

Everest-MI-V-1G-Ethernet-Webserver-Demo

Getting Started

Project: Everest-MI-V-1G-Ethernet-Webserver-Demo Getting Started		created:	S. Rieche	Date	2019-10-29
		edited:	S. Rieche	Date:	2019-10-29
		approved:		Date:	
Filename: Everest-MI-V_1G_Ethernet_Webserver-DemoGetting_Started_V1_1.docx					
Arrow Central Europe GmbH		Version:	1.1	Page 1 of 7	15

Contents

1.	Revisio	on History	5
1.1	Revis	ion 1.1	5
1.2	Revis	ion 1.0	5
2.	Getting	Started	6
2.1	Preree	quisites	6
2.2	Handl	ing the Board	7
2.3	Board 2.3.1	I-Setup Revision PROTO Toggle-Switch S1 – PCIe	7 7
	2.3.2	Toggle -Switch S5 – SC SPI-Flash enable	7
	2.3.3	DIP-Switch S8 – FMC Voltage Selector	7
	2.3.4	Toggle -Switch S9 – VDDAUX2 & VDDAUX5 Voltage	7
2.4	Board 2.4.1	I-Setup Revision A and B Toggle-Switch S1 – PCIe	8 8
	2.4.2	Toggle -Switch S5 – SC SPI-Flash enable	8
	2.4.3	DIP-Switch S8 – FMC Voltage Selector	8
	2.4.4	Toggle -Switch S9 – VDDAUX2 & VDDAUX5 Voltage	8
2.5	Powe	ring up the Board	9
3.	Demo [Design	0
3.1	Prere	quisites1	0
3.2	Desig	n Implementation1	1
3.3	Runni	ing the Design1	2

Figures

Figure 1: Everest Board	. 9
Figure 2: Design Implementation	11
Figure 3: SoftConsole v5.2 workspace launcher	12
Figure 4: SoftConsole v5.2 - starting the debug session	12
Figure 5: SoftConsole v5.2 - running the design	13
Figure 6: PolarFire com ports in the Windows Device Manager	13
Figure 7: initialization printouts on a hyperterminal (putty)	14
Figure 8: webserver main page	15

User Guide

Tables

Table 1: Software / IP Requirements	10
Table 2: Hardware Design Clock Frequencies	11

1. Revision History

The revision history describes the changes that were implemented in the document. The changes are listed by revision, starting with the most current publication.

1.1 Revision 1.1

Project updated for Libero v12.2

1.2 Revision 1.0

Revision 1.0 is the first publication of this document.

2. Getting Started

This demo design implements a MI-V soft processor subsystem with 1G Ethernet and UART functionality running a webserver application. Internal SRAM blocks are used for both, program and data memory. The UART terminal operates through USB connector J9.

2.1 Prerequisites

For the Everest Cortex M1 Demo the following is needed:

Item	Quantity
Everest DEV Board	1
12 V / 5 A wall-mounted power adapter	1
USB 2.0 A male to mini-USB B cable for	1
UART / Programming interface to PC	
Free one-year Libero Silver software license	1
CoreTSE license	1
Ethernet cable	1
PC with web browser	1

Note 1: The Everest DEV Board offers an on-board FlashPro5 programmer, which can be used to program and debug with Identify, SmartDebug and embedded application software using SoftConsole.

Note 2: A precompiled and generated stp-file is included. Therefor the CoreTSE license is not needed for firmware only development.

2.2 Handling the Board

Pay attention to the following points while handling or operating the board:

Handle the board with electrostatic discharge (ESD) precautions to avoid damage.

For information about ESD precautions see

https://www.microsemi.com/documentportal/doc_view/126483-esd-appnote.

2.3 Board-Setup Revision PROTO

2.3.1 Toggle-Switch S1 – PCle

Warning: S1-1 and S1-2 must not be at position on at the same time!

SWITCH ON	PCIe LANES
S1-1	x1
S1-2	x4

2.3.2 Toggle -Switch S5 – SC SPI-Flash enable

Warning: S5-1 and S5-2 must not be at position on at the same time!

SWITCH ON	SC SPI-FLASH
S5-1	ENABLE
S5-2	DISABLE

2.3.3 DIP-Switch S8 – FMC Voltage Selector

Warning: S8-1 to S8-4 must not be at position on at the same time!

SWITCH ON	FMC VOLTAGE
S8-1	3.3 V
S8-2	2.5 V
S8-3	1.8 V
S8-4	undefined (not connected)

2.3.4 Toggle -Switch S9 – VDDAUX2 & VDDAUX5 Voltage

Warning: S9-1 and S9-2 must not be at position on at the same time!

SWITCH ON	VDDAUX2 & VDDAUX5
S9-1	2.5 V
S9-2	FMC voltage

2.4 Board-Setup Revision A and B

2.4.1 Toggle-Switch S1 – PCle

SWITCH	PCIe LANES
S1-1 (marking)	x4
S1-2	x1

2.4.2 Toggle -Switch S5 – SC SPI-Flash enable

SWITCH	SC SPI-FLASH
S5-1 (marking)	DISABLE
S5-2	ENABLE

2.4.3 DIP-Switch S8 – FMC Voltage Selector

SWITCH	FMC VOLTAGE
S8-1 off, S8-2 off	1.8 V
S8-1 on, S8-2 off	2.5 V
S8-1 off, S8-2 on	undefined (not recommended)
S8-1 on, S8-2 on	3.3 V

2.4.4 Toggle -Switch S9 – VDDAUX2 & VDDAUX5 Voltage

SWITCH	VDDAUX2 & VDDAUX5
S9-1 (marking)	2.5 V
S9-2	FMC voltage

User Guide



Figure 1: Everest Board

2.5 Powering up the Board

The Everest DEV Board is powered up using either the 12 V DC jack or the PCIe connector. For programming connect it although with your computer using USB mini B connector J9.

3. Demo Design

3.1 Prerequisites

Table 1: Software / IP Requirements

Software	Version
Libero SoC PolarFire	V12.2
Synplify Pro	O2018.09M-SP1-1
FlashPro PolarFire	V2.0
IP	
PF_CLK_DIV	v1.0.103
CORETSE_AHB	v3.1.102
CoreAHBLite	v5.3.101
COREJTAGDEBUG	v2.0.100
CORERESET_PF	v2.0.112
MIV_RV32IMA_L1_AHB	v2.0.100
PF_INIT_MONITOR	v2.0.100
PF_SRAM_AHBL_AXI	v1.1.125
PF_IOD_CDR	v1.1.200
CoreTimer	v2.0.103
PF_CCC	v2.1.104
CoreUARTapb	v5.6.102
COREAHBTOAPB3	v3.1.100
CoreAPB3	v4.1.100
CORESPI	v5.1.104

Before you start you have to make sure, that all cores are downloaded to your local vault.

3.2 Design Implementation

The following table lists the clock frequencies used in the design.

Table 2: Hardware Design Clock Frequencies

Clock	Frequency (MHz)
PF_CCC REF_CLK_0	50
PF_CCC OUT0_FABCLK	100
HCLK / PCLK	100



Figure 2: Design Implementation

The design is already fully implemented and ready to be programmed on the Everest Board. The board has to be connected with the power supply and to the PC with the USB cable. All drivers have to be installed (which should happen automatically when plugged in the first time) To program the design, there are two possibilities:

- Programming via Libero PolarFire SoC: Programming is started with the "Run PROGRAM Action" Button in the Design Flow Pane
- Programming via FlashPro Software: For preproduction and production devices use the STAPL-file in the "Bitstream" folder. The STAPL-file for engineering samples is located in the folder "Bitstream_ES".A new FlashPro project has to be generated and the programming file loaded into.

3.3 Running the Design

In order to run the design, the RISCV-Processor has to be loaded with the firmware. To do so, load the provided SoftConsole Workspace.

SC Eclipse Launcher	×
Select a directory as workspace	
Microsemi SoftConsole v5.2 uses the workspace directory to store its preferences and developme	nt artifacts.
Workspace: bero_v12p2\Everest-MI-V-1G-Ethernet-Webserver-Demo\SoftConsole_v5p2 >>	Browse
Use this as the default and do not ask again	
Recent Workspaces	
	Launch Cancel

Figure 3: SoftConsole v5.2 workspace launcher

	nt 🕅 te
	C 1 1 1 1 1
🗞 Project Explorer 12 🖻 🖏 🕼 👻 🐃 🗖 👔 maine 12 🖟 corre_timene 🔅 hww.reg.accers.5 🕼 corre_time.e. 🕼 clock-arche. 🖓 clock-arche. 🦓 dhepe.e. 👔 dhepe.e.	
<pre>> @ up,webser#</pre>	#, ₽ ▽
🗄 Problems 🔅 Tasks 🗳 Console 21 🗇 Properties 🧳 Search 🛃 🖻 🖛	3•- 0
No conside to display at this time.	
Writable Smart Insert 198.1	

Figure 4: SoftConsole v5.2 - starting the debug session

A debug configuration is provided to download the firmware to the RISCV processor and start the application.

User Guide

Se SoftConsole_v5p2 - uip_webserver/main.c - CiLlbero_WS\LABS\Libero_v12p2\Everest-MI-V-1G-Ethernet-Webserver-Demo\SoftConsole_v5p2 - Microsemi SoftConsole v5.2 File Edit Source Refactor Navigate Search Project Run Window Help		- 0 ×
□ • □ □ □ □ □ □ • □ • □ • □ • □ = N 3. ○ □ + 元 2. 2. 4 • • 0 • 9. • 2 ● 2 + · 0 • 9. • 0 • 0 • · 0 • · · · · ·		Quick Access 🛛 😰 🛛 🔯
t≱ Debug ⊠ 🗽 💌 🗆 🗎	00- Variables 💁 Breakpoints 😰 1111 Registers 🛋 Modules 🌄 Peripherals	X 💥 🖉 🕾 🔍 🖽 🖂 🌣 🗖 🗆
Classic Constant Constant (Constant) Constant (Constant) V V Veget (Constant) Constant (Constant) Veget (Const) Veget (Constant) Veget (Constant) Veget (Const)	 [™] amalac (function main() (type Temporary) 	
	¢	
e maine 22 e core_timer.c S hw_reg_access.S e) core_tse.c e) clock-arch.c e) dhcpc.c		E Outline 🔤 Disassembly 23 📟 🗖
<pre>102 display_list_ytetus(); 103 } 104 / main 105 / main 105 / main 107 /</pre>		Description here Image Notes Image Notes
ug automot Delang (GES OpenCCD Delauging) opencet.ex (455) creates (/22) (455) creates (/22		
	Writable	Smart Insert 198 : 1

Figure 5: SoftConsole v5.2 - running the design

At startup the firmware checks if the configuration of the clock conditioning circuit ZL30722 is appropriate for the design and update it when necessary.

The link status is printed out over the uart interface and can be read out on a Hyperterminal console. The Hyperterminal should be configured to the following parameters:

- Baud Rate 115200, 8 Databits, 1 Stopbit, no parity (8N1)
- COM-Port is visible in Windows Device Manager (mostly the highest number of the four FlashPro5 ports, here COM18):

Geräte-Manager	-	×
Datei Aktion Ansicht ?		
🗸 🛱 Anschlüsse (COM & LPT)		^
FlashPro5 Port (COM15)		
🛱 FlashPro5 Port (COM16)		
FlashPro5 Port (COM17)		
🛱 FlashPro5 Port (COM18)		
> 4 Audio, Video und Gamecontroller		
> 📢 Audioeingänge und -ausgänge		
> 🚡 Bildverarbeitungsgeräte		
> 🛄 Computer		\sim

Figure 6: PolarFire com ports in the Windows Device Manager

Arrow Central Europe GmbH

User Guide

{

static void init(void)

```
.
.
.
uip_ipaddr(ipaddr, 169, 254, 97, 129);
uip_sethostaddr(ipaddr);
uip_ipaddr(ipaddr, 169, 254, 97, 130);
uip_setdraddr(ipaddr);
uip_ipaddr(ipaddr, 255, 255, 0, 0);
uip_setnetmask(ipaddr);
httpd_init();
display_link_status();
```

COM18 - PuTTY	_	\times
uart init done		^
2L30722 eeprom equals 2L30722 image file nothing to be done		
Ethernet link up: 1000Mbps Full Duplex		
		× .

Figure 7: initialization printouts on a hyperterminal (putty)

After starting the application in debug mode, the webserver is reachable with a standard browser on the configured IP-Address:

😂 Welcome to the uIP w	eb server! - Internet Ex	plorer			<u>_ ×</u>
🔆 🕞 マ 🩋 http://16	9.254.97.129/ 🔎 💌	😼 逡 Welcome to the	e uIP web ser ×		☆ ☆
Front page	File statistics	Network statistics	Network connections]	
These web pages are se	rved by a small web server	running on top of the uIP em	bedded TCP/IP stack.		
Click on the links above f	or web server statistics.				

Figure 8: webserver main page