MCF5307 ColdFire
Janus Development Board
Manual

CJ Design Services
RR#1 Sebringville ON
N0K 1X0
cjd@emorrison.ca
Phone: 519.393.6039
Fax: 519.393.6039

Limited Warranty

The Janus Development Board is warranted against defects in materials and workmanship for a period of ninety days from the date of purchase from CJ Design Services.
This warranty does not apply if, in the opinion of CJ Design Services, the product has been damaged by accident, misuse, neglect, or as a result of service or modification.
In no event will CJ Design Services be liable for any incidental or consequential damages.

Disclaimer

CJ Design Services reserves the right to change this product without prior notice. Information furnished by CJ Design Services is believed to be accurate and reliable. However, no responsibility is assumed by CJ Design Services for its use; nor for any infringement of patents or other rights of third parties, which may result from its use.

Trademark Notice

Motorola is a registered trademark of Motorola Inc.
Microsoft and Windows are registered trademarks of Microsoft Corporation.
Adobe and Acrobat are trademarks of Adobe Systems Incorporated.
All other brand and product names are trademarks or service marks of their respective owners.

Table of Contents

Introduction

Thank you for purchasing the Janus Development Board. This board is designed to quickly develop and prototype ideas into solutions based on the ColdFire processor.

Access to the development board is achieved through the serial terminal port. New programs and data can be downloaded either through the serial aux port or more efficiently through the Ethernet port.

The board's features include:

- Motorola MCF5307 CPU with 90MHz instruction rate
- 10Base-T Ethernet connection
- PC100 SDRAM socket for external memory
- Upgradeable monitor program in non-volatile flash memory

Getting Started

Connections to power and a terminal are required for board operation. In order to run the monitor program SDRAM must be added to the board also.

Adding Memory

In order to run the monitor program or any substantial user code, SDRAM must be added to the board. All single-sided PC100 SDRAM modules are accepted by the board. Simply insert the SDRAM module into the SDRAM socket (J3) and firmly press in until the end levers snap into place. The board will automatically detect the type of memory and configure itself accordingly.

A quick memory test is performed on power-up. If the performance of the memory is suspect, there is a full memory test available in the diagnostics menu.

Connecting Power

Power is supplied to the board through the 2.1mm jack (J10). It is designed for 5-9VDC with the center pin being negative. The power supply must be capable of supplying 600mA.

If circuitry is added via the processor expansion port extra current may be required.

Connecting the Terminal

Use a straight-through serial cable to connect the Terminal Port of the Janus Development Board to a terminal (or PC acting as a terminal). A 9-pin-to-25-pin adapter may be needed to connect to some terminals or PC's.

The terminal must use the following settings for power-up:

Baud:	19200
Data Bits:	8
Parity:	none
Stop Bits:	1
Handshaking:	hardware

These settings are required on power-up. Once powered, the serial port speed can be adjusted through the configuration menus or monitor program.

Using the Embedded Software

Board Power-Up Sequence

When the board powers up the following actions are taken:

- The CPU reset vector fetches the values at address 0.
- These values cause the CPU to start executing code for the bootrom.
- In the bootrom, the following parts of the board are setup in the following order:
    a. MCF5307 SIM
    b. MCF5307 Parallel Port
    c. MCF5307 Serial Ports
    d. MCF5307 Chip Selects
    e. MCF5307 DRAM Controller
- The 128K flash bank containing janusROM is copied to DRAM and control is transferred to the start of DRAM.
- JanusROM then sets up it's environment – clearing memory, configuring any remaining devices, and finally going into interactive mode waiting for the user.

Onboard monitor – janusROM

The onboard monitor, janusROM, provides facilities for startup parameters, downloading software into DRAM (and flash), inspecting memory and registers, and performing simple debugging (tracing and break-points). By entering the command *help* or *?* from the *janusROM>* prompt, a list of all possible commands with basic summary information will be listed. Each command can be entered with a *–h* to get command specific details.

Parameters

The flash device used on this board is partitioned into several different spaces – each of which can be erased with no effect on the rest of the chip. One of these banks is used by janusROM to store parameter values. These values allow customization of many aspects of the board. To list all the parameters used by the system, simply enter the *params* command with the *–l* option. This is the same command that is used to update or erase parameters. The following table lists all the default parameters and what they mean to a running system.

PARAMETER	MEANING
terminal-speed	Baud rate of the terminal serial port.
aux-speed	Baud rate of the aux serial port.
monitor	Which port (terminal or aux) janusROM will run on.
download	Which port (terminal or aux) will be the default for downloads.
download-format	Format (binary or srec) to use for downloads.
download-address	Location to download binary images.
macaddr	MAC address to use for the on-board ethernet.
ip	IP address to use when performing udp/ip transactions.
gateway	Currently Unused
netmask	Currently Unused
tftp-server	IP address to use by default for tftp downloads.
tftp-file	File name to request by default for tftp downloads.
tftp-format	File format to use by default for tftp downloads (binary or srec)
password	SHA encrypted password string.

Downloading User Programs

There are two ways in which software can be downloaded to the board from janusROM – serial or tftp (ethernet). In the case of serial the data must be in Motorola s-record format. When using tftp the file can be either raw binary data or srecord format. The commands inside of janusROM are *transfer* (or *tr*, for serial) and *tftp*. You can get more detailed help from within janusROM itself for both of these commands.

Modifying Memory and Registers

Please see the help within janusROM for *peek* and *poke* for reading/writing to any address value. Also see *regdisp* (display registers), *regset* (modify registers) and *disasm* (to disassemble any address range). Additionally, *bc* (block compare), *bm* (block move), *bf* (block fill), and *mm* (memory modify) can be used for changing and updating memory blocks. There may be additional commands, see the built-in help within janusROM.

Debugging Programs

There are two main forms of debugging from within janusROM – tracing and breakpoints. When in trace mode, the user will be returned to the monitor prompt before every instruction is performed. When in trace mode the prompt will change from the normal *janusROM>* to *trace>*. Hitting enter at this new prompt will cause the next instruction to be executed. While in trace mode breakpoints are ignored as they would be redundant. There are two ways to get into trace mode – either by giving the *–t* option to

the *go* command when starting your program or by issuing *cont* command with a *–t* to continue after a breakpoint but with tracing enabled. The same *cont* command can be used when in trace mode to move back to normal execution, the *–t* is a trace mode toggle.

Breakpoints work as expected. The *br* command can be used to add, remove and list all active breakpoints. As the hardware provides no assistance with breakpoints, janusROM uses the TRAP vectors 13 and 14 to perform software breakpoints. As such, no user programs should use these TRAP vectors for their own purposes. The swapping in and out of these TRAP instructions is handled transparently within janusROM and the user will never see the modified instructions. When a breakpoint is hit control will be given back to janusROM and a register display and next-instruction disassembly will be listed. Until the *cont* command is issued to continue running the program, all parts of the monitor are usable as normal.

Hardware Description

This section describes the various hardware resources need to configure and extend the functionality of the board.

Jumper Settings (J4 + J7)

The board has various jumpers to configure its operation. The following table describes their function:

Jumper Set	Pin	Signal	Description		
Boot ROM Select (J7)	1	~CS0-INT	Connect	Boot Device	
	2	~CS0	**1+2**	Onboard flash	
	3	~CS0-EXT	2+3	Boot ROM on Bus Expansion Port	
ColdFire Config (J4)	1	GND	DIV1	DIV0	BCLK/PCLK\
	2	DIV0	CLOSED	CLOSED	¼
	3	GND	CLOSED	OPEN	N/A
	4	DIV1	**OPEN**	**CLOSED**	**½**
			OPEN	OPEN	1/3
	5	GND	FREQ1	FREQ0	Clock Frequency
	6	FREQ0	CLOSED	CLOSED	32-55MHz
	7	GND	CLOSED	OPEN	56-77MHz
	8	FREQ1	**OPEN**	**CLOSED**	**78-90MHz**
			OPEN	OPEN	N/A
	9	GND	PP_CONF	Parallel Port Bits 9-15	
	10	PP_CONF	CLOSED	Parallel port	
			OPEN	**Alternate**	
	11	GND	DP_SIZE1	DP_SIZE0	Data Port Size
	12	DP_SIZE0	CLOSED	CLOSED	32 bits
	13	GND	**CLOSED**	**OPEN**	**8 bits**
	14	DP_SIZE1	OPEN	--	16 bits
	15	GND	AUTO_ACK	Wait State Config	
	16	AUTO_ACK	CLOSED	Disabled	
			Open	**15 wait states**	

NOTE: Bold selections indicate board defaults as shipped.

Memory Map

The table below describes the arrangement of devices on the processor data bus:

Address Range		Number of Bits	Description
Start	End		
0x00000000	0x0FFFFFFF	8	Flash
0x10000000	0x1FFFFFFF	32	SDRAM
0x20000000	0x3FFFFFFF	32	Peripheral Space
0x40000000	0xFFFF7FFF	-	Unused
0xFFFF8000	0xFFFFFFFF	32	Internal SRAM

Parallel Port (J1)

The parallel port provides a flexible 8-bit bi-directional interface to external devices. These signals are also connected to LED's to aid in system debugging. The parallel port signals are contained in the Bus Expansion Control Port Connector J1. These signals are listed in the following table:

Parallel Port Bit	J1 Pin	LED
7	2	L8
6	4	L7
5	6	L6
4	8	L5
3	10	L4
2	12	L3
1	14	L2
0	16	L1

Bus Expansion Port (J2)

This port extends the processor bus to external devices. Use this port to connect external devices to the processor data bus. The specific signals are listed in the following table:

Pin Name	J2 Pin	Pin Name	J2 Pin
GND	1	~AS	2
R/~W	3	~TA	4
A1	5	A0	6
A3	7	A2	8
A5	9	A4	10
A7	11	A6	12
A9	13	A8	14
A11	15	A10	16
A13	17	A12	18
A15	19	A14	20
A17	21	A16	22
A19	23	A18	24
A21	25	A20	26
A23	27	A22	28
D30	29	D31	30
D28	31	D29	32

D26	33	D27	34
D24	35	D25	36
D22	37	D23	38
D20	39	D21	40
D18	41	D19	42
D16	43	D17	44
D14	45	D15	46
D12	47	D13	48
D10	49	D11	50
D8	51	D9	52
D6	53	D7	54
D4	55	D5	56
D2	57	D3	58
D0	59	D1	60

Bus Expansion Control Port (J1)

This port extends the processor bus to external devices. Use this port to connect external devices to the processor data bus. The specific signals are listed in the following table:

Pin Name	J1 Pin	Pin Name	J1 Pin
TIN1	1		2
TOUT1	3		4
TIN0	5		6
TOUT0	7		8
	9		10
	11		12
	13		14
3.3V	15		16
~CS0_EXT	17	~BG	18
~BWE0	19	~BR	20
~BWE1	21	~BD	22
~BWE2	23		24
~BWE3	25	~TS	26
~OE	27	SIZ0	28
~CS6	29	SIZ1	30
~CS7	31	CK-PORT	32
~RESET	33	GND	34

Board Anomalies

The following are a list of known anomalies that must be considered when designing with this board.

1. IRQ1 is in an unknown state and must be masked out in the IMASK register