

EBDI-Lite
Documentation

Version 2.10

January 1998

Table of Contents

EBDI-Lite Overview	3
Getting Started	4
Appendix 1: Connector, Jumpers	5
Appendix 2: Development Tools	8

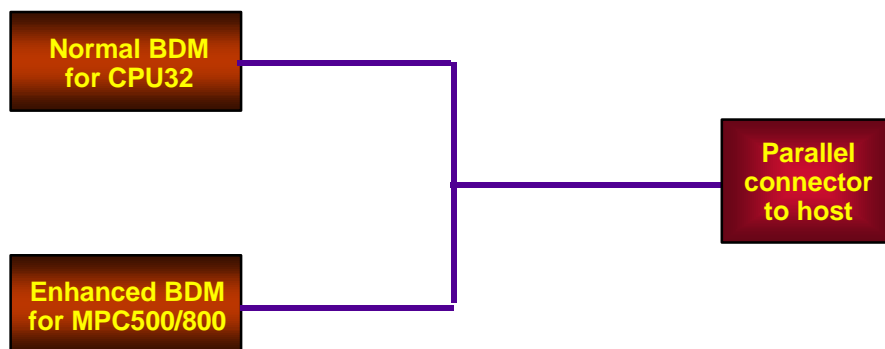
EBDI-Lite Overview

The EBDI-Lite is a minimum interface from the parallel port of a standard PC to a microcontroller based target.

The following microcontrollers are currently supported:

- MPC500 RISC microcontrollers
- MPC800 RISC microcontrollers
- CPU32(+) CISC microcontrollers
- ColdFire microcontrollers

Below is a logical block diagram of the EBDI-Lite.



Features:

- Parallel channel to communicate with the host
- Enhanced BDM for the MPC500/800 RISC microcontrollers
- Normal BDM for the CPU32 and ColdFire based CISC microcontrollers
- Uses the EBDS or some high level software on the PC to communicate with the target

Getting Started

For the first steps the following things are needed:

1. a target based on a microcontroller with a background debugger
2. a power supply for the target
3. a debug interface like the EBDI-Lite (a “cable”)
4. a standard PC as host system
5. some kind of debug software like EBDS or some high level software.

Please follow the list below to build an evaluation system using the EBDI-Lite. For a description of the connectors and the correct settings of the jumpers refer to appendix 1.

1. Power up and boot the pc.
2. Unpack the EBDI-Lite.
3. Refer to appendix 1 to check the target selector jumper on the interface.
4. Connect the flat ribbon cable of the EBDI-Lite to the debug interface connector on the target. Make sure to correctly interface to pin 1.
5. Connect the EBDI-Lite to the parallel port on the pc.
6. Be sure that the host is already running before you now power up the target.
7. Start the debug software on the pc. If you are using the old BD32 software you have to set the port-variable to icd.
8. Make sure that the target has entered the background mode. The RUN led on the cable or the target should be off.
9. Make sure that the target is stopped before you issue any commands to it.

Appendix 1:

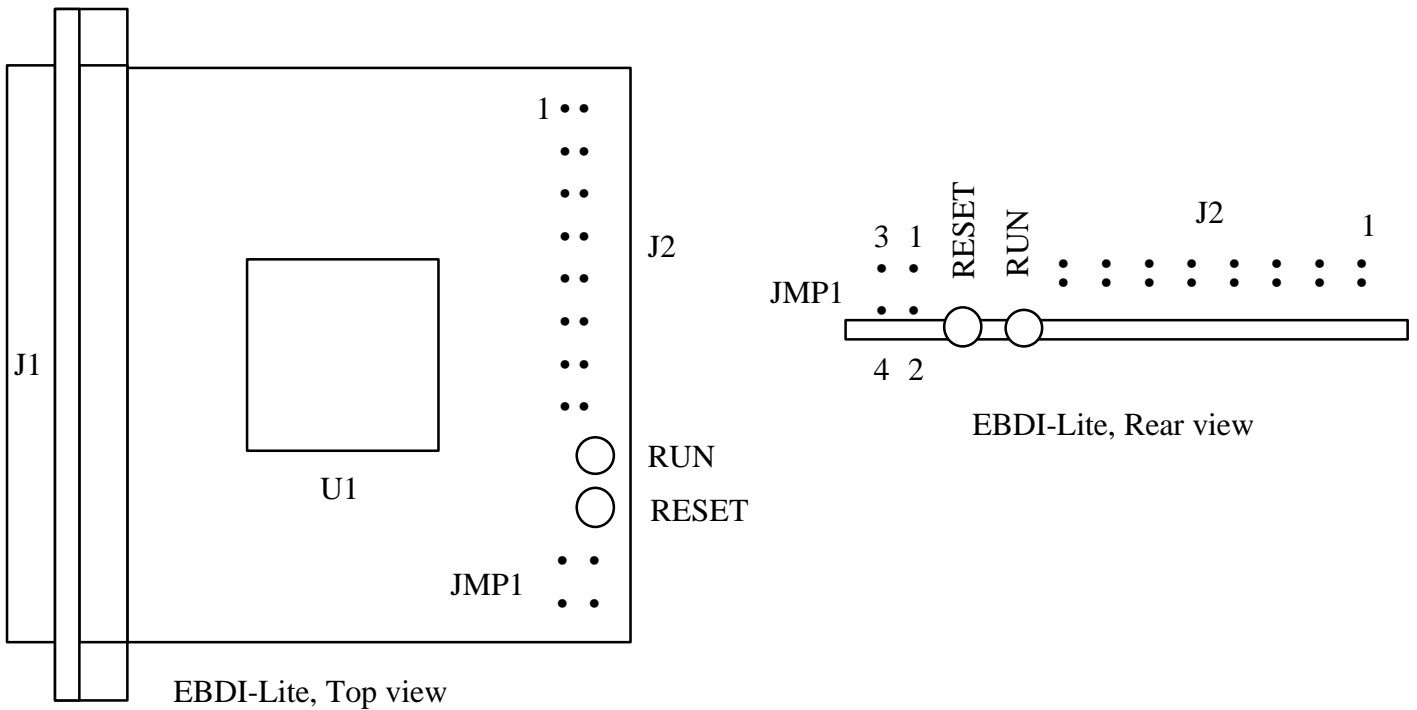
Connectors, Jumpers

Version 2.01

Januar 1998

Connectors, Jumpers

Below is a simplified diagram of the EBDI-Lite pcb.



JMP1: Select the type of microcontroller on the target

	3 - 4	1 - 2
CPU32	open	open
MPC500/800	open	shortened
ColdFire	shortened	open
unused	shortened	shortened

BDM Connectors

MPC500/800

16 pin Berg

Pin	Dir	Name
1	o	VFLS0
2	o	RESET_O
3	-	GND
4	i	DSCLK
5	-	GND
6	o	VFLS1
7	i/o	RESET*
8	i	DSI
9	-	3,3V(+5V)
10	o	DSO
11	o	WP0
12	o	WP1
13	o	WP2
14	o	WP3
15	o	WP4
16	o	WP5

**Pins 10 - 16
are optional !**

CPU32

10 pin Berg

Pin	Dir	Name
1	o	DS*
2	i/o	BERR*
3	-	GND
4	i	DSCLK
5	-	GND
6	o	FREEZE
7	i/o	RESET*
8	i	DSI
9	-	+5V
10	o	DSO

ColdFire

26 pin Berg

Pin	Dir	Name
1	o	CLK_CPU
2	i	BKPT*
3	-	GND
4	i	DSCLK
5	-	GND
6	-	res. (FREEZE)
7	i/o	RESET*
8	i	DSI
9	-	+5V
10	o	DSO
11	-	GND
12	o	PST3
13	o	PST2
14	o	PST1
15	o	PST0
16	o	DDATA3
17	o	DDATA2
18	o	DDATA1
19	o	DDATA0
20	-	GND
21	-	res.
22	-	res.
23	-	GND
24	-	res.
25	-	VCC_CPU
26	i	TEA*

Appendix 2:

Development Tools

Version 2.01

Januar 1998

Debugging Software (1 of 5)

The Enhanced Background Debug Software EBDS

- ▼ Running under MS-DOS
- ▼ All information is presented in windows
- ▼ User interface supports mouse and macros
- ▼ Via EBDI you get:
 - ▼ normal background debugging for CPU32 based microcontrollers
 - ▼ background debugging for the ColdFire
 - ▼ background debugging for M*Core based Devices
 - ▼ enhanced background debugging for MPC500/MPC800 based microcontrollers
 - ▼ JTAG debugging for MPC600 based microprocessors
- ▼ Future Enhancements:
 - ▼ support for HC12, ...
 - ▼ port to Windows95 and Windows NT

Debugging Software (2 of 5)

The GNU Software from the FSF

- ▼ Running under MS-DOS and X-Windows
- ▼ Raw commandline interface
- ▼ Free of charge
- ▼ Not recommended for commercial use

Debugging Software (3 of 5)

The SingleStep from SDS

- ▼ Running under MS-Windows and X-Windows
- ▼ Fully integrated environment: Editor, Compiler, Assembler, Linker and Debugger
- ▼ Simulator, Target Monitor and JTAG/Background version
- ▼ Expandable with user supplied DLL e.g. for simulation of peripherals
- ▼ Uses the DiabData C-Compiler

Debugging Software (4 of 5)

The XRAY from MRI

- ▼ Running under MS-Windows and X-Windows
- ▼ Fully integrated environment: Editor, Compiler, Assembler, Linker and Debugger
- ▼ Simulator, Target Monitor and JTAG/Background version
- ▼ Expandable with user supplied DLL e.g. for simulation of peripherals
- ▼ Uses the DiabData C-Compiler

Debugging Software (5 of 5)

The Hi-Cross+/Hi-Wave from Hiware

- ▼ Running under MS-Windows only
- ▼ Fully integrated environment: Editor, Compiler, Assembler, Linker and Debugger
- ▼ Simulator, Target Monitor and Background version
- ▼ Expandable with user supplied DLL e.g. for simulation of peripherals