    1    |   0   |
  |   3  | PCI      | PCI (GRPCI, PCIDMA)       |    0     |    0    |   1   |
  |   4  | GR1553B  | MIL-STD-1553B             |    0     |    1    |   0   |
  |   5  | GRCAN    | CAN core 0 & 1            |    0     |    0    |   1   |
  |   6  | L4STAT   | LEON4 Statistics          |    0     |    0    |   1   |
  |   7  | APBUART  | UART 0                    |    0     |    1    |   0   |
  |   8  | APBUART  | UART 1                    |    0     |    1    |   0   |
  |   9  | SPICTRL  | SPI Controller            |    0     |    0    |   1   |
  |  10  | MCTRL    | PROM/IO                   |    0     |    1    |   0   |
  +------+----------+---------------------------+----------+---------+-------+

grmon3>
```

5. Board Package

5.1. Overview

The board package distributed together with this document contains GRMON3 scripts and MKPROM initialization functions that are specific to the GR740 or GR-CPCI-GR740. The package is named gr-cpci-gr740-<*version*>.

5.2. MKPROM2 bdinit functions

The subdirectory `GRMON3` contains initialization code to be used with the MKPROM2 boot-PROM builder. To creare boot-PROMs for GR740 MKPROM2 version 2.0.60 or later must be used.

The board package's MKPROM2 directory contains the following files:

- *bdinit.c* - bdinit functions that will be called by MKPROM2.
- *bdinit_gr740_sdctrl0.ci* - File included by *bdinit.c*. Contains initialization code for the GR740 SDRAM memory controller.
- *bdinit_gr740_mctrl0.ci* - File included by *bdinit.c*. Contains initialization code for the GR740 PROM memory controller.
- *bdinit_gr740_l2cache.ci* - File included by *bdinit.c*. Contains initialization code for Level-2 cache.
- *bdinit_gr740_uart.ci* - File included by *bdinit.c*. Contains initialization code for FTMEM and ALTEN register that set up the pin multiplexing so that uart0 and uart1 can be used along with the PROM.
- *bdinit_gr740_grcg.ci* - File included by *bdinit.c*. Contains clock-gating unit enabling and disabling routines.

The bdinit sequence (defined in bdinit.c file) contains the following steps:

- *bdinit0*: Called before the LEON registers have been initialized but before the memory has been cleared.

  1. Setup memory controller (SDCTRL). Please note that the board package assumes the default board configuration and provided memory modules with the board. If you change the memory modules, you must check if this setup still applies.
  2. Enable 2T signaling on memory controller (SDCTRL). This improves the SDRAM signal timing, required for some memory modules.
  3. Initialize memory controller (SDCTRL), which is required after enabling 2T signaling.
  4. Setup PROM write lead out cycles (MCTRL), which is required to be able to write to the PROM.
- *bdinit1*: Called after the LEON registers have been initialized but before the memory has been cleared.

  1. Invalidate L2 cache contents.
  2. Enable L2 cache after invalidate has finished.
- *bdinit2*: Called after the memory has been initialized but before the application is loaded.

  1. Setup pin multiplexing to enable UART0 and UART1. This assumes that the board is configured in the default board configuration.
  2. (Optional) Enable/disable cores on clock-gating unit. Please note that UART0 and MCTRL cores are enabled by default and should not be enabled/disabled here, since that will cause a reset on the cores that are being used by the boot code.

The first step in creating a boot-PROM image for GR-CPCI-GR740 is to compile the bdinit.c file. This is done with the command **sparc-elf-gcc -O2 -c -o bdinit.o gr-cpci-gr740-bp/MKPROM2/bdinit.c**. Note that this requires the Bare-C Compiler (BCC) available from https://www.gaisler.com. Also note that the **-O2** is important, since MKPROM requires bdinit0 and bdinit1 functions to be leaf and not allocate any stack space, i.e. no local variables, see [RD-13] for more information.

The next step is to run **mkprom2** specifying flags that are specific for the GR-CPCI-GR740. The full MKPROM2 command for creating an image is:

```
/opt/mkprom2/mkprom2 -v  \
     -stack <stack pointer> -ramsize <size of RAM in KiB> -sparcleon0 \
     -memcfg1 0x0803c0ff -memcfg3 0x08000000   \
     -dump -v -rstaddr 0xc0000000 -uart 0xFF900000 -freq <frequency> -baud 38400 \
     -bdinit <image> -o <output name>
```

Making a boot-PROM image for a hello world application, named *hello* for a 250 MHz system frequency with 256 MiB of RAM gives requires the MKPROM2 command line below. Note that the application should be linked for RAM starting at address 0. For BCC, this is accomplished with the `-Wl,-msparcleon0` flag.

```
/opt/mkprom2/mkprom2 -v  \
     -stack 0x0fffff00 -ramsize 262144 -sparcleon0 \
     -memcfg1 0x0803c0ff -memcfg3 0x08000000   \
     -dump -v -rstaddr 0xc0000000 -uart 0xFF900000 -freq 250 -baud 38400 \
     -bdinit hello -o hello.prom
```

The output of calling mkprom2 2.0.60 with the options above is:

```
bash-3.2$ /opt/mkprom2/mkprom2 -v  \
>      -stack 0x0fffff00 -ramsize 262144 -sparcleon0 \
>      -memcfg1 0x0803c0ff -memcfg3 0x08000000   \
>      -dump -v -rstaddr 0xc0000000 -uart 0xFF900000 -freq 250 -baud 38400 \
>      -bdinit hello -o hello.prom

LEON2/3/ERC32 MKPROM prom builder for BCC, ECOS, RTEMS and ThreadX v2.0.60
Copyright Gaisler Research 2004-2007, all rights reserved.

phead0: type: 1, off: 65536, vaddr: 0, paddr: 0, fsize: 27472, msize: 29528
phead1: type: 1, off: 95064, vaddr: 7358, paddr: 7358, fsize: 0, msize: 4
section: .text at 0x0, size 24560 bytes
Uncoded stream length: 24560 bytes
Coded stream length: 13454 bytes
Compression Ratio: 1.825
section: .data at 0x5ff0, size 2912 bytes
Uncoded stream length: 2912 bytes
Coded stream length: 827 bytes
Compression Ratio: 3.521

creating LEON3 boot prom: ./hello.prom
Searching for compiler to use (sparc-elf, sparc-rtems or sparc-linux):
sparc-elf-gcc (BCC 4.4.2 release 1.0.45) 4.4.2
Copyright (C) 2009 Free Software Foundation, Inc.
This is free software; see the source for copying conditions.  There is NO
warranty; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.


sparc-elf-gcc -O2 -g -N -T/opt/mkprom2/linkprom -Ttext=0xc0000000
/opt/mkprom2/promcore.o /opt/mkprom2/prominit.o /opt/mkprom2/prominit_leon3.o
/opt/mkprom2/promcrt0.o /opt/mkprom2/promload.o /opt/mkprom2/promdecomp.o
-nostdlib /opt/mkprom2/prombdinit.o dump.s bdinit.o -o hello.prom
multidir:
```

The *hello.prom* image can now be programmed to the board's Flash. A transcript of this operation using GRMON3 is included below

```
grmon3> flash erase all
  Erase in progress
  Block @ 0xc07e0000 : code = 0x80  OK
  Erase complete

grmon3> flash load hello.prom
  C0000000 .text                     31.1kB /  31.1kB   [===============>] 100%
  Total size: 31.12kB (404.72kbit/s)
  Entry point 0xc0000000
  Image /home/ag/hello.prom loaded

grmon3> verify hello.prom
  C0000000 .text                     31.1kB /  31.1kB   [===============>] 100%
  Total size: 31.12kB (6.07Mbit/s)
  Entry point 0xc0000000
  Image of /home/ag/hello.prom verified without errors

grmon3> flash

  Intel-style 8-bit flash on D[31:24]

  Manuf.        : Intel
  Device        : MT28F640J3
  Device ID     : 96ff440a0000403f
  User ID       : ffffffffffffffff

  1 x 8 Mbytes = 8 Mbytes total @ 0xc0000000

  CFI information
  Flash family  : 1
  Flash size    : 64 Mbit
  Erase regions : 1
  Erase blocks  : 64
  Write buffer  : 32 bytes
  Lock-down     : Not supported
  Region  0     : 64 blocks of 128 kbytes

grmon3>
```

When the board is power-cycled, the following will appear at UART 0 (38400, 8N1):

```
  MkProm2 boot loader v2.0
  Copyright Gaisler Research - all rights reserved

  system clock   : 250.0 MHz
  baud rate      : 38422 baud
  prom           : 512 K, (2/2) ws (r/w)
  sram           : 0 K, 1 bank(s), 0/0 ws (r/w)

  decompressing .text to 0x00000000
  decompressing .data to 0x0000a490

  starting hello

Hello world!
```

Connecting to the GR740 after the boot will show the Level-2 cache as enabled, since it is enabled by the bdinit functions used for the PROM image.

Additional notes on creating boot-PROMs:

- The value written on bdinit_gr740_sdctrl0.ci are the parameters written into the SDRAM controller registers. These parameters depend on the type of SDRAM SODIMM used. If the SDRAM SODIMM is replaced then the simplest way to obtain new parameters is to connect to the design with GRMON3 and issue **info reg** and copy the values that GRMON3 has initialized the SDRAM memory controller with.
- The *-stack* and *-ramsize* parameters should be set according to the amount of available memory. In the example above the stack can be set at top of the 256 MiB area and still work when switching memory interface to the 128 MiB SDRAM as the memory area will wrap.
- The *-memcfg* parameters specify values written to the PROM/IO memory controller configuration registers. In the example above, the PROM width used was 8 bits. To use a 16-bit wide PROM interface, the field at bits 9:8 of MEMCFG1 should be changed from 0 to 1 (*-memcfg1 0x0803c1ff*). The PROM width setting must match with the PROM width selection made via the bootstrap signal GPIO[10] and the setting of jumper JP6f on the board. See section 4.5.4 of the GR-CPCI-GR740 Development Board User's Manual.
- In case the PROM has been programmed with a destructive application it is possible to prevent the processor's from starting up by holding the RESET and BREAK buttons on the front-panel, and then releasing RESET will still pressing BREAK.

6. Frequently Asked Questions / Common Mistakes / Know Issues

6.1. Clock gating

Several of the design's peripherals may be clock gated off. GRMON3 will enable all clocks if started with the flag `-cginit`. Within GRMON3, the command **grcg enable all** will have the same effect.

6.2. GRMON3 issues

When connected to the board, the message "stack pointer not set" will be shown by the command **info sys** in case GRMON3 doesn't find any memory.

When connecting to the board with FTDI in a Linux based system, the message "unable to claim usb device. Make sure the default FTDI driver is not in use" is printed. Make sure that you have set the FTDI udev rules. Please refer to [RD-4]. If the problem persist, try stopping the kernel driver *ftdi_sio*.

6.3. Level-2 cache initialization

The Level-2 cache (L2C) should always be enabled in order to obtain adequate performance. When connection to the board is established using GRMON3/GRMON2, l2cache is automatically enabled by GRMON debugger. Otherwise, the L2C has to be enabled by software, as shown in Chapter 5. When enabling L2C it is important that all entries in the cache are invalidated. Otherwise power-on values in the cache's internal memories may be interpreted as valid cache data.

The Level-2 cache contents can be invalidated, and the cache then enabled with the following GRMON3 sequence:

```
grmon3> l2cache invalidate
  invalidate all cache lines

grmon3> l2cache enable
```

6.4. Main memory problems

If the memory is behaving erraticaly, please make sure that the configuration plug is properly adjusted on the connector J23.

6.5. Main memory interface EDAC

EDAC on the main memory interface can be enabled via GRMON3. First all memory that will be used needs to be initialized. This can be done with help of the hardware memory scrubber. Next, the EDAC is enabled by EDAC enable bit must be set in the memory controller's FT Configuration Register. The full initialization sequence, with also Level-2 cache enabled becomes:

```
grmon3> scrub clear 0 0x0fffffff
  0x00000000                        256.0MB / 256.0MB   [===============>] 100%

grmon3> wmem 0xffe00020 0x1

grmon3> l2cache invalidate
  invalidate all cache lines

grmon3> l2cache enable
```

Note that the **scrub** above initialises 256 MiB of memory. If the system has 128 MiB of memory then the command is **scrub clear 0 0x07ffffff**.

The scrubber monitors the Memory AHB bus for errors reported by the memory controller. The command **scrub** shows the status:

```
grmon3> scrub
  AHB status register: Not triggered
  Scrubber status:     Done init 00000000-0fffffff

  Error count/limits:  UE:0/0, CE:0/0, SEC:0/off, SBC:0/off
  Scrubber config:     Loop:0, Extstart:0, Extclear:0, Delay: 0
```

In case errors have been detected, the **scrub** will show the affected address:

```
grmon3> scrub
  AHB status register: ERROR at addr 0f4027c0 (mst 0 hsize 4 hwrite 0)
  Scrubber status:     Idle

  Error count/limits:  UE:1/0, CE:0/0, SEC:0/off, SBC:0/off
  Scrubber config:     Loop:0, Extstart:0, Extclear:0, Delay: 0
```

6.6. Flash programming

If Flash programming is too slow, use the EDCL debug link to program the flash, since it can offer up to 10x speed-ups.

6.7. PROM EDAC

To boot with PROM EDAC enabled you need add the "-bch8 -romsize 8192" to your MKPROM2 flags. MKPROM2 will then create a PROM binary with the EDAC checkbits (-bch8) located at the end of the PROM memory area, given by the board's default configuration PROM size of 8 MiB (-romsize 8192, in KiB). The outfile file with .bch8 extension can be loaded to the PROM as usual. To enable PROM EDAC, set GPIO[14] to pull-up (PU).

When having custom pin multiplexing solutions, you need to make sure that the PROM address lines pin multiplexing is correct and matches the PROM size given to MKPROM2. For instance, to have a PROM size of 8 MiB flash, which is the default configuration, you need the PROM_ADDR[20:22] pins to access the EDAC checkbits. In the board pin multiplexing, these pins can be assigned to the PROM memory or to the 1553TXNB, 1553TXNB and 1553RXENB signals. If for instance, we need to use the 1553 interface signals, we cannot have a 8 MiB flash and therefore we need to adjust the MKPROM2 flags for a 1 MiB flash to be able to use the PROM_ADDR[20:22] pins for the 1553 interface. For that purpose, change the MKPROM flags from "-romsize 8192" to "-romsize 1024". Please note that this custom configuration leaves PROMIO_ADDR 21 and 20 (JP11 7 and 8) floating, which are connected to the flash memory (this is a board limitation).

6.8. Ethernet

If ethernet port 1 is not working properly, make sure that a) the pin multiplexing is correctly configured and b) the configuration plug is properly connected. See Section 2.5.2.

If EDCL is not working, please make sure that a) GPIO[8] and GPIO[9] are set to route traffic on the Debug AHB bus; b) a proper IP address has been selected either through GPIO[0-5] or via GRMON3; and c) FP-S3-7 selects the correct network speed. See Section 2.6.2.

6.9. UART

If the UART is either displaying rubbish characters or not displaying anything, make sure that a) the pin multiplexing is properly configured and b) the FTMEN and ALTEN registers allow UART operation. See Section 2.5.1.

6.10. Pin multiplexing (PROMIO/Peripherals)

The default full 8MiB PROM in 8-bit mode and the UART can be used at the same time. The 16-bit PROM mode cannot be used with UARTs at the same time. See Section 2.5.1.

In case the full PROM and all the interfaces are desired, what can be done is to connect the IOs from the GR740 to both the PROM and the interface transceivers, boot up in the full-PROM mode by pulling down gpio(15) during reset, and after completing the decompression of your PROM to RAM, switch over the IOs to their interface functions. This will have the side effect of toggling the interface IO lines according to the address driven to the PROM during the decompression phase which might cause unwanted interface communication. For 1553 this has been solved by making the TX-inhibit lines non-multiplexed which will prevent any 1553 data from going out. For the other IOs you will have to make a system-level decision as to whether you can tolerate some IO interfaces toggling during boot-up, and if not, you could use one of the GPIOs to drive an enable signal to those IO interfaces transceivers to disable them during boot. Please note that this is not supported currently on the GR-CPCI-GR740 development board.

6.11. Can't boot

First, check that the BREAK button is not asserted (should be in the rightmost position). Second, check that your boot image is properly loaded into the flash memory (starting address 0xc0000000). You can use the **verify** command of GRMON3 that will do it for you. Third, make sure that both memory controllers (mctrl0 and sdctrl0) are properly initialized. To check if that is happening, you can connect with GRMON3 with no initialization flag (-ni) once your system has been powered up. For instance, using JTAG/FTDI debug link, the command is **grmon -ftdi -ni**. Please note, that if you use GRMON3 without the **-ni** option, GRMON3 initializes both memory controllers and thus, the state left by your boot code cannot be analysed. Once in grmon, check the value of the memory controller configuration registers, as shown below:

```
grmon3> info reg -v mctrl0
  Memory controller with EDAC
      0xff903000  Memory config register 1                0x0803c0ff
      30     pbrdy             0x0         PROM area bus ready enable
      29     abrdy             0x0         Asynchronous bus ready enable
      28:27  iobusw            0x1         I/O bus width
      26     ibrdy             0x0         I/O bus ready enable
      25     bexcn             0x0         Bus error enable
      23:20  iows              0x0         I/O wait states
      19     ioen              0x0         I/O enable
      17:14  prombanksz        0xf         PROM bank size
      11     pwen              0x0         PROM write enable
       9:8   promwidth         0x0         PROM width
       7:4   promwws           0xf         PROM write wait states
       3:0   promrws           0xf         PROM read wait states

      0xff903004  Memory config register 2                0x00000020
      31     sdramrf           0x0         SDRAM refresh enable
      30     sdramtrp          0x0         SDRAM TRP parameter
      29:27  sdramtrfc         0x0         SDRAM TRFC parameter
      26     sdramtcas         0x0         SDRAM TCAS parameter
      25:23  sdrambanksz       0x0         SDRAM bank size
      22:21  sdramcolsz        0x0         SDRAM column size
      20:19  sdramcmd          0x0         SDRAM command
      18     d64               0x0         SDRAM 64-bit data bus
      17     sdpb              0x0         SDRAM page burst
      14     se                0x0         SDRAM enable
      13     si                0x0         SRAM disable
      12:9   rambanksz         0x0         RAM bank size
       7     rbrdy             0x0         RAM bus read enable
       6     rmw               0x0         Read-modify-write enable
       5:4   ramwidth          0x2         RAM width
       3:2   ramwws            0x0         RAM write wait states
       1:0   ramrws            0x0         RAM read wait states

      0xff903008  Memory config register 3                0x08000000
      28     rse               0x0         Reed-Solomon EDAC enable
      27     me                0x1         Memory EDAC available
      26:12  sdramreload       0x0         SDRAM refresh counter reload value
      11     wb                0x0         EDAC diagnostic write bypass enable
      10     rb                0x0         EDAC diagnostic read bypass enable
       9     re                0x0         RAM EDAC enable
       8     pe                0x0         PROM EDAC enable
       7:0   tcb               0x0         Test checkbits

      0xff903010  Memory config register 5                0x00000000
      29:23  iohws             0x0         IO lead out
      13:7   romhws            0x0         ROM lead out

      0xff903014  Memory config register 6                0x00000000
      13:7   ramhws            0x0         RAM lead out

      0xff903018  Memory config register 7                0x00000000
      31:16  brdyncnt          0x0         Bus ready count
      15:0   brdynrld          0x0         Bus ready reload value

grmon3> info reg -v sdctrl0
  PC133 SDRAM Controller
      0xffe00000  SDRAM config register                   0xfea087a4
      31     refresh           0x1         SDRAM refresh enable
      30     trp               0x1         SDRAM TRP parameter
      29:27  trfc              0x7         SDRAM TRFC parameter
      26     tcas              0x1         SDRAM TCAS parameter
      25:23  banksz            0x5         SDRAM bank size
      22:21  colsz             0x1         SDRAM column size
```

```
20:18   cmd                 0x0         SDRAM command
17      pb                  0x0         Pageburst
16      ms                  0x0         Mobile SDR support
15      d64                 0x1         SDRAM 64-bit data bus
14:0    reload              0x7a4       Refresh reload value
```

These registers have to be set up by the boot code in order to get a working memory.

6.12. FTDI/JTAG

If the JTAG connection is not working. Check the following things:

- Make sure that the DSU is enabled (FP-S3-1) in open position (see Figure 2.2).
- Check that all JP4 jumpers are connected (see Figure 2.6).
- Check that the GPIO ribbon cable that goes from the front panel to the PCB J4 connector is properly connected (see Figure 6.1). Sometimes this connector moves a little bit out due to movement of the board and creates an unstable JTAG connection.
- Some USB cables do not fit properly on the board connector due to the plastic protector of the cable that prevents the connector to go deeper into the board connector. Try removing a little bit of the plastic protection and see if that works.

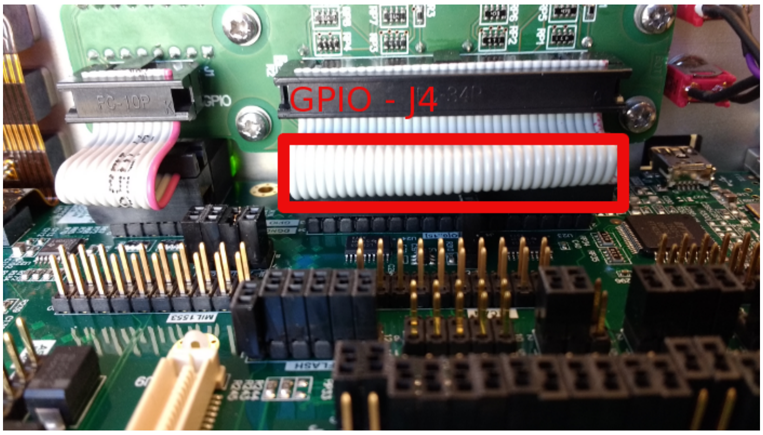

Figure 6.1. GR-CPCI-GR740 GPIO J4 connection.

6.13. SDRAM not working

If the SDRAM is not working. Check the following things:

- Make sure that the configuration plug is properly installed on the desired configuration (see Figure 2.5). This connector has created problems after significant physical manipulation and movement of the board, creating an unstable SDRAM connection. Try re-installing it.

7. Support

For support contact the support team at support@gaisler.com.

When contacting support, please identify yourself in full, including company affiliation and site name and address. Please identify exactly what product that is used, specifying if it is an IP core (with full name of the library distribution archive file), component, software version, compiler version, operating system version, debug tool version, simulator tool version, board version, etc.

The support service is only for paying customers with a support contract.

Appendix A. Default configuration

This appendix describes the jumper and switch positions for the default configuration of the board. See section *Setting Up and Using the Board* on the GR-CPCI-GR740 Development Board User's Manual.

Table A.1. Jumper configuration

Jumper	Default configuration
JP1	Connected.
JP2	All disconnected.
JP3	All disconnected.
JP4	All connected.
JP5	1-2, 3-4 connected. 5-6 disconnected
JP6	All connected except 1-2.
JP7	All disconnected.
JP8	Connected.
JP9	Connected.
JP10	Disconnected.
JP11	1, 2, 13 and 14 in position B-C. The rest in position A-B.
JP12	Disconnnected.
JP13	Disconnected.
JP14	Connected.
JP15	Connected.
JP16	Connected.
JP17	Disconnected.
JP18	Disconnected.
JP19	Disconnected.
JP20	Disconnected.
JP21	Disconnected.
JP22	Connected.
JP23	Connected.
JP24	All disconnected.

Table A.2. Switch configuration

Switch	Default configuration
FP-S1-1	Pull-up.
FP-S1-2	Pull-up.
FP-S1-3	Pull-up.
FP-S1-4	Pull-up.
FP-S1-5	Pull-up.
FP-S1-6	Pull-up.
FP-S1-7	Pull-up.
FP-S1-8	Pull-up.
FP-S2-1	Pull-up.
FP-S2-2	Pull-up.

Switch	Default configuration
FP-S2-3	Pull-down.
FP-S2-4	Pull-up.
FP-S2-5	Pull-up.
FP-S2-6	Pull-up.
FP-S2-7	Pull-down.
FP-S2-8	Pull-down.
FP-S3-1	Open.
FP-S3-2	Open.
FP-S3-3	Closed.
FP-S3-4	Closed.
FP-S3-5	Closed.
FP-S3-6	Open.
FP-S3-7	Closed.
FP-S3-8	Open.
S1	Right position.

Frontgrade Gaisler AB
Kungsgatan 12
411 19 Göteborg
Sweden
frontgrade.com/gaisler
sales@gaisler.com
T: +46 31 7758650
F: +46 31 421407

Frontgrade Gaisler AB, reserves the right to make changes to any products and services described herein at any time without notice. Consult the company or an authorized sales representative to verify that the information in this document is current before using this product. The company does not assume any responsibility or liability arising out of the application or use of any product or service described herein, except as expressly agreed to in writing by the company; nor does the purchase, lease, or use of a product or service from the company convey a license under any patent rights, copyrights, trademark rights, or any other of the intellectual rights of the company or of third parties. All information is provided as is. There is no warranty that it is correct or suitable for any purpose, neither implicit nor explicit.