BSP Targeting the Freescale MPC8555CDS Board

User's Guide

Rev. 1.4

10/2006

How to Reach Us:

Home Page:
www.freescale.com

E-mail:
support@freescale.com

USA/Europe or Locations Not Listed:
Freescale Semiconductor
Technical Information Center, CH370
1300 N. Alma School Road
Chandler, Arizona 85224
+1-800-521-6274 or +1-480-768-2130
support@freescale.com

Europe, Middle East, and Africa:
Freescale Halbleiter Deutschland GmbH
Technical Information Center
Schatzbogen 7
81829 Muenchen, Germany
+44 1296 380 456 (English)
+46 8 52200080 (English)
+49 89 92103 559 (German)
+33 1 69 35 48 48 (French)
support@freescale.com

Japan:
Freescale Semiconductor Japan Ltd.
Headquarters
ARCO Tower 15F
1-8-1, Shimo-Meguro, Meguro-ku,
Tokyo 153-0064, Japan
0120 191014 or +81 3 5437 9125
support.japan@freescale.com

Asia/Pacific:
Freescale Semiconductor Hong Kong Ltd.
Technical Information Center
2 Dai King Street
Tai Po Industrial Estate
Tai Po, N.T., Hong Kong
+800 2666 8080
support.asia@freescale.com

For Literature Requests Only:
Freescale Semiconductor Literature Distribution Center
P.O. Box 5405
Denver, Colorado 80217
1-800-441-2447 or 303-675-2140
Fax: 303-675-2150
LDCForFreescaleSemiconductor@hibbertgroup.com

Information in this document is provided solely to enable system and software implementers to use Freescale Semiconductor products. There are no express or implied copyright licenses granted hereunder to design or fabricate any integrated circuits or integrated circuits based on the information in this document.
Freescale Semiconductor reserves the right to make changes without further notice to any products herein. Freescale Semiconductor makes no warranty, representation or guarantee regarding the suitability of its products for any particular purpose, nor does Freescale Semiconductor assume any liability arising out of the application or use of any product or circuit, and specifically disclaims any and all liability, including without limitation consequential or incidental damages. “Typical” parameters that may be provided in Freescale Semiconductor data sheets and/or specifications can and do vary in different applications and actual performance may vary over time. All operating parameters, including “Typicals”, must be validated for each customer application by customer’s technical experts. Freescale Semiconductor does not convey any license under its patent rights nor the rights of others. Freescale Semiconductor products are not designed, intended, or authorized for use as components in systems intended for surgical implant into the body, or other applications intended to support or sustain life, or for any other application in which the failure of the Freescale Semiconductor product could create a situation where personal injury or death may occur. Should Buyer purchase or use Freescale Semiconductor products for any such unintended or unauthorized application, Buyer shall indemnify and hold Freescale Semiconductor and its officers, employees, subsidiaries, affiliates, and distributors harmless against all claims, costs, damages, and expenses, and reasonable attorney fees arising out of, directly or indirectly, any claim of personal injury or death associated with such unintended or unauthorized use, even if such claim alleges that Freescale Semiconductor was negligent regarding the design or manufacture of the part.

Learn More: For more information about Freescale products, please visit www.freescale.com.

Freescale™ and the Freescale logo are trademarks of Freescale Semiconductor, Inc. All other product or service names are the property of their respective owners.
© Freescale Semiconductor, Inc. 2006. All rights reserved.

Contents

About This Manual vii

Audience vii

Organization...vii

Conventions vii

Definitions, Acronyms, and Abbreviations...vii

Chapter 1 Introduction 1-1

1.1 LTIB Overview 1-1

1.2 BSP Overview 1-1

Chapter 2 LTIB Basics 2-1

2.1 Installing the BSP 2-1

2.2 Running LTIB 2-1

Chapter 3 Target Configuration 3-1

3.1 Supported Target Revisions..3-1

3.2 Target System Memory Map 3-1

3.3 Target Set-up 3-1

Chapter 4 Target Deployment 4-1

4.1 Host Set-up 4-1

4.2 Flashing U-Boot 4-2

4.3 Configuring U-Boot..4-3

4.3.1 Configuring U-Boot for Ramdisk deployment..4-3

4.3.2 Configuring U-Boot for NFS deployment 4-3

4.4 Development Deployment...4-4

4.4.1 Ramdisk deployment...4-4

4.4.2 NFS deployment 4-4

4.5 Production Deployment....4-5

4.5.1 Setting U-Boot environments....4-6

4.5.2 Program kernel to flash....4-6

4.5.3 Program ramdisk to Flash....4-6

4.5.4 Booting up board....4-7

About This Manual

This User Manual provides information on the basic features supported by the BSP and provides you with instructions about how to accomplish these tasks:

- Install the BSP on a host development system.
- Run Linux Target Image Builder (LTIB) to build target images.
- Deploy built images to the MPC8555CDS board.
- Boot Linux on the MPC8555CDS board.

Audience

This document is addressed to developers who want to take advantage of the Freescale Linux Target Image Builder (LTIB) for the MPC8555CDS Board Support Package (BSP).

Organization

This document is organized into 4 chapters.

Chapter 1	Provides an introduction to the MPC8555CDS BSP.
Chapter 2	Provides basic information on LTIB.
Chapter 3	Provides important target set-up information.
Chapter 4	Provides host and target-specific build and deployment information.

Conventions

This document uses the following notational conventions:

- Courier monospaced type indicates commands, command parameters, code examples, expressions, data types, and directives.
- Italic type indicates replaceable command parameters.
- All source code examples are in C.

Definitions, Acronyms, and Abbreviations

The following list defines the abbreviations used in this document.

AppTRK	Application Target Resident Kernel
BSP	Board Support Package
CCS	Command Control Server
CDC	CPU Daughter Card
CDS	Configurable Development System
CW	CodeWarrior
DDR	Dual Data Rate

LTIB	Linux Target Image Builder
MAC	Media Access Control
NFS	Network File System
PCI	Peripheral Component Interconnect
PPC	PowerPC
SEC	Security Engine
TFTP	Trivial File Transfer Protocol
TSEC	Three-Speed Ethernet Controller
UART	Universal Asynchronous Receiver Transmitter
USB	Universal Serial Bus

Chapter 1
Introduction

1.1 LTIB Overview

The Linux Target Image Builder (LTIB) is a tools framework used to manage, configure, extend and build Linux software elements to easily build a Linux target image and a root filesystem. LTIB runs on an x86 PC running the Linux OS.

This BSP operates with LTIB running on a host development system with the following:

- Ethernet card
- Serial port
- 1 GB of free disk space required
- NFS Server
- TFTP Server
- rsync
- perl

NOTE: Be aware that some host side packages may not function properly on every Linux distribution. The following are platforms where LTIB was tested.

- Redhat: 7.3, 8.0, 9.0
- Fedora Core: 1, 2, 3
- Debian: 3.1r0 (stable), unstable
- SuSE: 8.2, 9.2, 10.0

1.2 BSP Overview

This MPC8555CDS BSP is designed for use with LTIB. Once the BSP is installed and running with its basic configuration, you can use LTIB to customize your project.

The BSP components provide the tools, device drivers, and additional features needed for your embedded Linux project.

Linux 2.6.11 kernel

- Targeting the Freescale MPC8555CDS board (CDC Ver. 1.1, Carrier Card Ver1.3 and Arcadia X3.1)
- Support PQ3Lite MPC8555 processor
- U-boot 1.1.3
- Linux 2.6.11 Kernel supporting the e500 core
- DUART driver
- DDR memory supported

- TSEC1 and TSEC2 drivers
- I2C driver
- 32bit PCI driver for PCI1 (33MHz/66MHz) (Support RealTek 8139B(it belongs to secondary PCI), RealTek 8139D, RealTek8169S, Intel Pro1000 82540, Intel Pro1000 82544 3COM 3C996 and 3COM 3C905 cards) and PCI2 (33MHz/66MHz) (Support RealTek 8139D, Intel Pro1000 82540 and 3COM 3C905 cards)
- CPM enabled with interrupt supported
- USB host mode driver
- Security Engine (SEC) 2X driver and testing program
- TSI310 supported on Arcadia X3.1. PCI secondary bus driver workable. Supporting the on board VIA VT82C686B IDE controller and RTL8139 Ethernet controller
- Silicon 680 IDE controller driver and IDE hard disk utilities (ext3 and FAT partition)
- Local Bus SDRAM support
- Codewarrior U-Boot debug and kernel debug supported and initialization files included
- CodeTEST SWIC mode and HWIC PCI probe mode supported
- TCP/IP stack
- Ftp client and server
- Telnet client and server
- Web server (boa)
- Integrated with LTIB
- Both NFS and Ramdisk filesystem deployment supported
- Integration of AppTRK for Common PPC

Bootloader version

- U-Boot 1.1.3

Toolchain version

- Gcc3.4.3–e500, Glibc2.3.3-spe, Binutils 2.15, supporting the e500 core

Documentation. See START_HERE.html on this CD.

Chapter 2
LTIB Basics

2.1 Installing the BSP

Please follow the steps below to install LTIB on your host machine.

1. As root, mount the ISO image on your machine:

   ```
   mount –o loop MPC8555CDS_20061017-ltib.iso /mnt/cdrom
   ```

2. As a non-root user, install the LTIB:

   ```
   /mnt/cdrom/install
   ```

   You will be prompted to input the desired LTIB install path. Be sure the user has the correct permissions for the install path.

There are no uninstall scripts. To uninstall LTIB you need to remove the `/opt/freescale/pkgs`, `/opt/freescale/ltib` and `<`*`install_path`*`>/ltib` directories manually.

2.2 Running LTIB

To run LTIB, change to the directory into which you installed it and run `./ltib`.

```
cd <install_path>/ltib

./ltib
```

The first time LTIB runs on your machine a number of host packages are built and installed that support LTIB. This may take a few minutes.

Important Note: Please be sure to set the “Target System Configuration” options for your network environment the first time you build.

The LTIB can only run by a non-root user.

To modify the project configuration simply run:

`./ltib –-configure` (or `-c`; type `--help` to see configuration options)

This will re-prompt you for the platform/board configuration. In the board configuration screens, change settings and select packages as appropriate. When you exit the configuration screen your target image will be adjusted accordingly.

Once you build your project you will get following directory/image files:

- **rootfs/** – directory, the root file system that will be deployed on your board.
- **rootfs/boot/uImage** – kernel image that can be loaded with U-Boot
- **rootfs.ext2.gz.uboot** – ramdisk image that can be loaded with U-Boot
- **rootfs.ext2.gz** – gzipped ramdisk image
- **rootfs/boot/u-boot.bin** – U-Boot binary image that can be programmed into MPC8555CDS board Flash.

If you want to fully re-configure and re-compile all the packages, you can do the following. This is generally not necessary.

1. Clean up all the configure files and objects thoroughly:

   ```
   ./ltib –m distclean
   ```

2. You will be prompted to confirm your choice. Type yes to perform a distclean.

3. Run litb

   ```
   ./ltib
   ```

More information on LITB can be found in `<`*`install path`*`>/ltib/doc.` Or on the web at http://savannah.nongnu.org/projects/ltib.

Chapter 3
Target Configuration

3.1 Supported Target Revisions

The target system is the MPC8555CDS board. This BSP is known to work on the following board revision:

MPC8555 CDC Ver.1.1
Carrier Card Ver. 1.3
Arcadia X3.1

3.2 Target System Memory Map

After system startup, the boot loader maps system memory as shown below.

Range Start	Range End	Definition	Size
0x0000_0000	0x7fff_ffff	DDR	2G
0x8000_0000	0x9fff_ffff	PCI1 MEM	512M
0xa000_0000	0xbfff_ffff	PCI2 MEM	512M
0xe000_0000	0xe00f_ffff	CCSR	1M
0xe200_0000	0xe2ff_ffff	PCI1 IO	16M
0xe300_0000	0xe3ff_ffff	PCI2 IO	16M
0xf000_0000	0xf7ff_ffff	SDRAM	128M
0xf800_0000	0xf80f_ffff	NVRAM/CADMUS	1M
0xf900_0000	0xf90f_ffff	CodeTEST	1M
0xff00_0000	0xff7f_ffff	FLASH (2nd bank)	8M
0xff80_0000	0xffff_ffff	FLASH (boot bank)	8M

3.3 Target Set-up

1. Connect the MPC8555CDS board to the network via the TSEC 1 port on the board

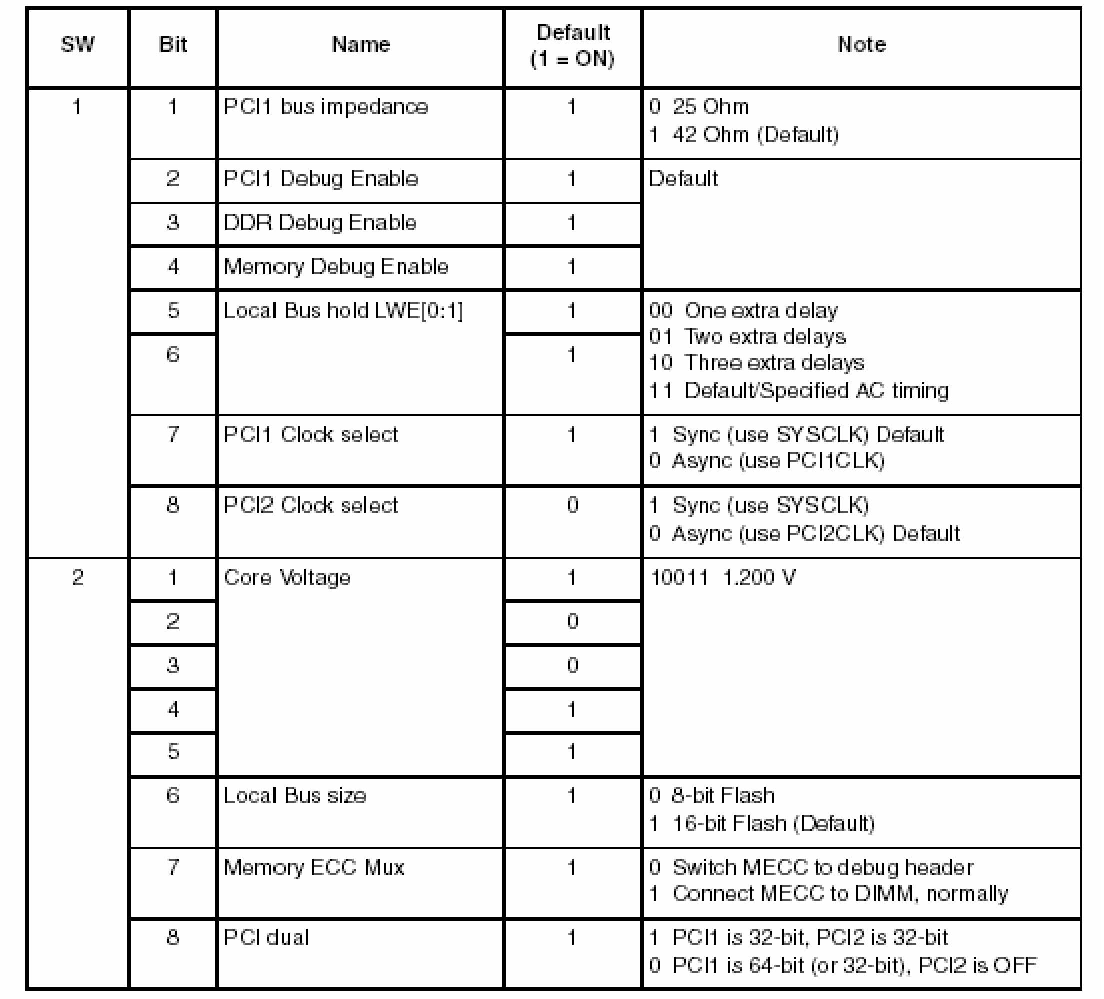
2. Connect the MPC8555CDS board to the host machina via the serial port with an RS-232 cable.

3. Setup the hyperterminal in the host machine with 115200bps, 8-N-1, no flow control.

4. Verify all the switches and jumpers are setup correctly default value:

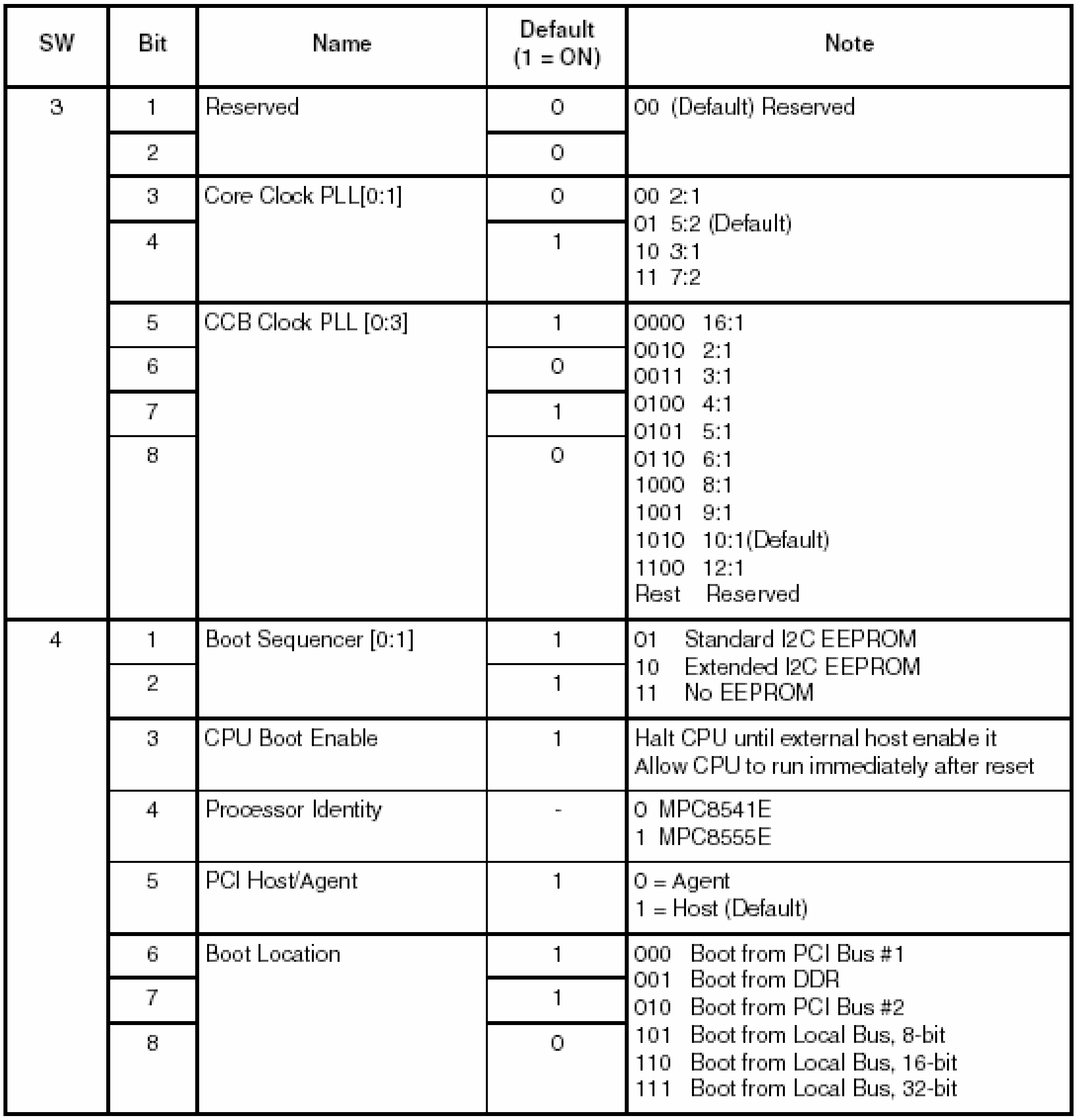
(1). Set the swithces of MPC8555 CDC as Table 3-1.

Table 3-1 Default Setting of MPC8555 Processor Card(Ver1.1)



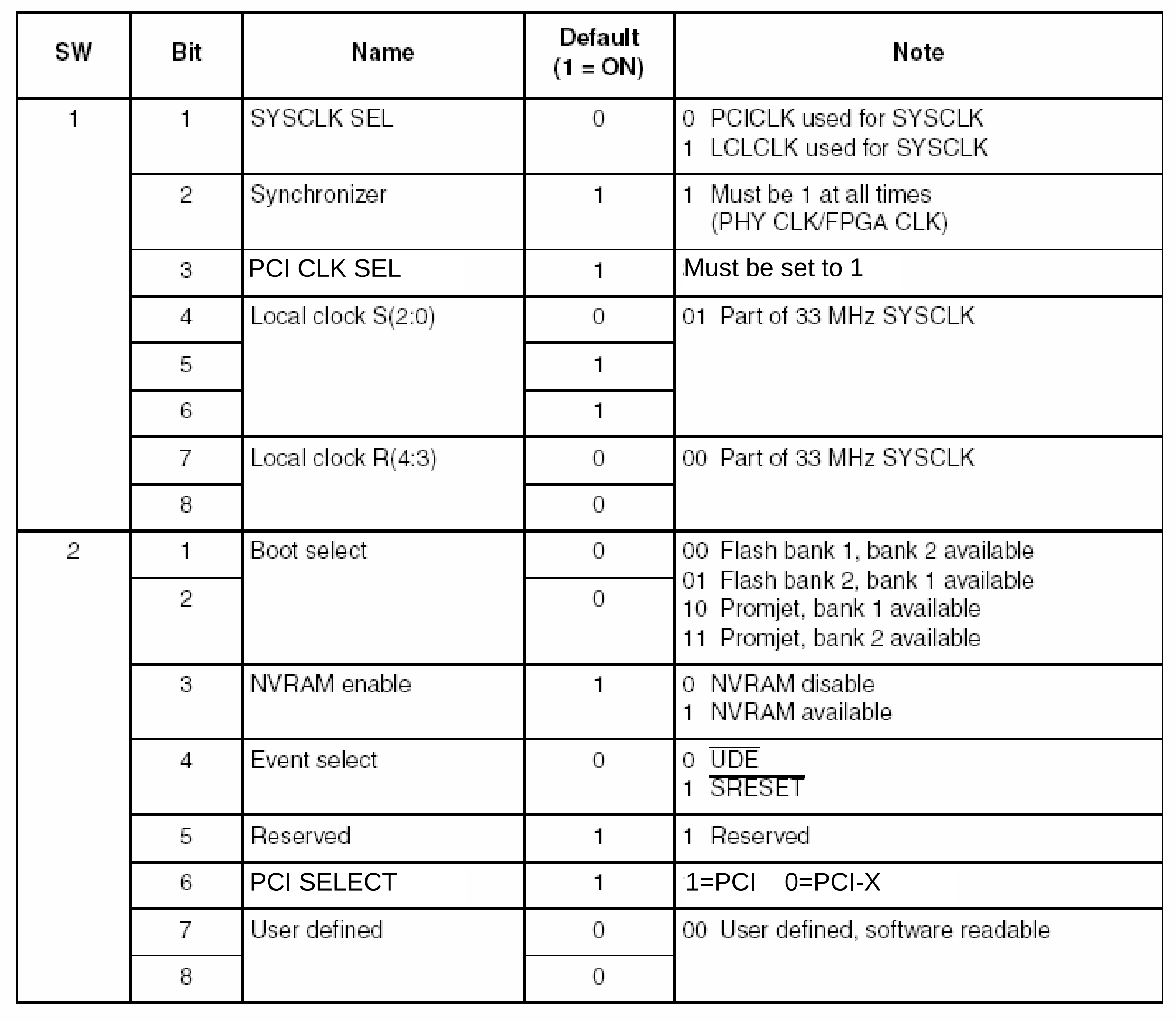
SW	Bit	Name	Default (1 = ON)	Note
1	1	PCI1 bus impedance	1	0 25 Ohm 1 42 Ohm (Default)
	2	PCI1 Debug Enable	1	Default
	3	DDR Debug Enable	1	
	4	Memory Debug Enable	1	
	5	Local Bus hold LWE[0:1]	1	00 One extra delay 01 Two extra delays 10 Three extra delays 11 Default/Specified AC timing
	6		1	
	7	PCI1 Clock select	1	1 Sync (use SYSCLK) Default 0 Async (use PCI1CLK)
	8	PCI2 Clock select	0	1 Sync (use SYSCLK) 0 Async (use PCI2CLK) Default
2	1	Core Voltage	1	10011 1.200 V
	2		0	
	3		0	
	4		1	
	5		1	
	6	Local Bus size	1	0 8-bit Flash 1 16-bit Flash (Default)
	7	Memory ECC Mux	1	0 Switch MECC to debug header 1 Connect MECC to DIMM, normally
	8	PCI dual	1	1 PCI1 is 32-bit, PCI2 is 32-bit 0 PCI1 is 64-bit (or 32-bit), PCI2 is OFF

Table 3-1 Default Setting of MPC8555 Processor Card (Continued)



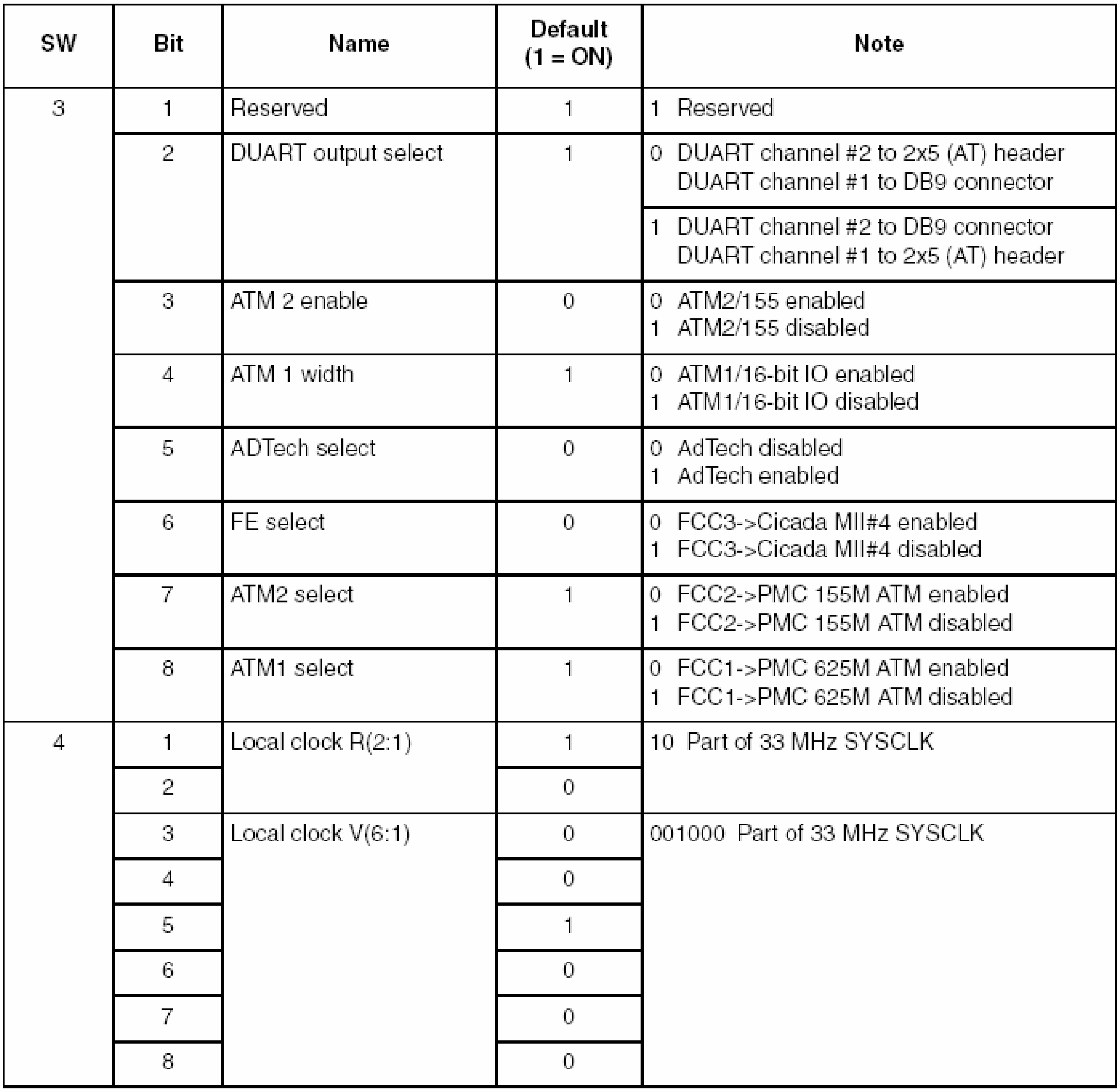
SW	Bit	Name	Default (1 = ON)	Note
3	1	Reserved	0	00 (Default) Reserved
	2		0	
	3	Core Clock PLL[0:1]	0	00 2:1 01 5:2 (Default) 10 3:1 11 7:2
	4		1	
	5	CCB Clock PLL [0:3]	1	0000 16:1 0010 2:1 0011 3:1 0100 4:1 0101 5:1 0110 6:1 1000 8:1 1001 9:1 1010 10:1(Default) 1100 12:1 Rest Reserved
	6		0	
	7		1	
	8		0	
4	1	Boot Sequencer [0:1]	1	01 Standard I2C EEPROM 10 Extended I2C EEPROM 11 No EEPROM
	2		1	
	3	CPU Boot Enable	1	Halt CPU until external host enable it Allow CPU to run immediately after reset
	4	Processor Identity	-	0 MPC8541E 1 MPC8555E
	5	PCI Host/Agent	1	0 = Agent 1 = Host (Default)
	6	Boot Location	1	000 Boot from PCI Bus #1 001 Boot from DDR 010 Boot from PCI Bus #2 101 Boot from Local Bus, 8-bit 110 Boot from Local Bus, 16-bit 111 Boot from Local Bus, 32-bit
	7		1	
	8		0	

Table 3-2 Default Setting of Carrier Card (Ver.1.3)



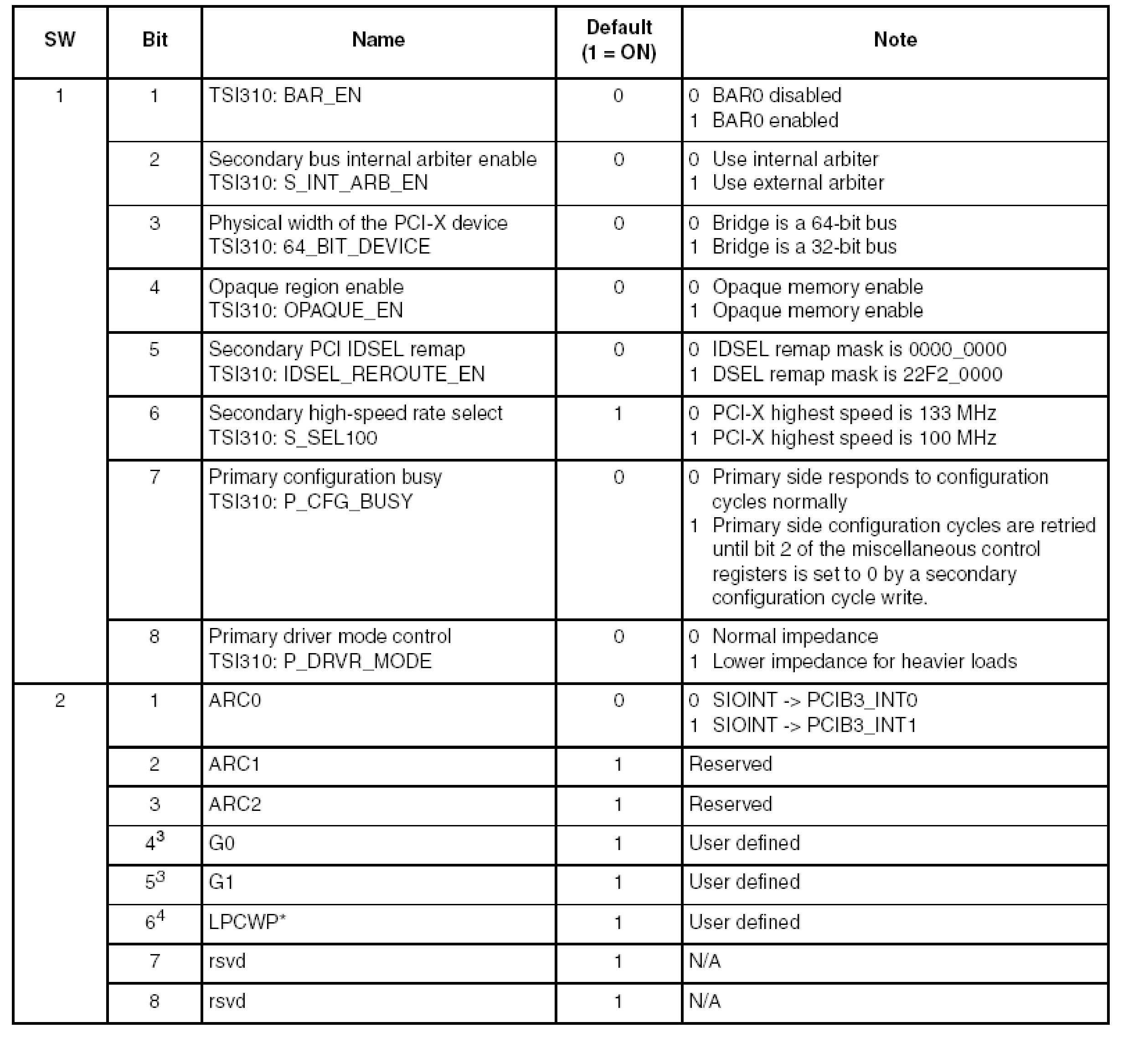
SW	Bit	Name	Default (1 = ON)	Note
1	1	SYSCLK SEL	0	0 PCICLK used for SYSCLK 1 LCLCLK used for SYSCLK
	2	Synchronizer	1	1 Must be 1 at all times (PHY CLK/FPGA CLK)
	3	PCI CLK SEL	1	Must be set to 1
	4	Local clock S(2:0)	0	01 Part of 33 MHz SYSCLK
	5		1	
	6		1	
	7	Local clock R(4:3)	0	00 Part of 33 MHz SYSCLK
	8		0	
2	1	Boot select	0	00 Flash bank 1, bank 2 available 01 Flash bank 2, bank 1 available 10 Promjet, bank 1 available 11 Promjet, bank 2 available
	2		0	
	3	NVRAM enable	1	0 NVRAM disable 1 NVRAM available
	4	Event select	0	0 $\overline{\text{UDE}}$ 1 $\overline{\text{SRESET}}$
	5	Reserved	1	1 Reserved
	6	PCI SELECT	1	1=PCI 0=PCI-X
	7	User defined	0	00 User defined, software readable
	8		0	

Table 3-2 Default Setting of Carrier Card (Continued)



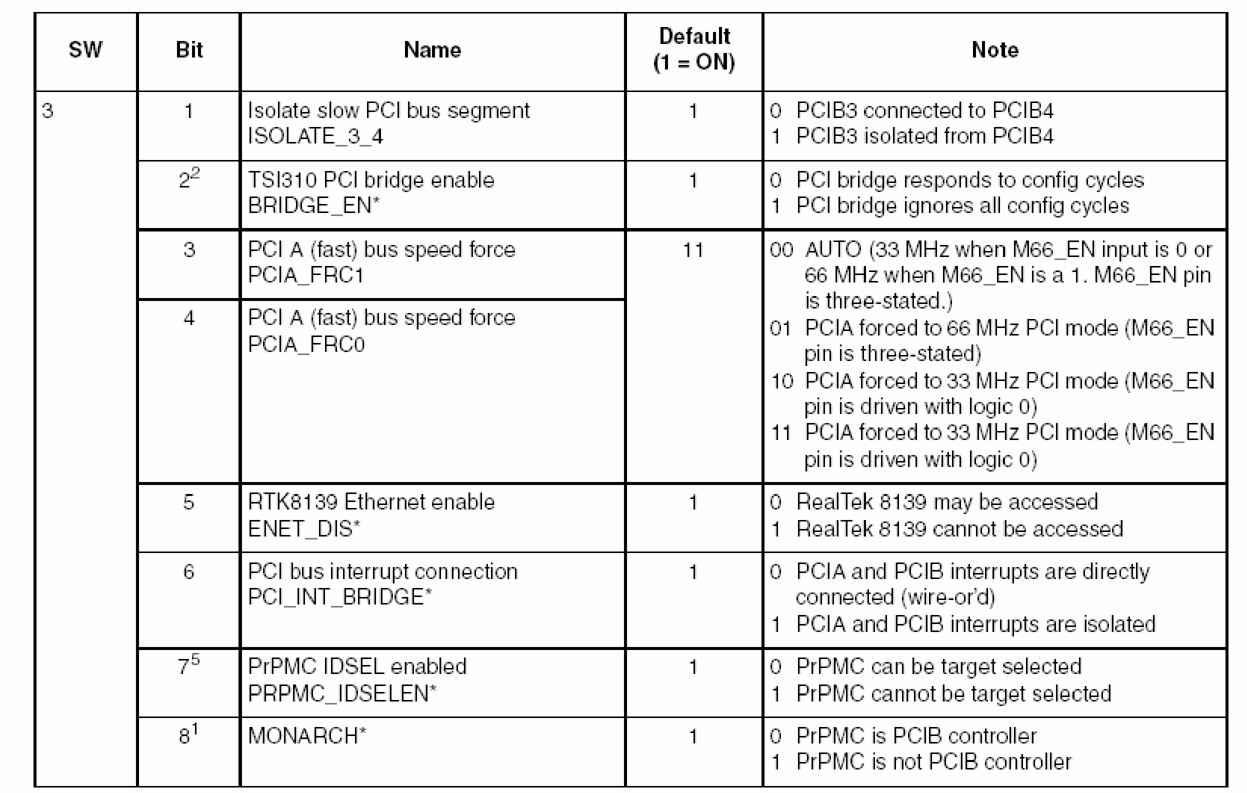
SW	Bit	Name	Default (1 = ON)	Note
3	1	Reserved	1	1 Reserved
	2	DUART output select	1	0 DUART channel #2 to 2x5 (AT) header DUART channel #1 to DB9 connector
				1 DUART channel #2 to DB9 connector DUART channel #1 to 2x5 (AT) header
	3	ATM 2 enable	0	0 ATM2/155 enabled 1 ATM2/155 disabled
	4	ATM 1 width	1	0 ATM1/16-bit IO enabled 1 ATM1/16-bit IO disabled
	5	ADTech select	0	0 AdTech disabled 1 AdTech enabled
	6	FE select	0	0 FCC3->Cicada MII#4 enabled 1 FCC3->Cicada MII#4 disabled
	7	ATM2 select	1	0 FCC2->PMC 155M ATM enabled 1 FCC2->PMC 155M ATM disabled
	8	ATM1 select	1	0 FCC1->PMC 625M ATM enabled 1 FCC1->PMC 625M ATM disabled
4	1	Local clock R(2:1)	1	10 Part of 33 MHz SYSCLK
	2		0	
	3	Local clock V(6:1)	0	001000 Part of 33 MHz SYSCLK
	4		0	
	5		1	
	6		0	
	7		0	
	8		0	

Table 3-3 Default Setting of Arcadia X3.1



SW	Bit	Name	Default (1 = ON)	Note
1	1	TSI310: BAR_EN	0	0 BAR0 disabled 1 BAR0 enabled
	2	Secondary bus internal arbiter enable TSI310: S_INT_ARB_EN	0	0 Use internal arbiter 1 Use external arbiter
	3	Physical width of the PCI-X device TSI310: 64_BIT_DEVICE	0	0 Bridge is a 64-bit bus 1 Bridge is a 32-bit bus
	4	Opaque region enable TSI310: OPAQUE_EN	0	0 Opaque memory enable 1 Opaque memory enable
	5	Secondary PCI IDSEL remap TSI310: IDSEL_REROUTE_EN	0	0 IDSEL remap mask is 0000_0000 1 DSEL remap mask is 22F2_0000
	6	Secondary high-speed rate select TSI310: S_SEL100	1	0 PCI-X highest speed is 133 MHz 1 PCI-X highest speed is 100 MHz
	7	Primary configuration busy TSI310: P_CFG_BUSY	0	0 Primary side responds to configuration cycles normally 1 Primary side configuration cycles are retried until bit 2 of the miscellaneous control registers is set to 0 by a secondary configuration cycle write.
	8	Primary driver mode control TSI310: P_DRVR_MODE	0	0 Normal impedance 1 Lower impedance for heavier loads
2	1	ARC0	0	0 SIOINT -> PCIB3_INT0 1 SIOINT -> PCIB3_INT1
	2	ARC1	1	Reserved
	3	ARC2	1	Reserved
	4[3]	G0	1	User defined
	5[3]	G1	1	User defined
	6[4]	LPCWP*	1	User defined
	7	rsvd	1	N/A
	8	rsvd	1	N/A

Table 3-2 Default Setting of Arcadia X3.1(Continued)



SW	Bit	Name	Default (1 = ON)	Note
3	1	Isolate slow PCI bus segment ISOLATE_3_4	1	0 PCIB3 connected to PCIB4 1 PCIB3 isolated from PCIB4
	2[2]	TSI310 PCI bridge enable BRIDGE_EN*	1	0 PCI bridge responds to config cycles 1 PCI bridge ignores all config cycles
	3	PCI A (fast) bus speed force PCIA_FRC1	11	00 AUTO (33 MHz when M66_EN input is 0 or 66 MHz when M66_EN is a 1. M66_EN pin is three-stated.) 01 PCIA forced to 66 MHz PCI mode (M66_EN pin is three-stated) 10 PCIA forced to 33 MHz PCI mode (M66_EN pin is driven with logic 0) 11 PCIA forced to 33 MHz PCI mode (M66_EN pin is driven with logic 0)
	4	PCI A (fast) bus speed force PCIA_FRC0		
	5	RTK8139 Ethernet enable ENET_DIS*	1	0 RealTek 8139 may be accessed 1 RealTek 8139 cannot be accessed
	6	PCI bus interrupt connection PCI_INT_BRIDGE*	1	0 PCIA and PCIB interrupts are directly connected (wire-or'd) 1 PCIA and PCIB interrupts are isolated
	7[5]	PrPMC IDSEL enabled PRPMC_IDSELEN*	1	0 PrPMC can be target selected 1 PrPMC cannot be target selected
	8[1]	MONARCH*	1	0 PrPMC is PCIB controller 1 PrPMC is not PCIB controller

Note:

1. This switch configures the MPMC card into the system controller, a mode which is required for normal PCI use. Disabling is provided for testing purposes only.
2. This switch allows software that does not wish to deal with PCI bridges, ignore them, at the cost of access to the other PCI domain.
3. Software-defined switches.
4. Optional feature.
5. Some PCI devices do not allow their own IDSEL to be asserted when operating as the PCI host; if so, use this switch to disable IDSEL. Not applicable for PCI agents.

Chapter 4
Target Deployment

This charpter describes how to deploy the U-Boot, Linux kernel and File system to the Target board. Three deployment methods- Ramdisk deployment, NFS deployment and Flash deployment are introduced.

4.1 Host Set-up

Host setup is critical for your BSP to function. The host must be running tftp and nfs in order for deployment to work. The following instructions are generic. Your system may be different and the commands should be adjusted accordingly.

1. Turn off firewall for tftp to work. `iptables -F` or type "`setup`" at the command line.
2. Install tftp-server.
3. Install nfs-server.
4. Create the tftboot directory.

   ```
   mkdir /tftpboot
   ```

5. Link `rootfs` to an exportable directory once you have built your project.

   ```
   ln –s <install_path>/ltib/rootfs /tftpboot/ltib
   ```

6. Copy over kernel, bootloader, and flash filesystem images for your deployment to the `/tftpboot` directory:

   ```
   cp <install_path>/ltib/rootfs/boot/* /tftpboot
   cp <install_path>/ltib/<flashfs> /tftpboot
   cp /mnt/cdrom/bootloaders/* /tftpboot
   ```

7. Edit `/etc/exports` and add the following line:

   ```
   /tftpboot/ltib/ <target board IP>(rw,no_root_squash, async)
   ```

8. Edit `/etc/xinetd.d/tftp` to enable tftp like this:

   ```
   {
   disable     = no
   socket_type = dgram
   protocol    = udp
   wait        = yes
   user        = root
   server      = /usr/sbin/in.tftpd
   server_args = /tftpboot
   ```

```
}
```

9. Restart the nfs and tftp servers on your host:

   ```
   /etc/init.d/xinetd restart
   /etc/init.d/nfsserver restart
   ```

10. Connect board to the network.
11. Connect the target to the host via a serial connection.
12. Start `minicom` and set it up to talk to the MPC8555CDS board:
    - Serial Setup: Select correct serial device; Hardware & Software Flow control = No; Bps = 115,200
    - Modem & dialing: Delete text for the following: Init String, Reset String, Hang-up String, No flow control
13. Power on board and see the console prompt.

4.2 Flashing U-Boot

The flash range is `0xFF000000` to `0xFFFFFFFF` , and is composed by 2 independent flash chips. It is possible to use either bank to boot using U-Boot. Switch 2 bit 2 on Carrier board is used for this purpose.

`0xFF000000 to 0xFF7FFFFF` – Bank 1, 8MB

`0xFF800000 to 0xFFFFFFFF` – Bank 2, 8MB

The starting address of current flash bank is always 0xFF800000, and that of alternative flash bank is always 0xFF000000

The U-Boot image should be programmed into Flash, starting from `0xFFF80000`, by CodeWarrior for PPC version 8.7 + PowerTAP Pro or USB TAP.

An alternative way is to use the U-Boot commands for programming with the precondition that there is already a workable U-Boot programmed in the Flash:

To Flash U-Boot into current booting bank (`0xFF800000 - 0xFFFFFFFF`):

```
=>tftp 1000000 u-boot.bin
=>protect off all
=>erase fff80000 ffffffff
=>cp.b 1000000 fff80000 80000
Then reset the board to boot it up.
```

To Flash U-Boot into the alternative bank (`0xFF000000 - 0xFF7FFFFF`):

```
=>tftp 1000000 u-boot.bin
=>erase ff780000 ff7fffff
=>cp.b 1000000 ff780000 80000
Then set the SW2[2] to opposite postion before boot up the board.
```

4.3 Configuring U-Boot

4.3.1 Configuring U-Boot for Ramdisk deployment

The images generated by LTIB allow you perform ramdisk deployment. Before performing ramdisk deployment, it is needed to configure U-Boot parameters.

At u-boot prompt, set u-boot environment like:

```
=>setenv ipaddr <board_ipaddress>

=>setenv serverip <tftp_serverip>

=>setenv gatewayip <your_gatewayip>

=>setenv bootargs root=/dev/ram rw console=ttyS1,115200

=>saveenv
```

Now the U-Boot is ready for performing ramdisk deployment.

4.3.2 Configuring U-Boot for NFS deployment

The NFS file system generated by LTIB allow you perform NFS deployment. Before performing NFS deployment, it is needed to configure U-Boot parameters.

At u-boot prompt, set u-boot environment like:

```
=>setenv ipaddr <board_ipaddress>

=>setenv serverip <tftp_serverip>

=>setenv gatewayip <your_gatewayip>

=>setenv bootargs root=/dev/nfs rw
nfsroot=<tftp_serverip>:<nfs_root_path
ip>=<board_ipaddress>:<tftp_serverip>:<your_gatewayip>:<your_netmask>:
mpc8555cds:eth0:off console=ttyS1,115200

=>saveenv
```

Now the U-Boot is ready for performing NFS deployment.

4.4 Development Deployment

4.4.1 Ramdisk deployment

1. Setting U-Boot environment

Please set U-Boot environment as described in chapter 4.3.1.

2. Booting up board

tftp images to the board, then boot it up.

```
=>tftp 1000000 uImage

=>tftp 2000000 rootfs.ext2.gz.uboot

=>bootm 1000000 2000000
```

Now your board boots up with ramdisk filesystem.

4.4.2 NFS deployment

1. Setting host NFS server environment

- On the Linux host NFS server, please add the following line in the file `/etc/exports`

  `nfs_root_path board_ipaddress(rw,no_root_squash,async)`

- Restart the nfs service:
  `/etc/init.d/nfs restart`

NOTE:

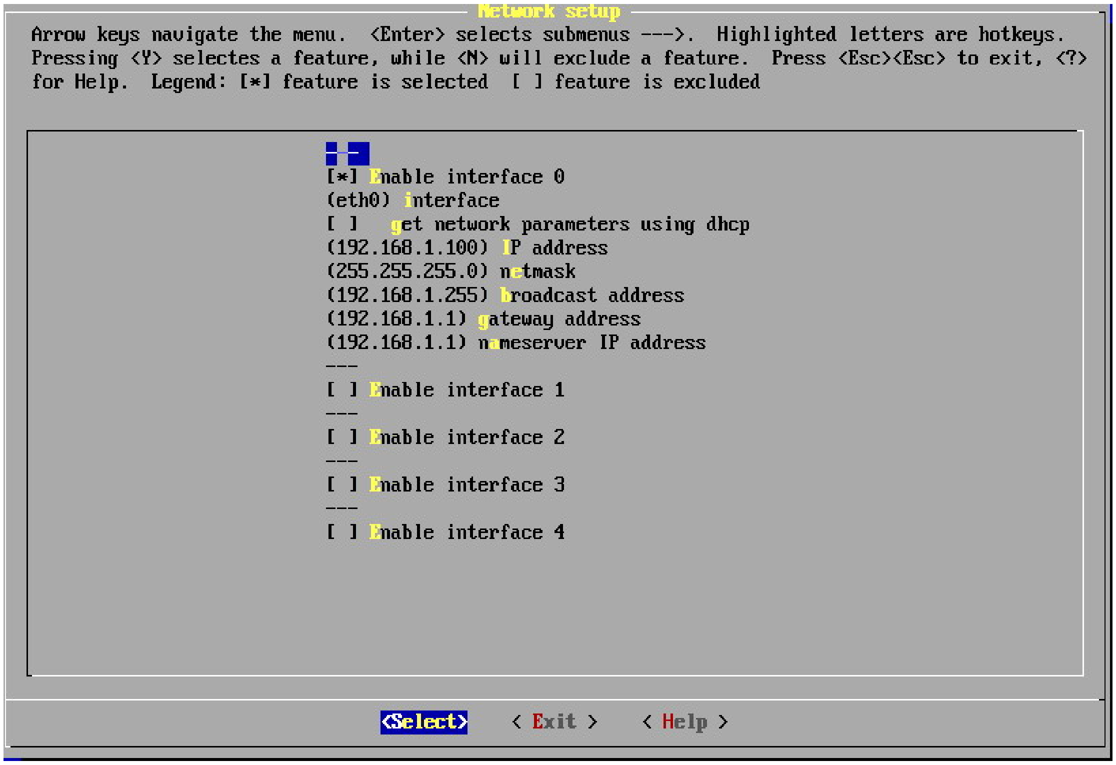
1. *`nfs_root_path:`* the NFS root directory path on NFS server.

2. Setting LTIB configuration

It is need to modify the board’s network configuration in LTIB:

```
./ltib --configure
```

Select *Target System Configuration*->*Options*->*Network setup*, Fig. 4-1 appears on the screen:



Network setup

```
Arrow keys navigate the menu.  <Enter> selects submenus --->.  Highlighted letters are hotkeys.
Pressing <Y> selectes a feature, while <N> will exclude a feature.  Press <Esc><Esc> to exit, <?>
for Help.  Legend: [*] feature is selected  [ ] feature is excluded

[*] Enable interface 0
(eth0)  interface
[ ]    get network parameters using dhcp
(192.168.1.100) IP address
(255.255.255.0) netmask
(192.168.1.255) broadcast address
(192.168.1.1) gateway address
(192.168.1.1) nameserver IP address
---
[ ] Enable interface 1
---
[ ] Enable interface 2
---
[ ] Enable interface 3
---
[ ] Enable interface 4
```

Fig. 4-1

Please modify the items according to your network configuration. If there is any problem, please ask your IT for help.

3. Setting U-Boot environment

Please set U-Boot environment as described in chapter 4.3.2.

4. Booting up board

tftp kernel image to the board, then boot it up.

```
=>tftp 1000000 uImage
```

```
=>bootm 1000000
```

Now your board boots up with NFS filesystem.

4.5 Production Deployment

The kernel and ramdisk can be programmed into the flash of the board for production deployment.

4.5.1 Setting U-Boot environments

U-Boot environment should be set up for production deployment as following:

```
=>setenv ipaddr <board_ipaddress>

=>setenv serverip <tftp_serverip>

=>setenv gatewayip <your_gatewayip>

=>setenv bootargs root=/dev/ram rw console=ttyS1,115200

=>setenv bootcmd bootm ff800000 ffa00000

=>saveenv
```

NOTE:

1. `ff800000:` the address that the kernel will be programmed to.
2. `ffa00000:` the address that the ramdisk will be programmed to.

4.5.2 Program kernel to flash

The kernel should be tftped to the ram, then be copied to the flash at address 0xff800000. To program the kernel to flash, please execute the following commands at U-Boot prompt:

```
=>tftp 1000000 uImage

=>erase ff800000 ff9fffff

=>cp.b 1000000 ff800000 200000
```

NOTE:

1. `200000:` To save time, it could be the actual size of kernel image.

4.5.3 Program ramdisk to Flash

The ramdisk should be tftped to the ram, then be copied to the flash at address 0xffa00000. To program the ramdisk to flash, please execute the following commands at U-Boot prompt:

```
=>tftp 2000000 rootfs.ext2.gz.uboot

=>erase ffa00000 fff7ffff

=>cp.b 2000000 ffa00000 580000
```

NOTE:

1. `580000:` To save time, it could be the actual size of ramdisk image.

4.5.4 Booting up board

The kernel can boot up automatically after the board powered on, or the following command can be used to boot up the board at U-Boot prompt:

```
=>bootm ff800000 ffa00000
```