SIMTEC
ELECTRONICS

EB675001DIP User Guide

Simtec Electronics

B J Dooks
V R Sanders

EB675001DIP User Guide

Simtec Electronics
by B J Dooks and V R Sanders

- ARM® is a registered trademark of ARM Limited.
- LINUX® is a registered trademark of Linus Torvalds.
- UNIX® is a registerd trademark of The Open Group.
- All other trademarks are acknowledged.

The product described in this document is under continuous development and improvement. All descriptions of usage and particulars of the product are given in good faith by Simtec Electronics. However all warranties expressed or implied, including but not limited to implied warranties or merchantability, or fitness for purpose, are excluded.

While every precaution has been taken in the preparation of this book, the publisher assumes no responsibility for errors or omissions, or for damages resulting from the use of the information contained herein.

Revision History

Revision 1.00	20th July 2005	VRS
Initial Release.		
Revision 1.01	27th July 2005	VRS
Formatting updates and improvements.		
Revision 1.1	9th January 2006	VRS
Improve design guide section.		
Revision 1.11	4th April 2006	VRS
Formatting updates and improvements.		
Revision 1.2	12th April 2007	VRS
Fix user CPLD pin numbering in Table 2.1, “60 way PTH connector PL3”.		
Revision 1.3	5th October 2007	VRS
Clarify IRQ0 use on the Section 2.6, “DM9000 Network controller” and Section 2.5, “Real Time Clock”.		

Table of Contents

List of Figures

List of Tables

List of Examples

EB675001DIP Development Board

About this document. This document describes the Simtec EB675001DIP Development board which provides a flexible devlopemnt system for both experimentation and intergrator solutions.

Intended Audience. This document is aimed at experienced engineers.

Related documents. Some additional documents which may be useful:

Bootstraping EB675001DIP [http://www.simtec.co.uk/products/EB675001DIP/files/EB675001DIP-bootstrap.html]
Connector and link pinouts. [http://www.simtec.co.uk/products/EB675001DIP/files/pinlist.html]
Memory map and control registers. [http://www.simtec.co.uk/products/EB675001DIP/files/mmap.html]
Mechanical Drawing [http://www.simtec.co.uk/products/EB675001DIP/files/EB675001DIP-mechanical.pdf]

Feedback. Any suggestions, comments or corrections concerning this document are welcomed, please contact Simtec Electronics giving:

The document title
The document revision
A clear explanation of your comments and how they apply

Chapter 1. Overview

This chapter describes:

- The kit contents
- Development tools
- Use of the development board
- Handling precautions

1.1. Kit contents

The EB675001DIP is a comprehensive ARM computing platform. The Kit contains:

- The EB675001DIP user guide
- A CD-ROM containing development software and documents relevant to the EB675001DIP
- The EB675001DIP board.

1.1.1. CD-ROM

The CD-ROM contains:

- A copy of all the freely available documentation, including this user guide.
- Datasheets for all major components used
- uCLinux distribution
- x86 cross building toolchain for GNU/Linux
- ABLE bootloader

The toolchain contains a GCC compiler, assembler and linker suitable for cross compiling ARM binaries from an x86 machine running GNU/Linux.

1.1.2. EB675001DIP Development board

The EB675001DIP board (gold specification) has the following major components:

- OKI ML675001 MCU
- 32MB SDRAM
- 16MBit NOR Flash
- 8Kbit I2C EEPROM
- Xilinx XL9572XL CPLD
- 10/100MBit Ethernet controller
- 9pin D-sub RS232 port

- JTAG header
- 30 5V tolerant GPIO or special function lines
- 40 IO lines from Xilinx XL9572XL CPLD
- Wide input voltage regulator

1.2. Development Tools

The development tools provided must be installed and run on a PC with a GNU/Linux Operating system (e.g. Debian, Ubuntu or Redhat distributions). The GCC toolchain provided creates executables that can be run on the EB675001DIP. This toolchain is also required to build uCLinux. Full details of building uCLinux are provided in notes [http://www.simtec.co.uk/products/EB675001DIP/files/uclinux-20041215-notes.html] on the EB675001DIP resources page.

In order to run the development tools the host PC requires:

- 500MHz or faster processor
- Installed GNU/Linux distribution
- 128MB RAM
- 1GB of hard disk space (3GB if building uCLinux)
- CD-ROM Drive

In addition to the development tools, the module usually requires serial communications to access its bootloader and booted system. Most modern PC have serial ports, however increasingly only USB ports are provided, most typical USB->Serial converters appear to work with the EB675001DIP.

1.3. Using the development board

The EB675001DIP is a complete system. With the addition of a suitable power supply and PC the Kit provides everything required to start producing the desired solution. There are generally two modes of operation:

- The first most basic method of using the EB675001DIP, and indeed only method on boards other than the gold specification, is to use the in built OKI downloader utility to transfer programs to be executed using raw hex records.
- The second method uses a full ABLE bootloader install and allows much more flexible use of the EB675001DIP. ABLE can start uCLinux images from a variety of sources including the network. In this mode of usage the serial port is used as a console to communicate with the device.

1.4. Handling precautions

This development board is intended for use either within a workshop/laboratory environment or as an integrator solution within a larger product. Because of this the EB675001DIP board is supplied without an enclosure. The lack of an enclosure means that standard electrostatic control procedures should be used when handling the board. When using the EB675001DIP outside an enclosure the following is recommended:

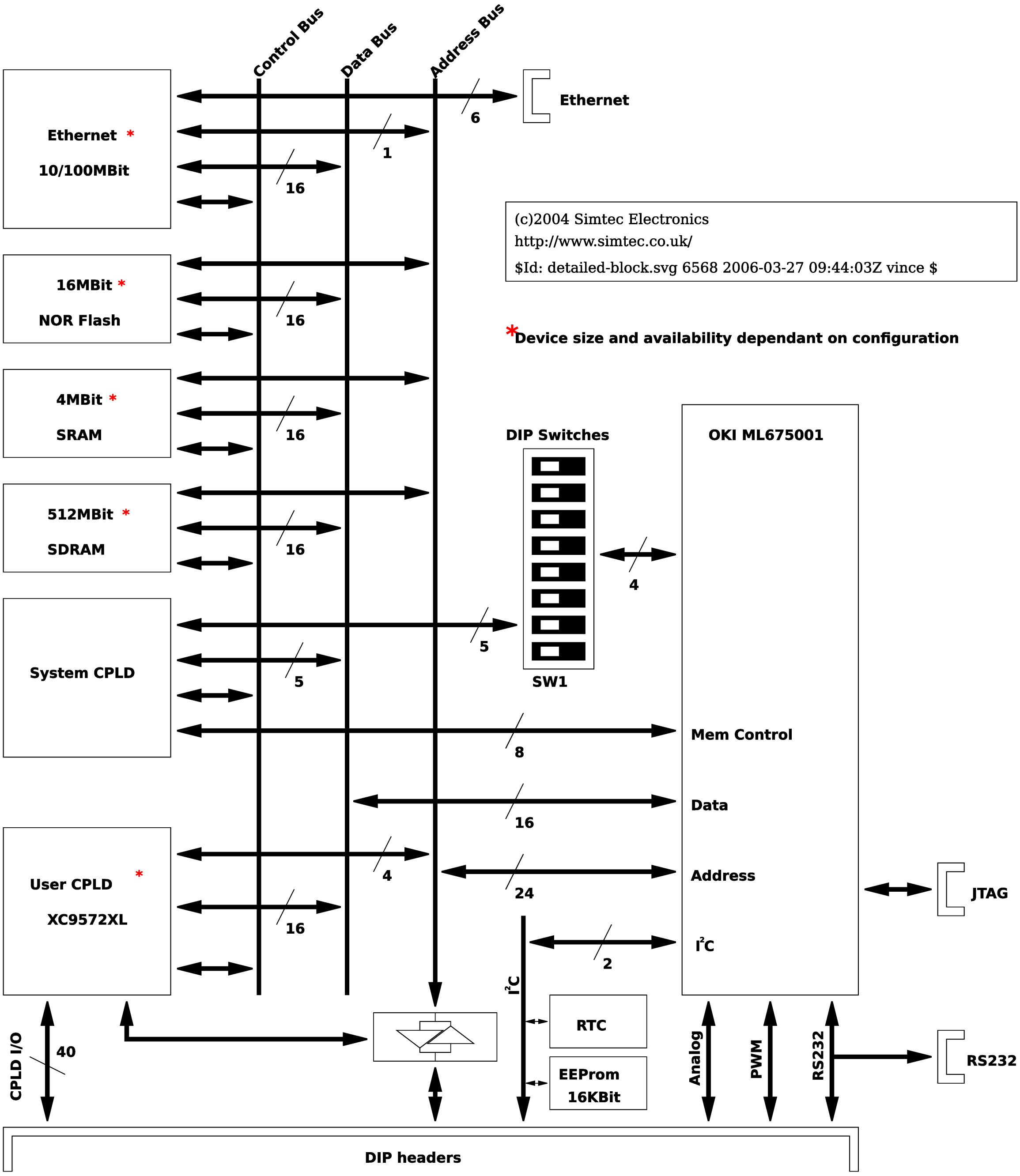
- Only hold the board by the edges
- Always use proper static handling equipment, as a minimum an earth strap

Chapter 2. Hardware Description

2.1. Overview

The EB675001DIP is a complete system that includes a large number of Input/Output facilities. The addition of the user programmable logic, in the form of the 64 Macrocell Xilinx CPLD, provides for an extremely large number of applications. This logic may be used to create quite complex designs relatively easily using freely available Xilinx tools. Simtec Electronics provide a number of application notes accessible via the EB675001DIP resources page or on the supplied CD-ROM, several of which demonstrate the use of the CPLD.

Figure 2.1. Detailed block diagram of the EB675001DIP



2.2. OKI ML675001 MCU

The ML675001 is a 144-pin plastic LFBGA System on Chip from OKI Semiconductor. The large number of peripherals within the device give flexibility to the user and the ability to use numerous I/O solutions without additional controllers.

The ML675001 does not have a traditional MMU and lacks the ability to operate with virtual memory, because of this only Operating systems which do not require an MMU, such as uCLinux, are suitable for use with this processor.

To improve performance the ML675001 has a cache controller which allows for greatly improved performance over devices without such capabilities.

The ML675001DIP has a number of flexible peripherals and interfaces, these are summarised as follows:

- ARM7TDMI 32-bit RISC CPU
- 32-bit mode (ARM) and/or 16-bit mode (Thumb)
- Built-in SDRAM external memory controller supports glueless connectivity to memory.
- External memory controller supporting external NOR flash and SRAM
- 32-KBytes built in zero-wait-state SRAM
- One fast interrupt source
- 23 internal interrupt sources
- 4 external interrupt sources
- Independent masking and priority settings for all interrupt sources
- Two DMA channels with external access
- One 16-bit system timer
- Six 16-bit auto reload timers with independent clock settings
- One flexible 16 bit dual-stage watchdog timer
- Two 16-bit PWM channels
- GPIO: 42 bits
- Four 10-bit Analog channels
- A single synchronous serial controller with master or slave operation
- Master mode I^{2}C controller
- A 16550 compatible asynchronous communications controller (UART) with integral baud rate generator.
- Simple serial controller with built in baud rate generator
- Flexible power management including standby and halt operation modes
- Built-in boot ROM accommodates in-circuit Flash ROM re-programming and field-updates

Most of these features are directly available to the user of the EB675001DIP and where appropriate have suitable external devices connected for extended functionality. The connection of the external devices are described in the remainder of this chapter.

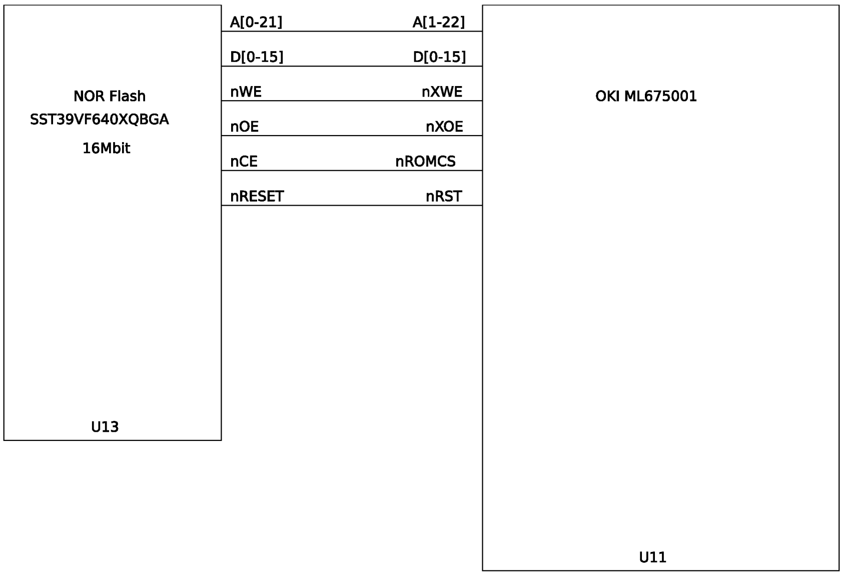
For further details on the ML675001 please consult the OKI ML675001 User manual [http://www2.okisemi.com/site/productscatalog/armsolutions/mcumpu/availabledocs/Intro-9980.html].

2.3. Memory

2.3.1. NOR boot flash

The EB675001DIP has provision for a single NOR flash device. This device typically contains the bootloader and uC-Linux image to boot, though it can contain anything the user requires. The flash is implemented as a single sixteen bit wide device, which is selected by use of the nROMCS chip select from the ML675001. This memory appears in bank 25 of the ML675001 memory map.

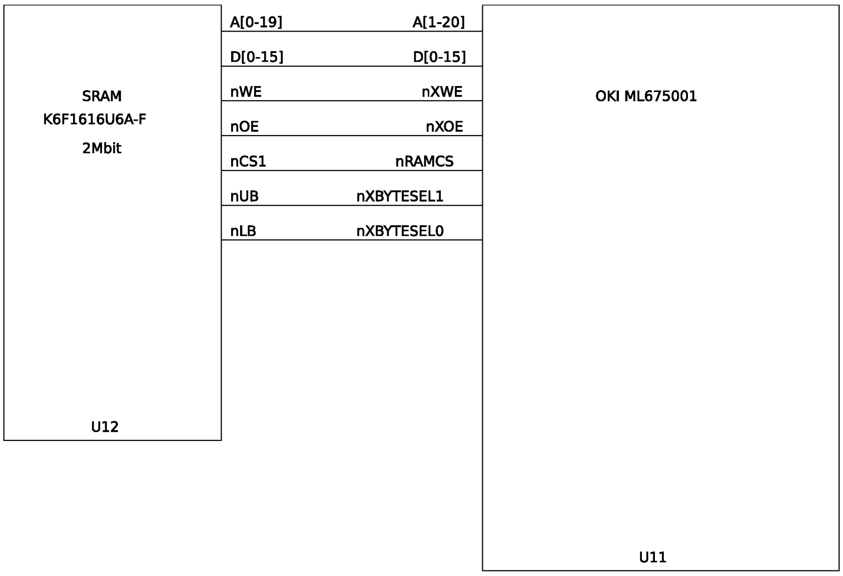
Figure 2.2. NOR flash to ML675001 attachment



2.3.2. SRAM

The EB675001DIP has provision for an SRAM device, this is in addition to the 32KB of internal zero wait state memory that is internal to the ML675001. This memory is typically only fitted in the silver configuration of the board. The SRAM is implemented as a single sixteen bit wide device, which is selected by use of the nRAMCS chip select from the ML675001. This memory appears in bank 26 of the ML675001 memory map.

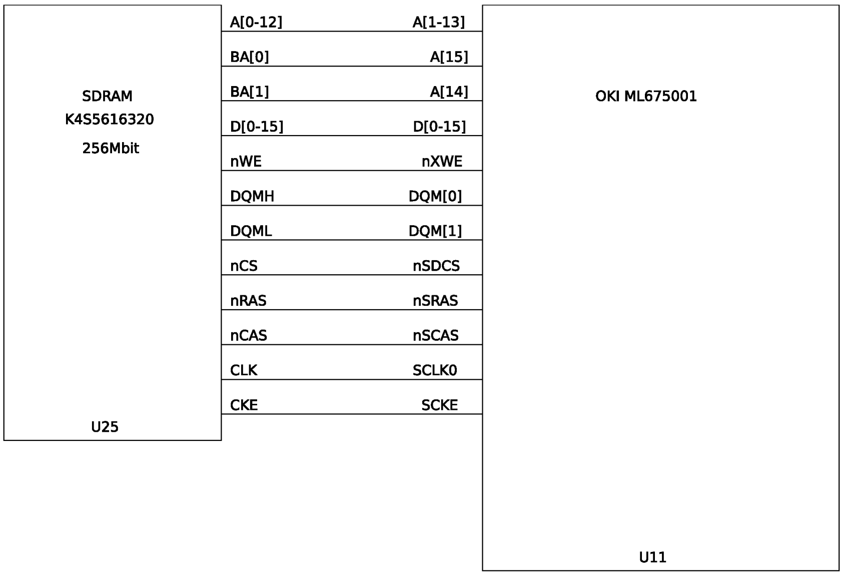
Figure 2.3. SRAM to ML675001 attachment



2.3.3. SDRAM

The EB675001DIP has provision for a single SDRAM device with either the default 256MBit capacity or a 512MBit capacity. This memory, where fitted, is accessed using the ML675001 SDRAM controller and appears in bank 24 of its memory map.

Figure 2.4. SDRAM to ML675001 attachment



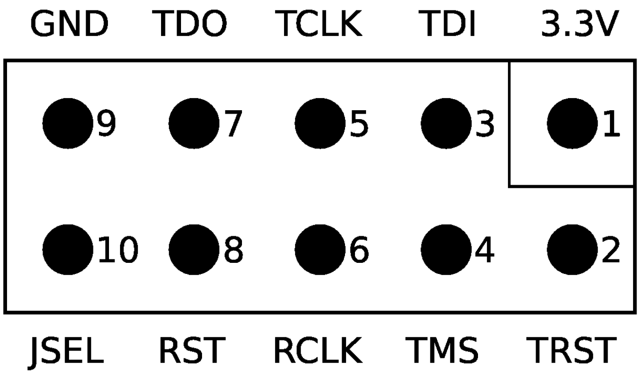
2.3.4. EEPROM

There is provision for a single EEPROM connected via the I^2C bus. All specification boards have an 8KBit (1KB) device fitted as standard but up to 256KBit can be accommodated. The device is typically used to hold the non volatile settings in the ABLE bootloader.

2.4. JTAG

The ML675001, system CPLD and user CPLD are all connected to the JTAG chain. The JTAG chain is available from this connector and allows reprogramming of the user CPLD and ICE debugging of the processor.

Figure 2.5. JTAG connector



The JTAG chain on the EB675001DIP is connected to the ML675001, the system CPLD and the user CPLD. Because of this care should be taken to ensure other devices are placed in bypass as required. BSDL files are available from OKI for the ML675001 and Xilinx for the CPLD devices.

2.5. Real Time Clock

The Real Time Clock (RTC) is connected to the ML675001 on the I^{2}C bus. The RTC is a Ricoh R2051 with a 32KHz crystal attached. This device has the capacity to hold the date and time and keep time using a minimum of power, typically 0.4uA with 3V supply.

There is also the ability to set alarms which may generate interrupts, this may be used to perform repeating tasks with long intervals without consuming large amounts of power.

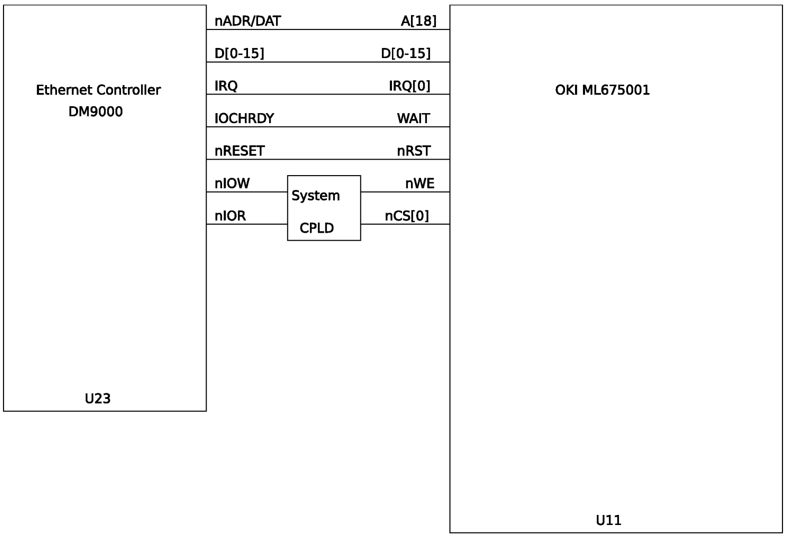
The R2051 interrupt is connected to the ML675001 IRQ0. This IRQ is shared with the Davicom ethernet controller and is also present on A13 of the PL4 connector.

2.6. DM9000 Network controller

The DM9000 network controller provides 100Mbit Ethernet connectivity to the system. It is decoded into the CS0 address space by the system CPLD as it uses the WAIT line to extend I/O cycles as necessary.

The DM9000 may interrupt the ML675001. The IRQ is connected to the ML675001 IRQ0 signal. The interrupt is shared with the Real Time Clock and is also presented on pin A13 of the PL4 connector.

Figure 2.6. DM9000 to ML675001 attachment



2.7. Xilinx CPLD

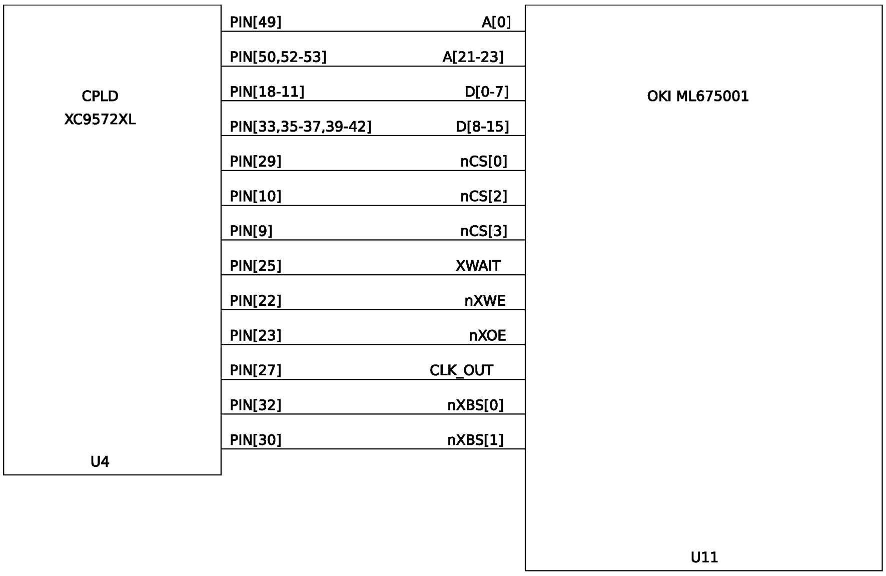
The EB675001DIP has a powerful addition to a development system in the form of a user programmable CPLD. This device can be programed to perform a large number of logic functions.

The application notes contain several examples of its flexibility in everything from a simple address decoder to a TV resolution video framebuffer.

Enough signals are directly connected from the CPU to the CPLD to allow for eight or sixteen bit wide accesses in three chip select regions. CS0 must be treated with care because it is pre decoded by the system CPLD to attach the DM9000 Ethernet controller.

The system wait line is also taken through the user CPLD to allow the CPLD to force I/O wait in CS0 if necessary. More details on the CPLD use and configuration can be found in the User CPLD guide [http://www.simtec.co.uk/products/EB675001DIP/files/cpld/cpld-guide.html]

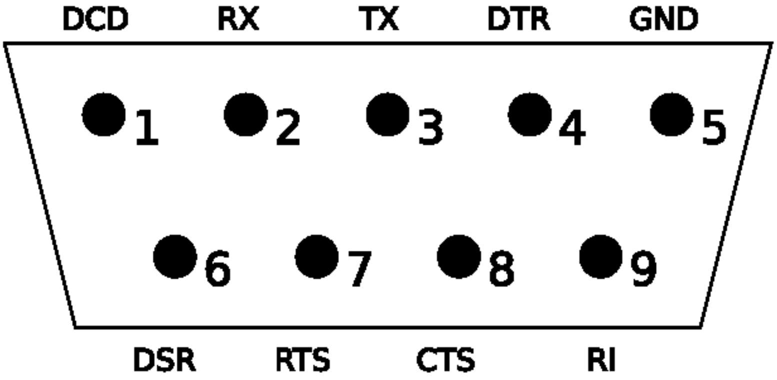
Figure 2.7. Xilinx XL9572XL to ML675001 attachment



2.8. Serial port

The ML675001 has a single 16550 type serial port complete with FIFO. This port is used as a console port by the bootloader and uCLinux. The port is level translated to RS232 signals by a MAX3243 device. The port is switchable from the buffered outputs to the unbuffered LVTTL levels by using the SERIAL_EN line. For more details on this ports usage refer to the EB675001DIP Connector and link pinouts [http://www.simtec.co.uk/products/EB675001DIP/files/pinlist.html] document.

Figure 2.8. Serial connector



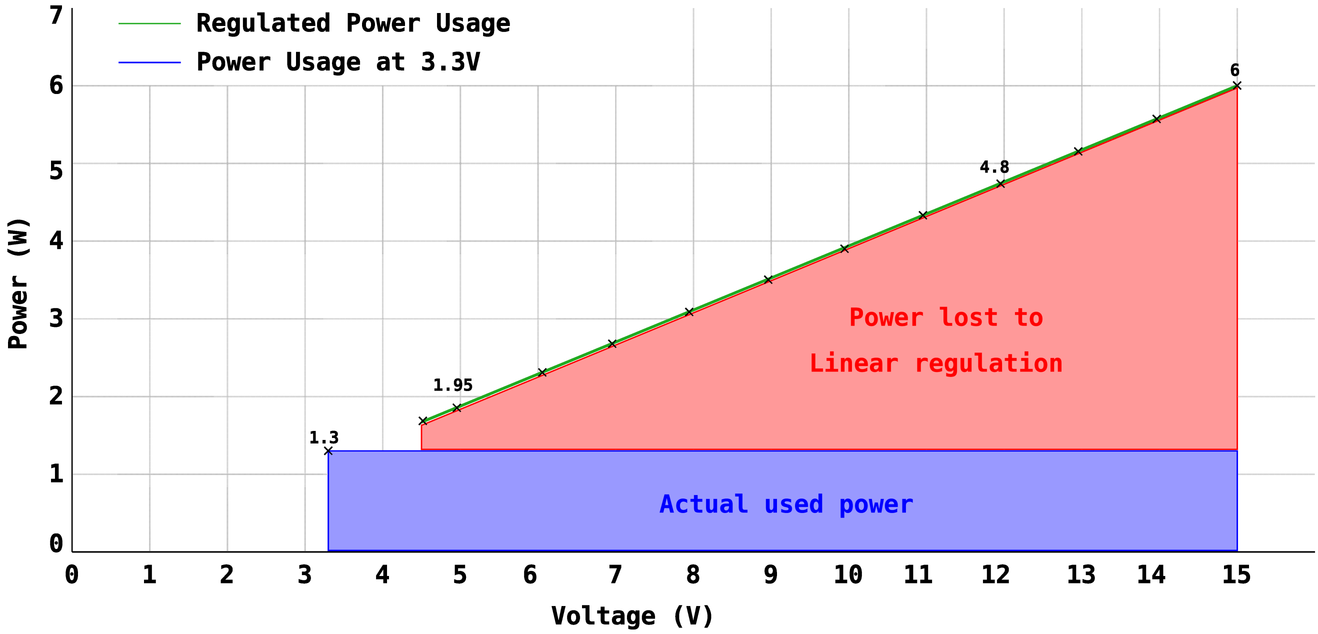
2.9. Power Supply

The module may be powered either with a direct regulated 3.3V supply capable of providing 0.5A or by a linearly regulated input with a range of 4.5V - 15V. Practical measurements taken from a gold specification module running uCLinux with active networking, serial and basic user CPLD code loaded, gives a baseline current draw of 0.39A at 3.3V (1.29W).

Measurements using the linearly regulated input give the expected results of a 0.39A current usage at any point of the acceptable input voltage range, the excess power being dissipated as heat into the modules ground plane.

All figures are subject to a 1% measurement accuracy and are for guidance purposes only.

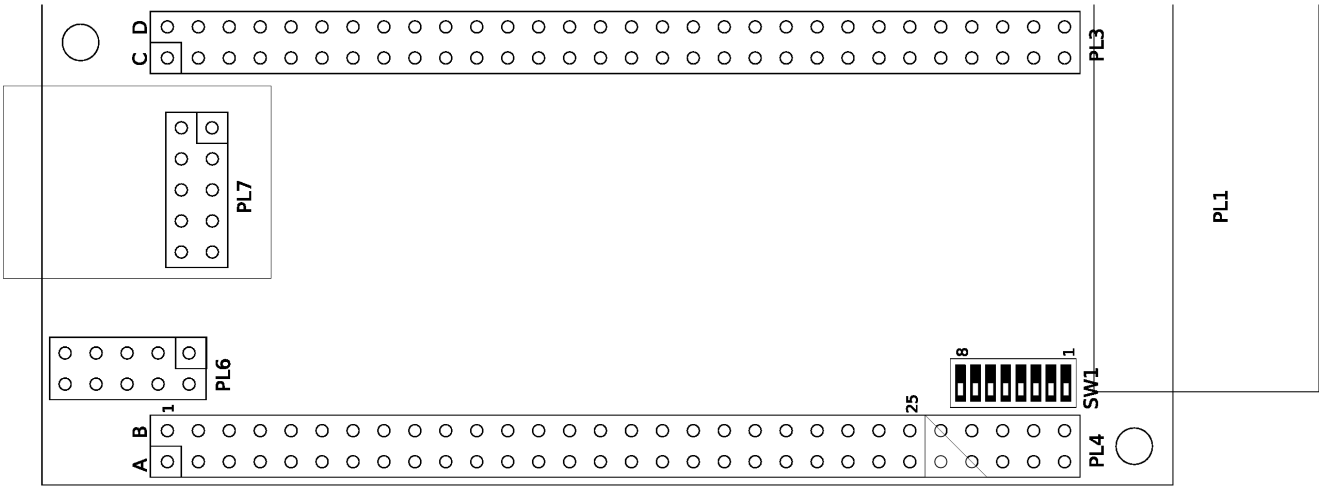
Figure 2.9. Graph of EB675001DIP power usage



2.10. Expansion connectors

The EB675001DIP has two rows of standard 0.1inch Plated Through Hole (PTH) connectors, sometimes referred to as a QDIP configuration.

Figure 2.10. Expansion connector placement



These connectors provide all the I/O signals to expand the use of the module. For full details the EB675001DIP Connector and link pinouts [http://www.simtec.co.uk/products/EB675001DIP/files/pinlist.html] document should be consulted, this contains addition information and comments relevant to using this product.

Table 2.1. 60 way PTH connector PL3

Description	Name	Pin	Pin	Name	Description
3.3V output from on-board regulator or external 3.3V regulated smoothed supply	3.3V	C1	D1	GND	Signal Ground
Supply Ground	GND	C2	D2	EXT_BAT	Supply for the Real Time Clock

Description	Name	Pin	Pin	Name	Description
Xilinx XC9572XL I/O Pin 92	CPLD_PIN92	C3	D3	CPLD_PIN91	Xilinx XC9572XL I/O Pin 91
Xilinx XC9572XL I/O Pin 89	CPLD_PIN89	C4	D4	CPLD_PIN90	Xilinx XC9572XL I/O Pin 90
Xilinx XC9572XL I/O Pin 86	CPLD_PIN86	C5	D5	CPLD_PIN87	Xilinx XC9572XL I/O Pin 87
Xilinx XC9572XL I/O Pin 82	CPLD_PIN82	C6	D6	CPLD_PIN85	Xilinx XC9572XL I/O Pin 85
Xilinx XC9572XL I/O Pin 79	CPLD_PIN79	C7	D7	CPLD_PIN81	Xilinx XC9572XL I/O Pin 81
Xilinx XC9572XL I/O Pin 77	CPLD_PIN77	C8	D8	CPLD_PIN78	Xilinx XC9572XL I/O Pin 78
Xilinx XC9572XL I/O Pin 74	CPLD_PIN74	C9	D9	CPLD_PIN76	Xilinx XC9572XL I/O Pin 76
Xilinx XC9572XL I/O Pin 71	CPLD_PIN71	C10	D10	CPLD_PIN72	Xilinx XC9572XL I/O Pin 72
Xilinx XC9572XL I/O Pin 68	CPLD_PIN68	C11	D11	CPLD_PIN70	Xilinx XC9572XL I/O Pin 70
Xilinx XC9572XL I/O Pin 66	CPLD_PIN66	C12	D12	CPLD_PIN67	Xilinx XC9572XL I/O Pin 67
Xilinx XC9572XL I/O Pin 64	CPLD_PIN64	C13	D13	CPLD_PIN65	Xilinx XC9572XL I/O Pin 65
Xilinx XC9572XL I/O Pin 61	CPLD_PIN61	C14	D14	CPLD_PIN63	Xilinx XC9572XL I/O Pin 63
Buffered CPU address line	SA[23]	C15	D15	CPLD_PIN60	Xilinx XC9572XL I/O Pin 60
Buffered CPU address line	SA[22]	C16	D16	CPLD_PIN59	Xilinx XC9572XL I/O Pin 59
Buffered CPU address line	SA[21]	C17	D17	GND	Signal ground
Buffered CPU address line	SA[20]	C18	D18	CPLD_PIN58	Xilinx XC9572XL I/O Pin 58
Buffered CPU address line	SA[19]	C19	D19	CPLD_PIN56	Xilinx XC9572XL I/O Pin 56
Buffered CPU address line	SA[18]	C20	D20	CPLD_PIN55	Xilinx XC9572XL I/O Pin 55
Buffered CPU address line	SA[17]	C21	D21	SA[8]	Buffered CPU address line
Buffered CPU address line	SA[16]	C22	D22	SA[7]	Buffered CPU address line
Buffered CPU address line	SA[15]	C23	D23	SA[6]	Buffered CPU address line
Buffered CPU address line	SA[14]	C24	D24	SA[5]	Buffered CPU address line
Buffered CPU address line	SA[13]	C25	D25	SA[4]	Buffered CPU address line
Buffered CPU address line	SA[12]	C26	D26	SA[3]	Buffered CPU address line
Buffered CPU address line	SA[11]	C27	D27	SA[2]	Buffered CPU address line
Buffered CPU address line	SA[10]	C28	D28	SA[1]	Buffered CPU address line
Buffered CPU address line	SA[9]	C29	D29	CPLD_PIN54	Xilinx XC9572XL I/O Pin 54

Description	Name	Pin	Pin	Name	Description
Signal Ground	GND	C30	D30	GND	Signal Ground (Key pin)

Table 2.2. 60 way PTH connector PL4

Description	Name	Pin	Pin	Name	Description
4.5 - 15V DC power supply	VIN	A1	B1	GND	Power ground
Power ground	GND	A2	B2	3.3V	3.3V output from on-board regulator or external 3.3V regulated smoothed supply
Xilinx XC9572XL I/O Pin 94	CPLD_PIN94	A3	B3	CPLD_PIN93	Xilinx XC9572XL I/O Pin 93
Xilinx XC9572XL I/O Pin 96	CPLD_PIN96	A4	B4	CPLD_PIN95	Xilinx XC9572XL I/O Pin 95
Xilinx XC9572XL I/O Pin 1	CPLD_PIN01	A5	B5	CPLD_PIN97	Xilinx XC9572XL I/O Pin 97
Xilinx XC9572XL I/O Pin 4	CPLD_PIN04	A6	B6	CPLD_PIN03	Xilinx XC9572XL I/O Pin 3
Xilinx XC9572XL I/O Pin 8	CPLD_PIN08	A7	B7	CPLD_PIN06	Xilinx XC9572XL I/O Pin 6
Inverted module reset output	RST	A8	B8	PWM1	Second Pulse Width Modulator output or CPU GPIO PIOC[1]
Inverted CPU fast interrupt line	FIQ	A9	B9	PWM0	First Pulse Width Modulator output or CPU GPIO PIOC[0]
Fourth CPU interrupt line or CPU GPIO 8	IRQ3	A10	B10	TC1	Terminal count for DMA Chanel 1 or CPU GPIO PIOB[4]
Third CPU interrupt line or CPU GPIO PIOE[7]	IRQ2	A11	B11	TC0	Terminal count for DMA Chanel 0 or CPU GPIO PIOB[5]
Second CPU interrupt line or CPU GPIO 6	IRQ1	A12	B12	DACK1	DMA acknowledge/clear for channel 1 or CPU GPIO PIOB[3]
First CPU interrupt line or CPU GPIO 5	IRQ0	A13	B13	DREQ1	DMA request for channel 1 or CPU GPIO PIOB[2]
I2C Serial Clock or CPU GPIO PIOE[4]	SCL	A14	B14	DACK0	DMA acknowledge/clear for channel 0 or CPU GPIO PIOB[1]
I2C Serial Data or CPU GPIO PIOE[3]	SDA	A15	B15	DREQ0	DMA request for channel 0 or CPU GPIO PIOB[0]
Synchronous serial (SSIO) data output	SDO	A16	B16	SRXD	Simple serial port (SIO) data receive (not the UART port) or CPU GPIO PIOB[7]
Synchronous serial (SSIO) data input	SDI	A17	B17	STXD	Simple serial port (SIO) data transmit or CPU GPIO PIOB[6]
Synchronous serial (SSIO) clock	SCK	A18	B18	RI	Unbuffered UART RI signal or CPU GPIO PIOA[7]

Description	Name	Pin	Pin	Name	Description
Signal ground	GND	A19	B19	RTS	Unbuffered UART RTS signal or CPU GPIO PIOA[6]
Fourth analog input	ANALOG_3	A20	B20	DTR	Unbuffered UART DTR signal or CPU GPIO PIOA[5]
Third analog input	ANALOG_2	A21	B21	DCD	Unbuffered UART DCD signal or CPU GPIO PIOA[4]
Second analog input	ANALOG_1	A22	B22	DSR	Unbuffered UART DSR signal or CPU GPIO PIOA[3]
First analog input	ANALOG_0	A23	B23	CTS	Unbuffered UART CTS signal or CPU GPIO PIOA[2]
Analog inputs ground reference	ANALOG_GND	A24	B24	TX	Unbuffered UART TX signal or CPU GPIO PIOA[1]
Analog inputs supply reference	VDD_ANALOG	A25	B25	RX	Unbuffered UART RX signal or CPU GPIO PIOA[0]
Buffered RS232 level UART DCD signal	RS232_DCD	A26	B26	RS232_DSR	Buffered RS232 level UART DSR signal
Buffered RS232 level UART RX signal	RS232_RX	A27	B27	RS232_RTS	Buffered RS232 level UART RTS signal
Buffered RS232 level UART TX signal	RS232_TX	A28	B28	RS232_CTS	Buffered RS232 level UART CTS signal
Buffered RS232 level UART DTR signal	RS232_DTR	A29	B29	RS232_RI	Buffered RS232 level UART RI signal
Serial port ground	GND	A30	B30	SERIAL_EN	RS232 buffer enable

Chapter 3. Bootloader

3.1. Overview

The Simtec Electronics Advanced Boot Load Environment (ABLE) is a portable modular boot loader for use in applications where an OS must be retrieved and started. ABLE provides extended functionality providing modules for a command line, video consoles, serial consoles, network booting and numerous other facilities.

ABLE is a powerful tool and provides a very flexible environment useful for both development and deployment of systems. ABLE is a boot loader, not an Operating System this distinction can sometimes lead to misunderstandings about the capabilities provided by ABLE. A boot loader in this context is a self contained program which retrieves and starts execution of an Operating System. It does not execute user programs itself (all the CLI commands are built in) and does not provide services to an Operating System once started (PC BIOS perform this role).

The modular nature of ABLE allows the use of the same building blocks for every supported platform. The integration and omission of various modules allow for specific driver sets depending on the peripherals of a platform. The flexibility of this approach allows for a common familiar environment across all supported platforms while still supporting a complete feature set.

This chapter only provides a brief introduction to ABLE. Full documentation can be found in the ABLE user guide [http://www.simtec.co.uk/products/SWABLE/files/able-set/book_userguide.html].

3.2. Getting Started

When a platform is initially powered or a hard reset performed, the ABLE environment will be started and each component module will be loaded in turn. The last module loaded is the ABLE shell, which will present the user with a command line interface.

ABLE has the ability to use a combination of input and output sources to interact with a user. The default is to use all the input and output devices available. For example, on the EB2410ITX both the console serial port and the video display will be used to output and the serial port for input (future versions may support USB keyboards for input).

Example 3.1. Video display after starting ABLE on EB2410ITX

```
selected all-wr for console write stream
selected all-rd for console read stream
DRAM: 128 Mb (134217728 bytes)
BAST: PMU version 1.02, ID 00:01:3d:00:01:6a
ABLE: 2.08 (s3c2410x) (vince@gerald) Fri Apr  8 16:35:26 BST 2005
hdc: TOSHIBA MK1003MAV: ATA PIO mode 4
hdc:Diagnosing disc drive: ok
(hdc) 1GB
(hd0) on ((hdc1):ext2)
(hd1) on (hdc2)
DM9000: dm0: r1, 00:01:3d:00:01:6a int phy, link ok, 100Mbit full duplex
NE2000: ne0: ISA/Generic, 00:01:3d:00:01:6b (EEPROM Invalid / Missing)
TMP101: I2C error (-2)
sys.autoshadow unset, automatically shadowing
>
```

Example 3.2. Serial display after starting ABLE on EB2410ITX

```
SuperIO controller fitted
Initialising
Detecting SDRAM size
SDRAM: BANK6 size 04000000
SDRAM: BANK7 size 04000000
```

```
ABLE: 2.08 (s3c2410x) (vince@gerald) Fri Apr  8 16:35:26 BST 2005
Processor: Samsung S3C2410A (arm920)
System: Machine bast/s3c2410x, Linux id 0x014b
.S3C2410X RTC: 01:46:54, 00/01/2003
NAND: configured boot slot is 0 (card slot)
NAND: found Samsung K9F1208u0a [131072,32,512]
(flash0) on (nand0p1)
(flash1) on ((nand0p2):jffs2)
EEPROM: 24cXX, 1024 bytes, single byte addressed
(nvram0) on (24cxx)
sys.speed is unset, Setting CPU Speed to 266MHz
no configuration, defaulting to VGA
X/Y values invalid, configuring automatically
Chrontel CH7006 detected
screen mode is 640x480, ?Hz, ?Hz HSync
video: video size 300K
configuring ch7006: vga
selected all-wr for console write stream
selected all-rd for console read stream
DRAM: 128 Mb (134217728 bytes)
BAST: PMU version 1.02, ID 00:01:3d:00:01:6a
ABLE: 2.08 (s3c2410x) (vince@gerald) Fri Apr  8 16:35:26 BST 2005
hdc: TOSHIBA MK1003MAV: ATA PIO mode 4
hdc:Diagnosing disc drive: ok
(hdc) 1GB
(hd0) on ((hdc1):ext2)
(hd1) on (hdc2)
DM9000: dm0: r1, 00:01:3d:00:01:6a int phy, link ok, 100Mbit full duplex
NE2000: ne0: ISA/Generic, 00:01:3d:00:01:6b (EEPROM Invalid / Missing)
TMP101: I2C error (-2)
sys.autoshadow unset, automatically shadowing
>
```

The input devices are controlled by using the cons-read parameter and similarly the cons-write parameter controls which output devices are used.

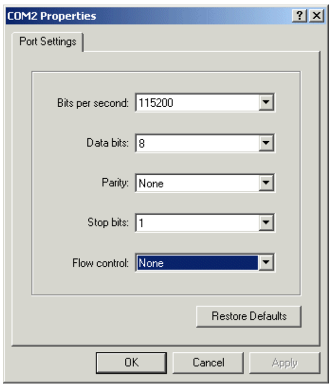
Typically the console serial port is used to interact with the ABLE CLI.

Unless the boot parameters are altered from their default settings the autoboot process will commence. To manually start an Operating System the command line must be used.

3.2.1. Using hyperterm as a serial console

To access the serial console from windows the hyperterm program can be used. Identify which serial port the platform is connected to and ensure a note is made of the correct COM port, e.g. COM1 or COM2.

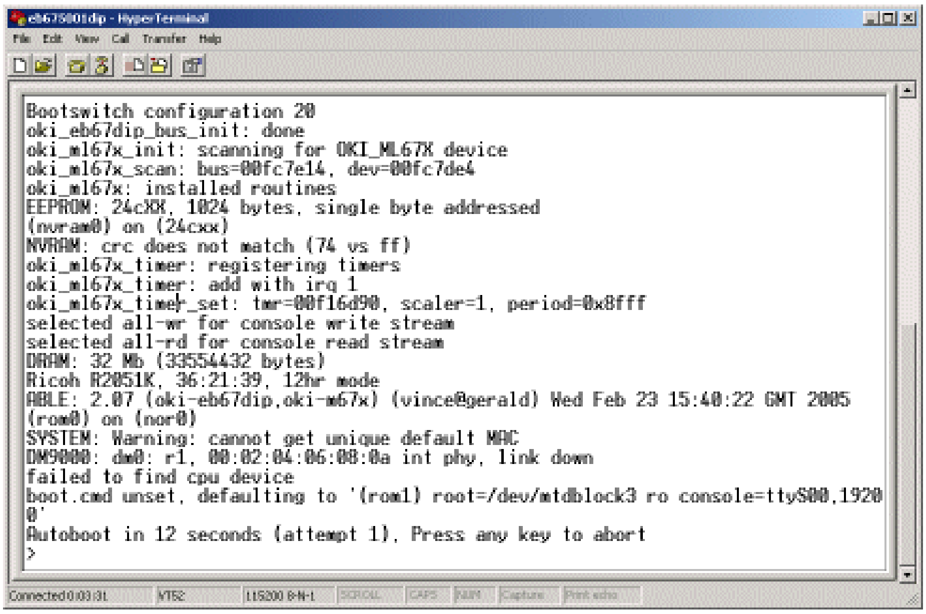
Figure 3.1. Hyperterm settings window



Bits per second: 115200
Data bits: 8
Parity: None
Stop bits: 1
Flow control: None

Start HyperTerminal and create a new connection. When prompted for which modem to use, instead choose the appropriate COM port, as noted earlier. Then the appropriate settings for your platform (please refer to platform specific documentation) typically these settings are 115200 bits per second, 8 data bits, no parity, 1 stop bit and no flow control as shown in Figure 3.1, “Hyperterm settings window”.

Figure 3.2. Hyperterm displaying ABLE output



eb675001dip - HyperTerminal
File Edit View Call Transfer Help

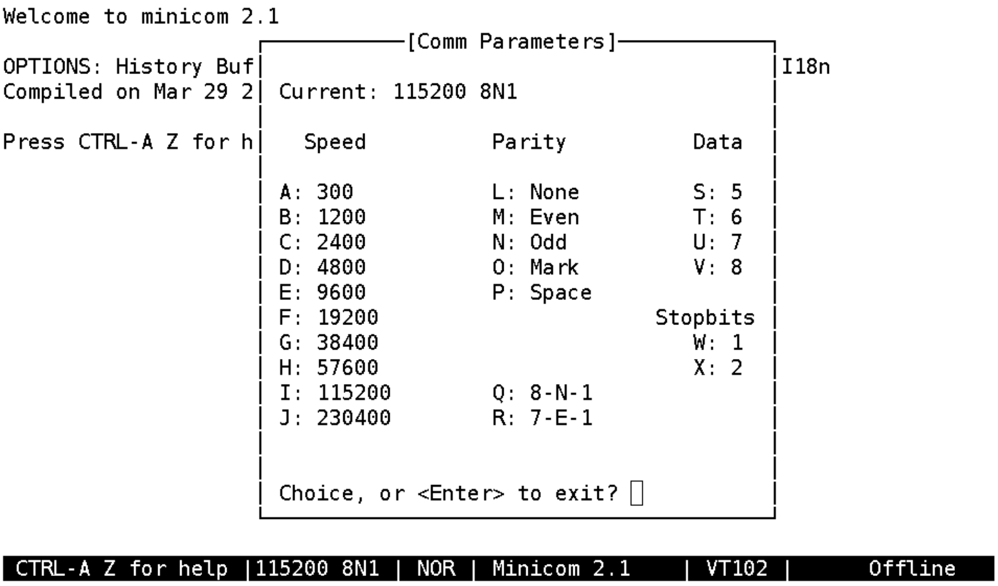
```
Bootswitch configuration 20
oki_eb67dip_bus_init: done
oki_ml67x_init: scanning for OKI_ML67X device
oki_ml67x_scan: bus=00fc7e14, dev=00fc7de4
oki_ml67x: installed routines
EEPROM: 24cXX, 1024 bytes, single byte addressed
(nvram0) on (24cxx)
NVRAM: crc does not match (74 vs ff)
oki_ml67x_timer: registering timers
oki_ml67x_timer: add with irq 1
oki_ml67x_timer_set: tmr=00f16d90, scaler=1, period=0x8fff
selected all-wr for console write stream
selected all-rd for console read stream
DRAM: 32 Mb (33554432 bytes)
Ricoh R2051K, 36:21:39, 12hr mode
ABLE: 2.07 (oki-eb67dip,oki-m67x) (vince@gerald) Wed Feb 23 15:40:22 GMT 2005
(rom0) on (nor0)
SYSTEM: Warning: cannot get unique default MAC
DM9000: dm0: r1, 00:02:04:06:08:0a int phy, link down
failed to find cpu device
boot.cmd unset, defaulting to '(rom1) root=/dev/mtdblock3 ro console=ttyS00,1920
0'
Autoboot in 12 seconds (attempt 1). Press any key to abort
>
```

Once the connection is established the output from ABLE should be seen in the hyperterm window as in Figure 3.2, “Hyperterm displaying ABLE output”

3.2.2. Using minicom as a serial console

To access the serial console from LINUX® the minicom program can be used. Identify which serial port the EB675001DIP is connected to and ensure a note is made of the correct device node, e.g. something like /dev/ttyS0 or /dev/ttyUSB0.

Figure 3.3. Minicom settings window



```
Welcome to minicom 2.1
                    ┌───────────[Comm Parameters]───────────┐
OPTIONS: History Buf│                                       │I18n
Compiled on Mar 29 2│ Current: 115200 8N1                   │
                    │                                       │
Press CTRL-A Z for h│   Speed          Parity          Data │
                    │                                       │
                    │ A: 300           L: None         S: 5 │
                    │ B: 1200          M: Even         T: 6 │
                    │ C: 2400          N: Odd          U: 7 │
                    │ D: 4800          O: Mark         V: 8 │
                    │ E: 9600          P: Space             │
                    │ F: 19200                     Stopbits │
                    │ G: 38400                        W: 1  │
                    │ H: 57600                        X: 2  │
                    │ I: 115200        Q: 8-N-1             │
                    │ J: 230400        R: 7-E-1             │
                    │                                       │
                    │                                       │
                    │ Choice, or <Enter> to exit? ▯         │
                    └───────────────────────────────────────┘


 CTRL-A Z for help |115200 8N1 | NOR | Minicom 2.1    | VT102 |      Offline
```

Start minicom and ensure the correct settings are selected (Default is Ctrl-A p). These settings are 115200 baud, 8 data bits, no parity and 1 stop bit as shown in Figure 3.3, “Minicom settings window”. Obviously Minicom should be using the correct serial port as noted earlier.

Chapter 4. Design Guide

4.1. Overview

The EB675001DIP is designed to be used as component within a larger system. This chapter describes some of the design considerations which might be useful when embedding the module.

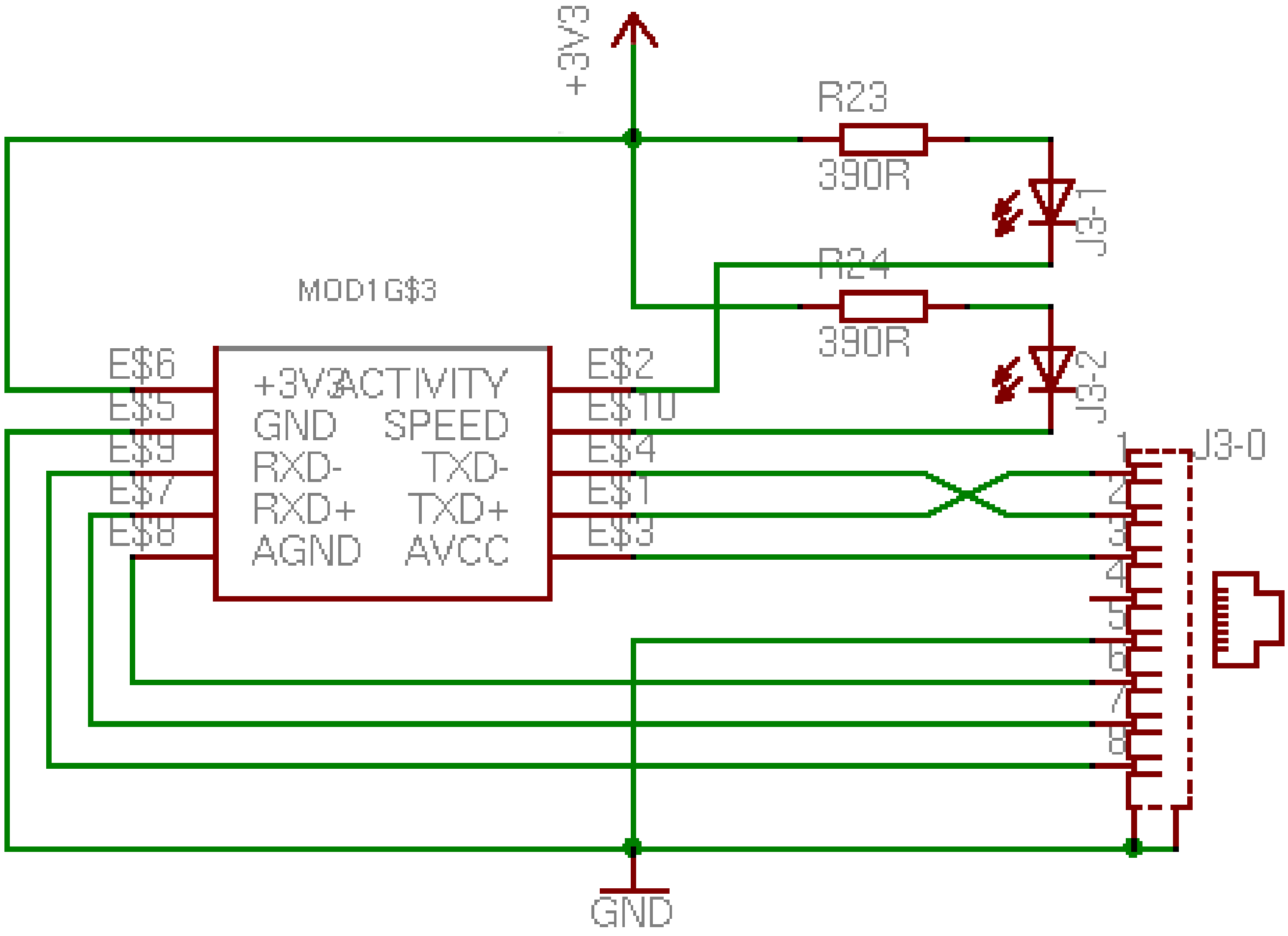
4.2. Connections

The EB675001DIP has four expansion connectors a designer might be interested in (the pinouts are detailed in Section 2.10, “Expansion connectors”).

The two main sixty way expansion headers (PL3 and PL4) are on a standard 0.1inch (2.54mm) grid the Appendix B, *Mechanical drawing* has details of the exact dimensions for all connectors. These connectors carry the signals from the CPU and User CPLD these signals are fully buffered and are 5V tolerant. The buffered RS232 serial signals are replicated on PL4 pins A26 to B29 if the 9way D connector isn't used. The SERIAL_EN signal on B30 allows the serial buffer to be disabled which allows the unbuffered serial signals to be used on B16 to B25.

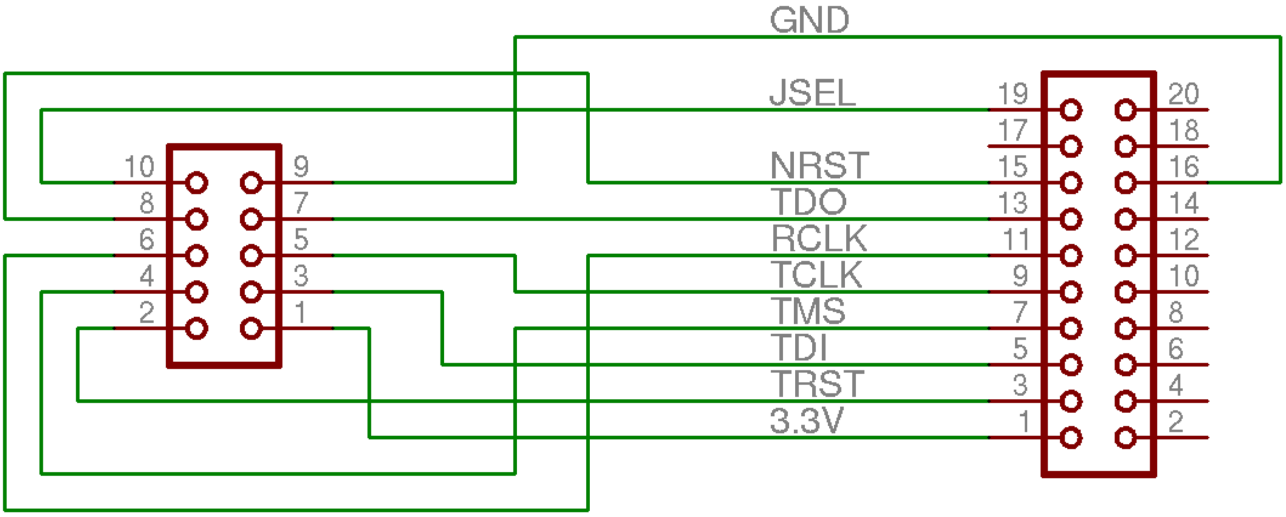
The Ethernet header (PL7) is again a standard 0.1inch 2x5 header which is only available if the module is purchased without the Ethernet jack fitted. A suitable RJ45 jack with integral magnetics and LEDs is the Bothhand L5041 which may be obtained along with the module if desired. The Ethernet expansion connector allows the connector to be placed in a more convenient position within a larger design.

Figure 4.1. Schematic fragment using EB675001DIP Ethernet header



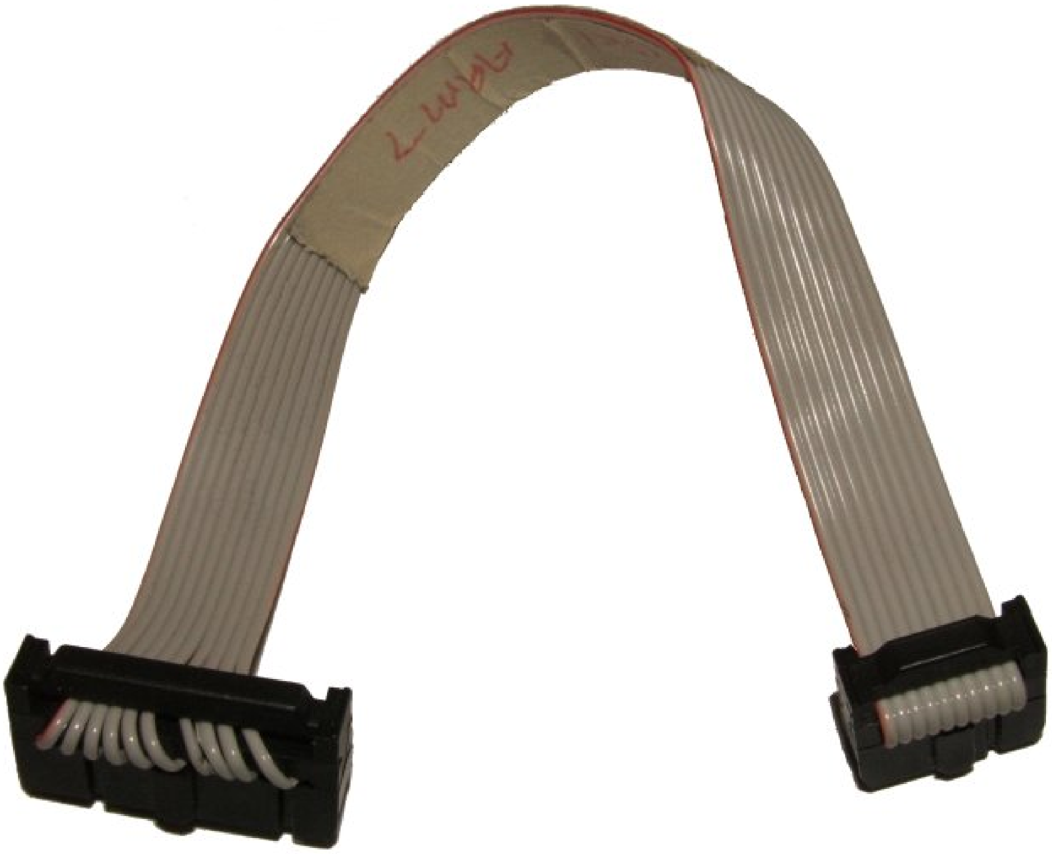
The JTAG header PL6 gives access to the on board JTAG chain comprising the OKI CPU, system CPLD and the user CPLD. The layout is a 2x5 0.1inch (2.54mm) pitch header which can be connected to a standard multi-ICE 2x10 connector with a simple IDC cable.

Figure 4.2. EB675001DIP header to multi-ICE JTAG cable schematic



This schematic of the cable while accurate does not make the simplicity of the cable immediately obvious. An image of a completed IDC cable gives a better representation.

Figure 4.3. EB675001DIP header to multi-ICE JTAG cable

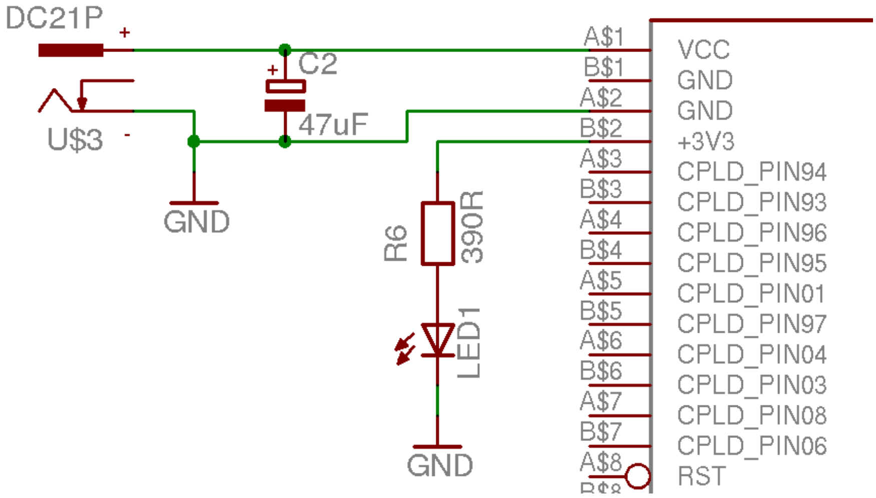


4.3. Power Supply

The first and perhaps most obvious requirement is the power supply to the module. The designer has two choices either to use the modules on board linear power regulator or to supply a suitably regulated 3.3V supply.

If the designer chooses to use the on board regulation a simple supply providing 5-12V DC is required, although there is adequate smoothing on board an additional 47uF electrolytic type capacitor will ensure any transient power sags will be dealt with. If using this supply method the modules 3.3V rails can supply 50mA or so of additional current to external devices such as buffers or small logic devices. Good decoupling and grounding must be used if the module supplies external devices.

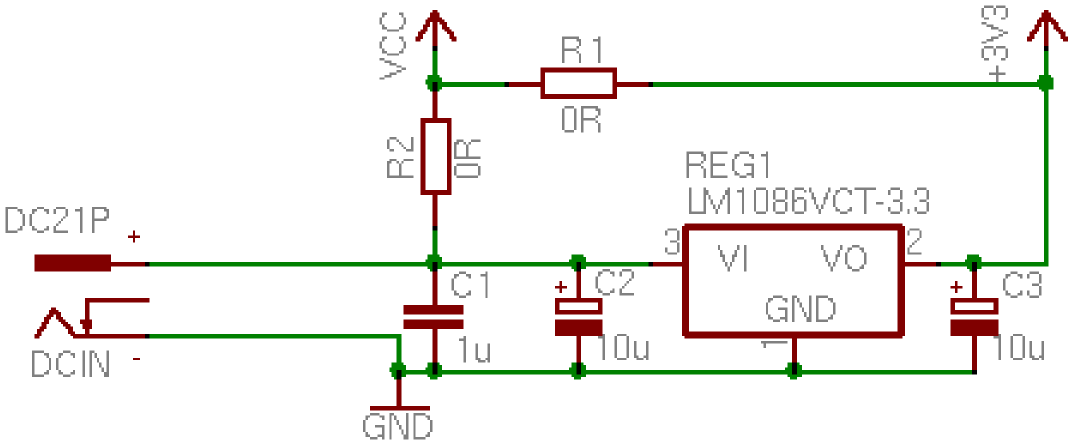
Figure 4.4. Schematic fragment using on board EB675001DIP power regulation



On board regulation is not efficient the exact power usage is outlined in Figure 2.9, “Graph of EB675001DIP power usage”

If the on board regulation will not meet the designers power requirement a 3.3V off board regulator can be used to supply power to both the module and the other application hardware. Because the user CPLD can alter the required power, provision for a maximum current of 0.5A (1.6W) should be made for the module. The circuit outlined in Figure 4.5, “Schematic fragment of flexible EB675001DIP power supply” provides for both on board or off board regulation. Only *one* of the zero ohm resistors R1 and R2 should be fitted. R1 for off board regulation and R2 (with REG1 omitted) if on board regulation is to be used. R1 should be used in off board mode to ensure the internal regulator is not placed in parallel with the external one.

Figure 4.5. Schematic fragment of flexible EB675001DIP power supply



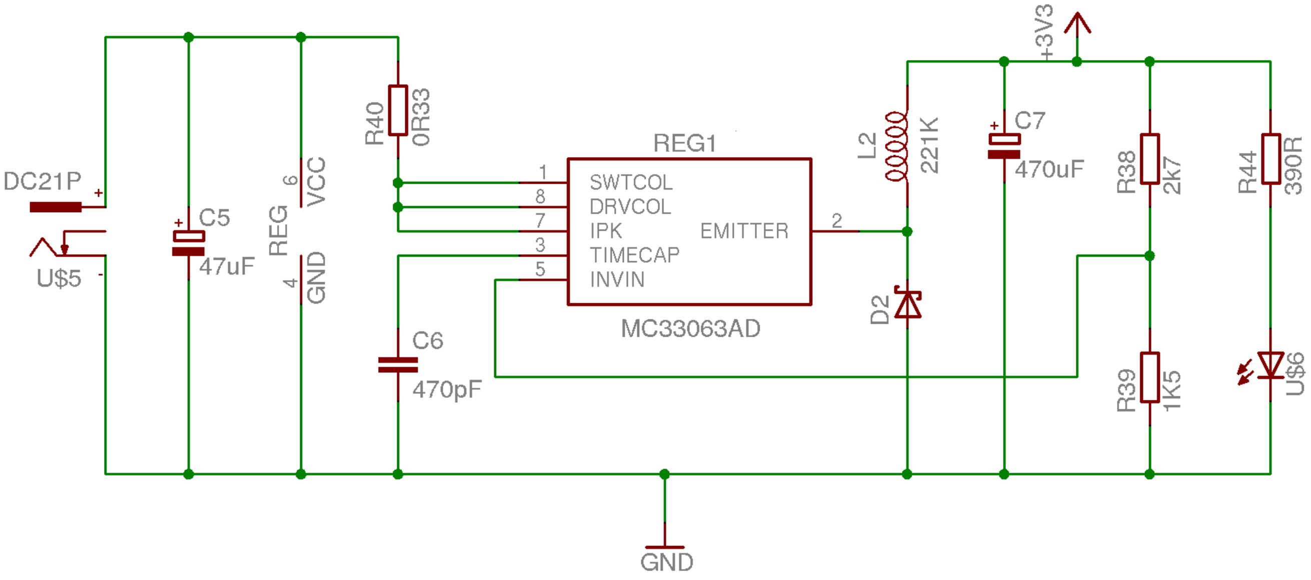
The linear regulator methods (both on board and external) work well for low input voltages and can produce adequate power for small circuits without requiring extensive heatsinks and cooling. For higher power or input voltage requirements a DC to DC converter can be used to generate 3.3V with much greater efficiencies.

One possible circuit is outlined here, this is a simple step down switcher arrangement capable of producing 3W from a wide input voltage. The example circuit shown here can be built from simple through hole components and is generally insensitive to layout issues providing the leads are kept short.

The Texas Instruments MC33063 is a simple eight pin device which can be configured for several modes of operation in addition to the step down configuration shown here. The inductor L2 is a 220uH coil a suitable part might be a TOKO 822LY-221K (this part would limit the output current).

It is of course possible to create this circuit using surface mount components, MC33063AD and A814AY-221K would be the major parts, although circuits and devices with superior performance may be selected in SMT designs.

Figure 4.6. Schematic fragment of DC-DC converter EB675001DIP power supply



There exist a large number of component and circuit choices for DC to DC converters and the user should select the appropriate one for their needs.

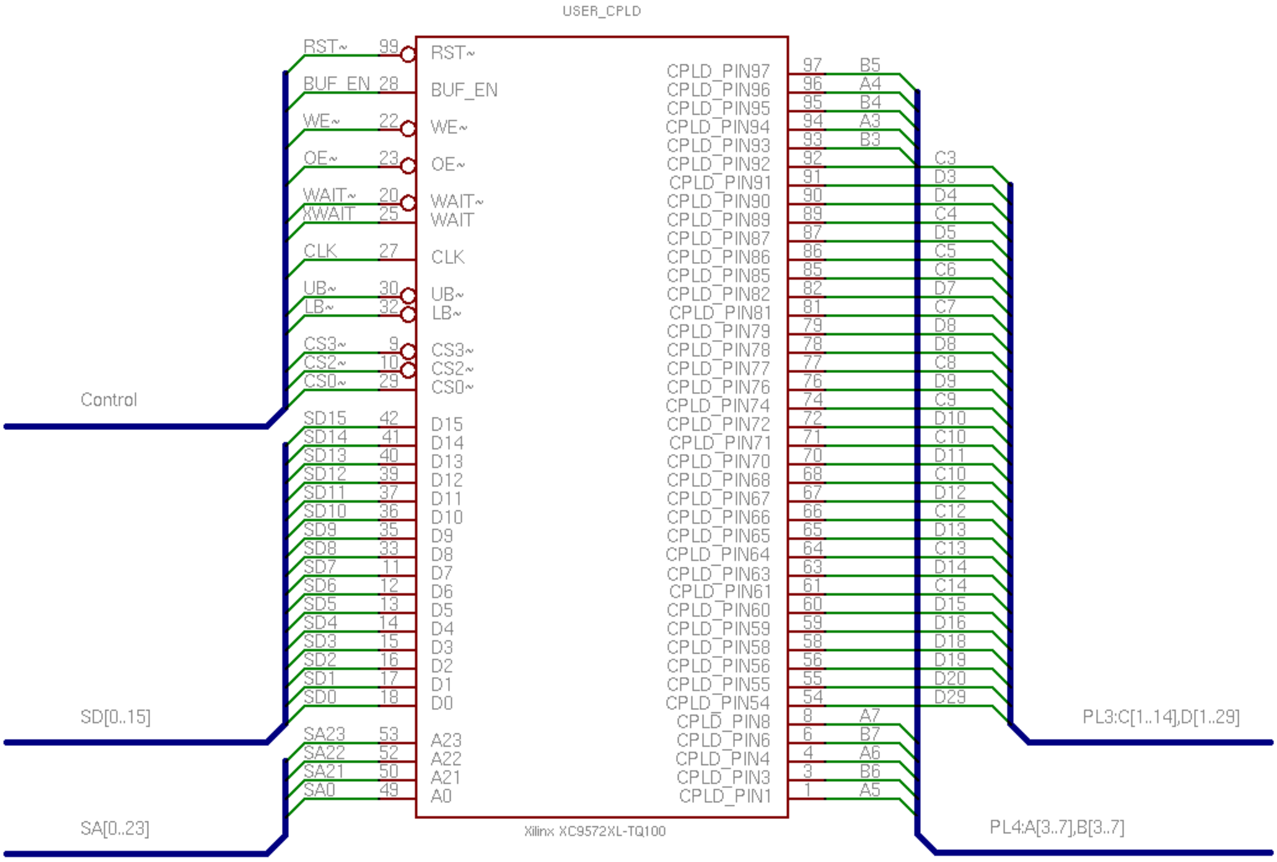
4.4. User CPLD

The Xilinx XC9572XL user CPLD means the EB675001DIP often requires very little external logic to implement a large variety of tasks.

The EB675001DIP has the ability to program the user CPLD with an XSVF file from the ABLE command prompt. The Xilinx tools can be used to generate an XSVF programming file of the user CPLD code. The PlayXSVF utility is executed from the ABLE command line which reconfigures the on board JTAG chain and programs the user CPLD without the use of an external JTAG cable. This is documented in the PlayXSVF User Guide [http://www.simtec.co.uk/products/EB675001DIP/files/playxsvf-book/].

The user CPLD is connected to the OKI processors external data, address and control lines as shown in Figure 4.7, “Pseudo schematic fragment of EB675001DIP user CPLD”. There are 40 uncommitted CPLD pins available on the expansion headers. The schematic fragment does show all the lines and the CPLD pins they are connected to however the EB675001DIP resources User CPLD section [http://www.simtec.co.uk/products/EB675001DIP/resources.html#usercpld] contains some examples and template projects for use with Xilinx webpack software. These examples *include* a suitable constraints and pin naming file so the signals can be referred to with meaningful symbolic names.

Figure 4.7. Pseudo schematic fragment of EB675001DIP user CPLD

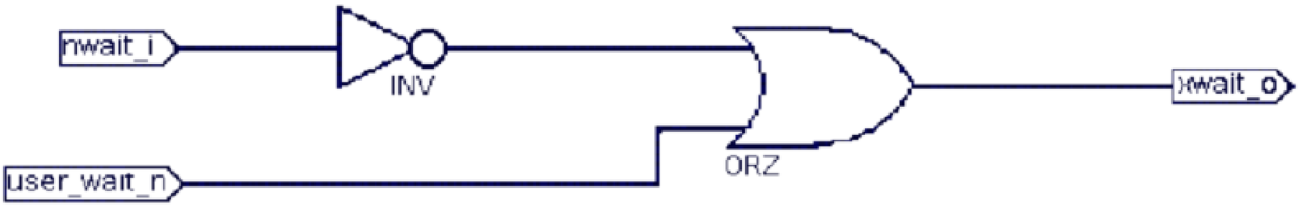


The user CPLD is coupled into the CPU IO XWAIT signal (on CPLD pin 25) which allows I/O cycles to be extended by slow peripherals. The XWAIT is only used by the CPU within the first external chip select region (CS0). The inverted I/O wait request from the Ethernet controller is also brought to the CPLD (pin 20). The designer *must* ensure the relevant logic is used to combine the nWAIT input and any internal I/O wait requirement to generate a correct XWAIT output. The supplied templates include the VHDL

```
xwait_o <= '1'
        when nwait_i='0' or user_wait_n='0' else '0'; user_wait_n
        <= '1';
```

or the schematic fragment

Figure 4.8. Webpack XWAIT schematic fragment



any user design must include this or something very similar or the module will appear to stop on any IO access.

The CPLD is intended to be accessed as a memory mapped device using the OKI processors external memory controller. The ML675001 has four external I/O regions these are decoded into 64MB regions at 0xF0000000, 0xF4000000, 0xF8000000 and 0xFC000000. A read or write by the CPU to these regions causes a full external I/O cycle to be performed qualified by the appropriate chip select lines CS0, CS1, CS2 or CS3. The OKI manual contains full information on how to adjust timing cycles and access type as appropriate.

The user CPLD has access to half of the CS0 area and all of CS2 and CS3. The CS0 signal presented to the user CPLD (sometimes referred to as CPLD_ENABLE) has already been decoded by the system CPLD and is active only for the first

32MB of the region, the other 32MB being decoded to the Ethernet controller. The CS0 area is provided, address space 0xF0000000 to 0xF0DFFFFF, because only in this region can the I/O Wait line be used to delay an IO cycle CPLD.

If the user design needs to generate interrupts an interrupt line one of IRQ1, IRQ2 and IRQ3 (on pins A12 A11 A10) should be connected to a CPLD nearby pin (A3 to A7). The IRQ0 should *not* be used as this is shared with the Ethernet controller and unless specially coded for will prevent correct network operation.

The CLK input is connected to the OKI CPU CKO clock output pin this clock runs at the CPU HCLK frequency which is typically configured to 58.976MHz (7.372MHz baud rate clock with PLL multiplier of 8).

The reset line is the global system reset and should be used to reset any internal state. The signal is active low and is typically held for several ms.

The numerous application notes that accompany the EB675001DIP web resources should give some further ideas to a designer.

Appendix A. Board Layout

Figure A.1. EB675001DIP board layout top side

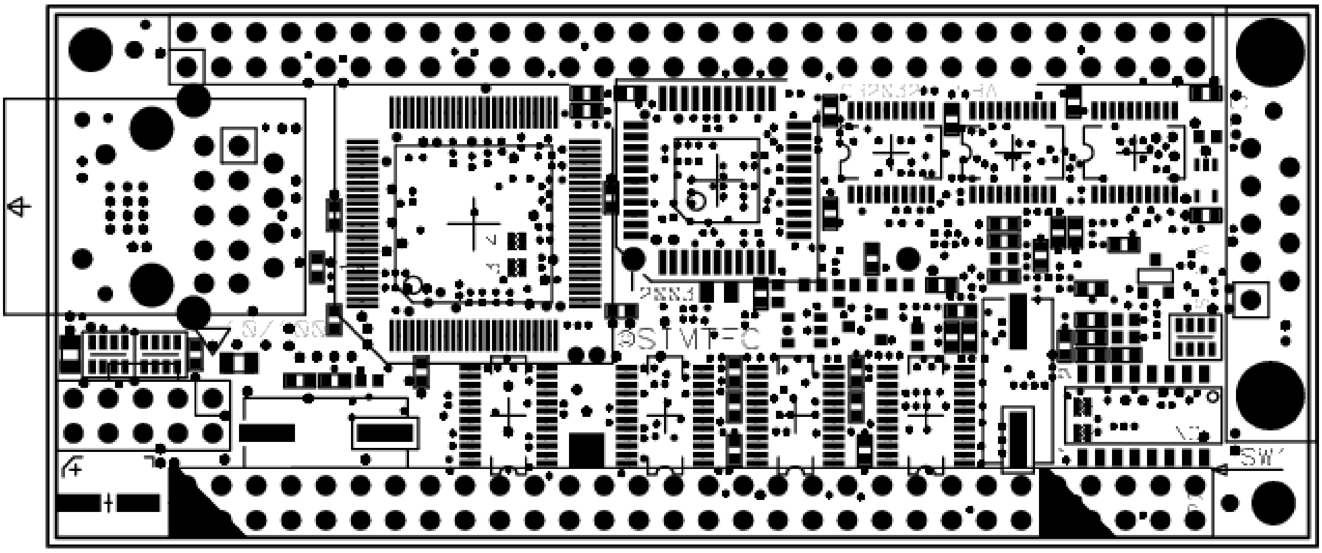
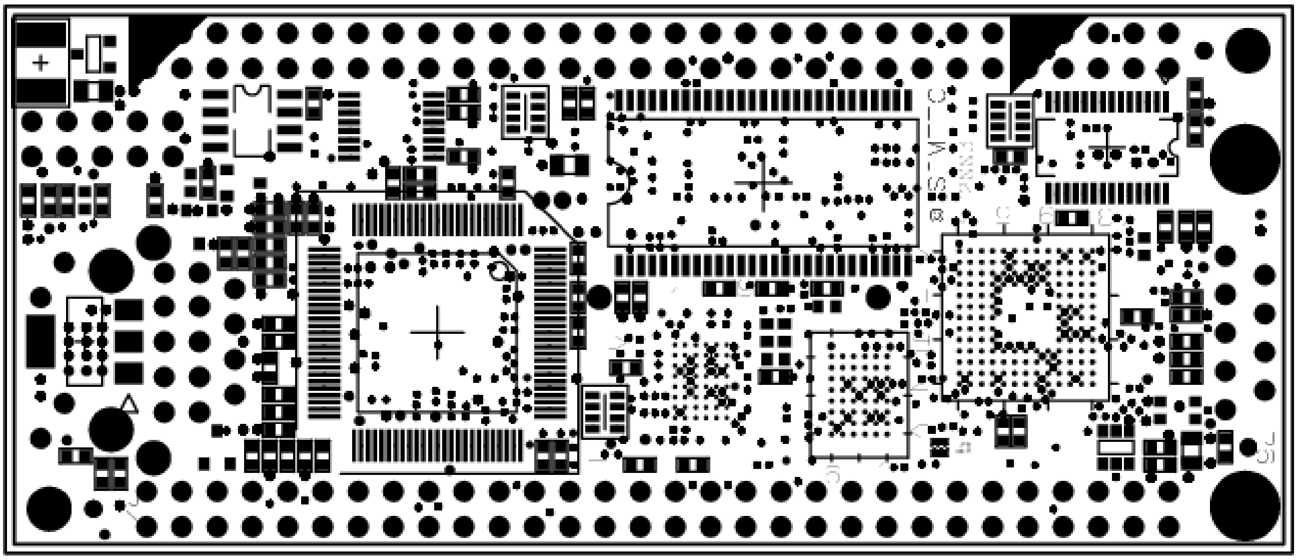
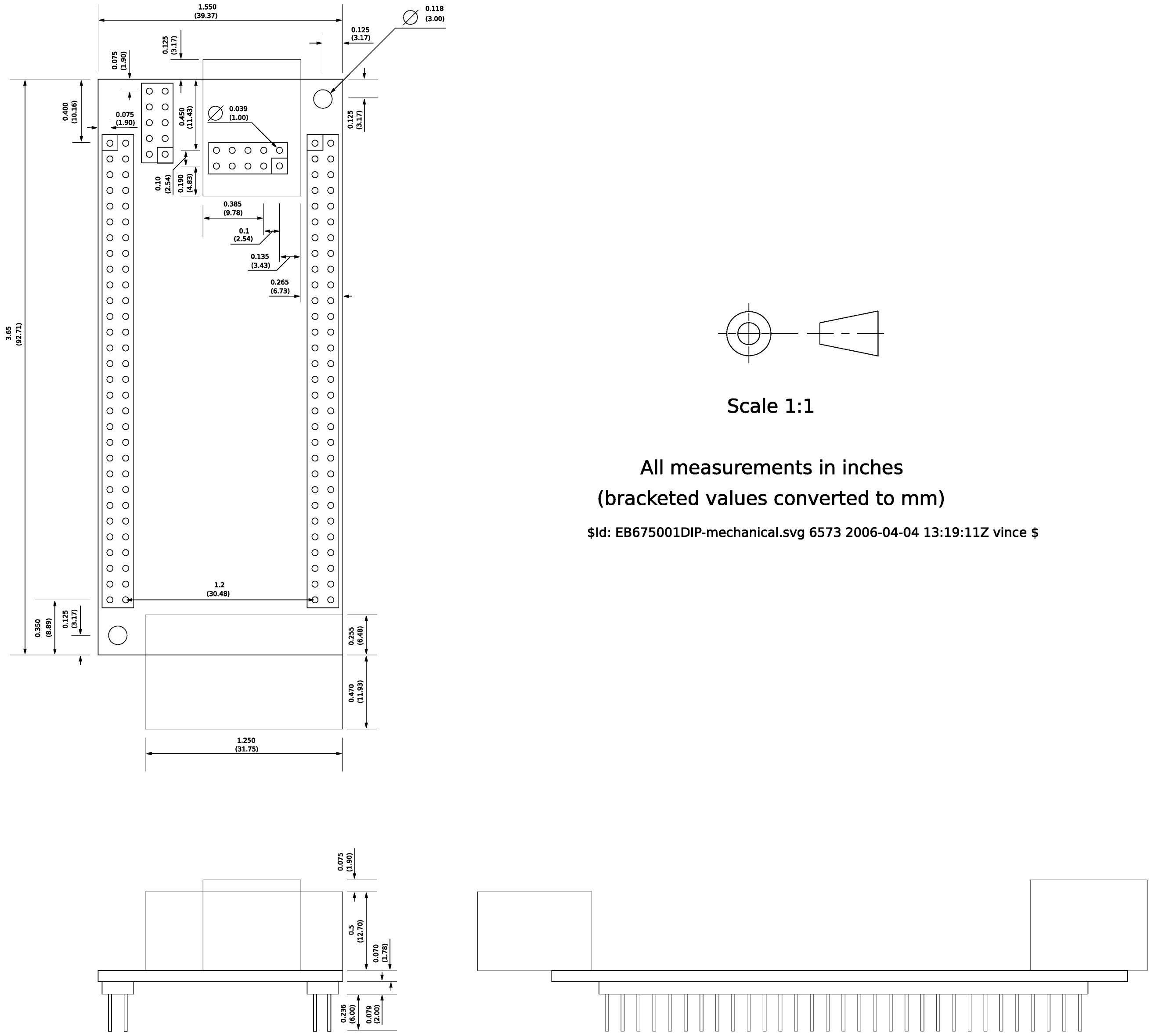


Figure A.2. EB675001DIP board layout bottom side



Appendix B. Mechanical drawing

Figure B.1. EB675001DIP Mechanical Drawing



Scale 1:1

All measurements in inches
(bracketed values converted to mm)

$Id: EB675001DIP-mechanical.svg 6573 2006-04-04 13:19:11Z vince $

Colophon

This Document was prepared in Docbook XML [http://www.docbook.org/] using the GNU emacs text editor. The source was combined with DocBook XSL Stylesheets [http://docbook.sourceforge.net/projects/xsl/] using an XSLT processor to produce output in various formats.

For web output the Saxon XSLT processor [http://saxon.sourceforge.net/] was used to convert the docbook XML directly to HTML.

For print output the Saxon XSLT processor [http://saxon.sourceforge.net/] was used to convert the docbook XML to Formatting Objects (FO) XML.

For general print documents the FO XML is converted to PDF and Postscript with the Apache project FOP [http://xmlgraphics.apache.org/fop/] utility.

For six by nine inch book output the Render X XEP digital typography tool was used to convert the FO XML to print ready PDF output. The URW Nimbus Sans font families were used to perform this typesetting.

The cover designs were developed in the GNU Image Manipulation Program [http://www.gimp.org/] (GIMP).