Freescale Semiconductor Application Note

U-Boot Debug using CodeWarrior for QorIQ LS series – ARM V7 ISA

1. Introduction

This document describes the steps required for U-Boot debugging using the CodeWarrior for QorIQ LS series – ARM V7 ISA.

This document includes the following sections:

- Build the U-Boot sources.
- Perform U-Boot debug in CodeWarrior for QorIQ LS series ARM V7 ISA.

2. Preliminary background

The following are the steps required to compile LS1021A U-Boot for the LS1021AQDS board.

Contents

1.	Introduction1
2.	Preliminary background1
3.	Create ARMv7 project2

4. U-Boot debug support......7



Create ARMv7 project

2.1. Downloads

Before U-Boot debug, following downloads are necessary:

- Linaro GCC 4.8 (Aarch32) 4.8-2013.12 toolchain
- U-Boot source code

You can gcc-linaro 4.8 (Aarch32) 4.8-2013.12 toolchain from <u>http://www.linaro.org/downloads/</u> or you can use the one installed with Linux version of CodeWarrior for QorIQ LS series – ARM V7 ISA.

U-boot source code will be provided together with SDK for LS1021AQDS board.

2.2. Compiling U-Boot

To compile U-Boot, perform these steps (UBUNTU OS was used to build U-Boot):

- 1. Go to U-Boot folder
- 2. Set the ARCH and CROSS_COMPILE environmental variables and build the U-Boot:

```
make ARCH=arm CROSS_COMPILE=<path_to_toolchain>/arm-linux-
gnueabihf- ls1021aqds nor
```

```
or
export ARCH=arm
export CROSS_COMPILE=<path_to_toolchain>/arm-linux-gnueabihf-
make ls1021aqds_nor
```

3. U-Boot image and U-Boot binary will be placed in U-Boot folder.

NOTE Default U-Boot will be built with dwarf-2 debug format.

3. Create ARMv7 project

To create an ARMv7 bare metal project for U-Boot debug, follow these steps:

- 1. Start CodeWarrior for QorIQ LS series ARM V7 ISA.
- 2. Choose **File** > **Import** to import the U-Boot executable file generated during the U-Boot compilation. It can be found in U-Boot folder.



Figure 1. CodeWarrior File menu

3. Choose the source to Import and select Next.

Figure 2. Importing dialog

a Import	
Select Import a CodeWarrior Executable file and create a project	Ľ
Select an import source:	
type filter text	
 C/C++ CodeWarrior Example Project Install Run/Debug Team 	

Create ARMv7 project

4. Specify Project name and Location, or use the default location and select Next.

Figure 3. Importing executable file dialog

	Import a CodeWarrior Executable file							
mport a Code Choose the loc	Warrior Execut ation for the new	table file						
Project name:	LS1021AQDS							
🔽 Use defaul	t location							
Location: D:\	workspace\arm7	140516\LS1021AQ)DS			Browse		

5. Browse to the U-Boot executable file and select **Open**. By default, CodeWarrior looks for an .elf extension, so change the file type in the lower right corner of **Select File** dialog.



File or folder not specifi	ed				
ile to import				Browse	
Copy the selected f	ile to cu	irrent project folder			
Coo - V « arm	v7 ▶ Is	i1-uboot ►	- ↓	Search Is1-uboot	
Organize 🔻 New	folder			8== -	
-	•	Name		Date modified	Туре
詞 Libraries		Makefile		07-May-14 11:53	File
Documents	_	mkconfig		07-May-14 11:52	File
👌 Music		README		07-May-14 11:53	File
E Pictures		rules.mk		07-May-14 11:52	MK File
Videos	=	snapshot.commit		07-May-14 11:52	сомм
		System.map		07-May-14 12:08 P	CodeW
📜 Computer		u-boot		07-May-14 12:08 P	File
Primary (C:)		u-boot.bin		07-May-14 12:08 P	BIN File
DATA (D:)		u-boot.lds		07-May-14 12:08 P	LDS File
👷 marius_home (\	7	u-boot.map		07-May-14 12:08 P	CodeW
🚽 sdk (\\10.171.72	7 4 4		III		

6. Select **Processor** type for the project and select **Next**.

Create ARMv7 project

Figure 5. Select Processor type

Import a CodeWarrior Executable	file		
Processor			
Choose the processor for this proj	ect		
Processor			
type filter text			
Layerscape Family		 	
QorIQ_LS1			
LS1020A			
LS1021A			
LS1022A			
Toolchain			
Bareboard Application			
Cinux Application			
Target OS			
None			
C Home			
C Linux Kernel			
C Linux Kernel			
C Linux Kernel			
🗇 Linux Kernel			
C Linux Kernel		 	

7. Select Debugger Connection Types, Board, Launch, Connection Type and select Next.

🥦 Import a CodeWar	rior Executable fil	e			
Debug Target Sett Target Settings	ings				P
Debugger Connecti Hardware Emulator Board Launch Download	on Types 1021AQDS Connection	Ţ	*		
🔽 Attach	Pefault		+		
Connection Type C	odeWarrior TAP (over USB) 🔻	<u></u>		
(?)		< Back	Next >	Finish	Cancel

- NOTE By default U-Boot will be generated as "Shared object file" and not as "Executable file". Using Download Launch will not work in this case; Attach Launch will be used instead.
 If U-Boot is not available on target board, Flash Programmer should be used to program U-Boot on target board.
- 8. Select the Configurations that you want to create and then, select Finish to close the wizard.

Figure 7. Select Configurations dialog

🥬 Import a CodeWarrior Executable file	
Configurations	
Choose the configurations you want to create	
Core index	
Core 0	
Core 1	
< Back Next > Finish	Cancel

4. U-Boot debug support

4.1. Debug environment

Use the following setup for U-Boot debugging on ARMv7 core:

- LS1021AQDS board.
- Compiled U-Boot for the NOR FLASH target.
- Flash U-Boot on the target board (for more information on how to program the U-Boot to NOR flash, see SDK documentation).
- Switches set for NOR boot (for more information on how to set switches, see SDK documentation).
- Latest release of CodeWarrior for QorIQ LS series ARM V7 ISA.

U-Boot debug support

• CWTAP probe.

4.2. Start U-Boot debugging

The U-Boot executable file generated during the U-Boot compilation should be imported as CodeWarrior project (for more information, see <u>Create ARMv7 project</u>).

After the CodeWarrior project is created, perform these steps to start U-Boot debug:

1. Choose **Run** > **Debug configurations**, to open **Debug configurations** dialog and select **Debug**.

Figure 8. Debug Configurations dialog

C/C++ - CodeWarrior	Development Studio		
File Edit Source Refa	actor Navigate Search Project Run w		F\$ \$
Image: CodeWarrior Project Image: CodWarrior Project Ima	Debug Configurations Create, manage, and run configuration Debug or run an application to a target.		x x
∰ Binaries ≥ Debug ♀ u-boot	Image: Second Secon	Name: LS1021AQDS_Debug_LS1021A_Attach Main %4 Arguments Source Environment Common Debug session type Choose a predefined debug session type or custom type for maximum flexibility Download Connect Ø Attach Custom Custom C/C++ application V C/C++ application Build (if required) before launching Build (if required) before launching Select configuration using 'C/C++ Application' Enable auto build Disable auto build Disable auto build Use workspace settings Configure Workspace Settings	
	Image: wide of the second s		Revert
 Build settings Debug settings 	?	Debug	Close

2. The connection initializes and configures the TAP, and then it will attach to board.

Figure 9. Debug view



3. To reinitialize the target from CodeWarrior, select Reset as shown in the figure.

Figure 10. Reset dialog

Debug 🛛 🧏 🎉 🔿 🕩 💷 🔳	* ←i = 1. e = * %	2" 🖤 🖷 🛲 m 🔻	V = (X) = V	ariables 💁 Breakpoints 🖾	🚺 Cache 🚻 Reg
 LS1021AQDS_Debug_LS1021A_Attach CodeWarrior ARM V7 Debugger, u Thread [ID: 0x0] (Running) D:\workspace\arm7140516\LS1021 	Nan	ne -	Context		
	🥬 Reset		6		
	Execute a target reset:				
	Target	Run out of reset	Initialize target	Initialize target script	Move Up
	▲ LS1021A				Move Down
	Cortex-A7-0 Cortex-A7-1				Restore Orde
	Note: Target initialization Reload settings from the f	n files only apply to debug target configuration: Re	gged cores.	Reset	Cancel

NOTE Make sure no initialization file is selected.

4. After reset debugger will prompt for source location.

U-Boot debug support

Figure 11. File location dialog

🖸 qixis_write() at /SDK 🔉 - uboot/board/freescale/common/qixis.c: 41 🛛	- 8
Can't find a source file at "/SDK/Is1-uboot/board/freescale/common/qixis.c"	
Locate the file or edit the source lookup path to include its location.	
View Disassembly	
Locate File	
Locate File	
Edit Source Lookup Path	
Apply to Common Source Lookup Path	

5. After the path is provided, source will become available in CodeWarrior.

Figure 12. File editor



6. Set a hardware breakpoint at start, using Debugger Shell command bp -hw start.

Figure 13. Debuger Shell view



7. Resume using F8 or Debugger Shell command go.

Figure 14. Debuger Shell view



8. Breakpoint will be hit and U-Boot debugging can be performed from start.

Figure 15. File editor

§ start.5 🖇	10
/*	~
* armboot - Startup Code for OMAP3530/ARM Cortex CPU-core *	1
* Copyright (c) 2004 Texas Instruments <r-woodruff2@ti.com> *</r-woodruff2@ti.com>	
<pre>* Copyright (c) 2001 Marius GrÄgger mæg@sysgo.de> * Copyright (c) 2002 Alex ZÄXpte <aru@sysgo.de> * Copyright (c) 2002 Gary Jennejohn <garyj@denx.de> * Copyright (c) 2003 Richard Woodruff <r-woodruff2@ti.com> * Copyright (c) 2006-2008 Syed Mohammed Khasim <x0khasim@ti.com> * SPDX-License-Identifier: GPL-2.0+ */</x0khasim@ti.com></r-woodruff2@ti.com></garyj@denx.de></aru@sysgo.de></pre>	
<pre>#include <asm=offsets.h> #include <config.h> #include <version.h> #include <asm system.h=""> #include <linux linkage.h=""></linux></asm></version.h></config.h></asm=offsets.h></pre>	
.globl _start	
start: b reset ldr pc, _undfind_instruction ldr pc, _software_interrupt ldr pc, _prefetch_abort ldr pc, _data_abort ldr pc, _inct_used ldr pc, _irug ldo pc, _fro	*
۲ ۱۵۱ pc, ۱۱۹ ۲	

U-Boot debug support

- 9. Debugging (step, run, or breakpoint) can be done till the U-Boot boot up.
 - **NOTE** If you encounter reset skid issue, the program will not stop at _start symbol. As a workaround you can set a hardware breakpoint at _start and move PC to _start symbol address. This issue has been resolved in FPGA v11 image.

How to Reach Us:

Home Page: www.freescale.com

E-mail: support@freescale.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 reserves the right to make changes without further notice to any products herein. Freescale makes no warranty, representation, or guarantee regarding the suitability of its products for any particular purpose, nor does Freescale 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 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 does not convey any license under its patent rights nor the rights of others. Freescale sells products pursuant to standard terms and conditions of sale, which can be found at the following address: <u>freescale.com/SalesTermsandConditions</u>.

Freescale, the Freescale logo, CodeWarrior, and QorlQ are trademarks of Freescale Semiconductor, Inc., Reg. U.S. Pat. & Tm. Off. Layerscape is trademark of Freescale Semiconductor, Inc. All other product or service names are the property of their respective owners. ARM, Cortex and TrustZone are trademarks or registered trademarks of ARM Ltd or its subsidiaries in the EU and/or elsewhere. All rights reserved.

© 2014 Freescale Semiconductor, Inc.



