Ronetix Application Note AN013: Multi-core Linux Kernel Debugging (i.MX6 SMP and AMP Debugging)

1. Overview

This article shows you how to debug Linux kernel running on multi-core CPU using PEEDI. It also briefly explains how to do an AMP debugging.

2. Prerequisites

- a PEEDI
- a SABRE Lite i.MX6 Development Board
- Linux kernel sources for the SABRE board – A host machine running Linux and a TFTP server to build the kernel for debugging

3. Downloading and building the kernel

You can download the kernel from https://github.com/boundarydevices/linux-imx6/tree/boundary-imx_3.0.35_4.0.0 Here you can also read how to configure it and build it. Do not forget to configure it to be built with “Debug info”, you can find this option in the “Kernel hacking” section of the config menu. After the kernel is built, you have to copy it from **/arch/arm/boot** to he SD card so the board can boot it. Next you have to use **nm vmlinux | grep start_kernel** command to get the address of the start_kernel() function and put it in the INIT section of the PEEDI configuration file so the execution will break there and wait for you to start debugging.

4. Setting PEEDI

First you need to prepare your CFG file. Since the CFG file itself is not that simple, you can download a ready to use one from https://download.ronetix.at/peedi/cfg_examples/cortex-a/mx6_smp.cfg. The following covers only the FLASH settings. You have to copy the CFG file to your TFTP server root directory, so PEEDI will be able to download it. Next you need to tell PEEDI to load the file using the **config tftp://SERVER_IP/mx6.cfg** command. The mx6.cfg file is a standard multi-core configuration file with the exception of that the CORE0 is set to a XXX_SMP core type. This tells PEEDI to manage all four cores in the background, but present them to gdb as a single core. This simplifies SMP multi-core debugging to a conventional single core one. Here is a part of the CFG file which shows the cores’ configuration.

```
[PLATFORM_CortexA8]
...
CORE0 = iMX6A_SMP, 2, 0xBA00477 ; TAP is Cortex-A SMP master CPU
CORE0_STARTUP_MODE = RESET ; stop the core immediately after reset
CORE0_ENDIAN = LITTLE ; core is little endian
CORE0_BREAKMODE = SOFT ; breakpoint mode
CORE0_INIT = INIT_LINUX ; init section with working U-BOOT
...
CORE1 = iMX6B, 2, 0xBA00477 ; TAP is Cortex-A CPU
CORE1_STARTUP_MODE = RESET ; stop the core immediately after reset
CORE1_ENDIAN = LITTLE ; core is little endian
CORE1_BREAKMODE = SOFT ; breakpoint mode
```

```
CORE2 = iMX6C, 2, 0xBA00477 ; TAP is Cortex-A CPU
CORE2_STARTUP_MODE = RESET ; stop the core immediately after reset
CORE2_ENDIAN = LITTLE ; core is little endian
CORE2_BREAKMODE = SOFT ; breakpoint mode

CORE3 = iMX6D, 2, 0xBA00477 ; TAP is Cortex-A CPU
CORE3_STARTUP_MODE = RESET ; stop the core immediately after reset
CORE3_ENDIAN = LITTLE ; core is little endian
CORE3_BREAKMODE = SOFT ; breakpoint mode

[INIT_LINUX]
break add hard 0x80008620 ; kernel break address got by 'nm vmlinux | grep start_kernel'
go
wait 20000 stop
break del all
beep 100 100
```

As you can see from the above, there is nothing special in the INIT section – it is just normal Linux kernel debugging one, it set a hardware breakpoint at the beginning of the **start_kernel()** function, starts the CPU, waits for it to hit the break and beeps to inform the user that it is time to start GDB for debugging.

5. Altera Cyclone V SoC debugging

The Altera Cyclone V SoC is an exception considering the multi-core debugging. You must set the PEEDI CFG file for single core debugging when debugging code prior to the code that starts all core, so setting CORE0 to

```
CORE0 = CYCLONE_VA, 0, 0x4BA00477
```

This is the CFG file – https://download.ronetix.at/peedi/cfg_examples/cortex-a/cyclone5.cfg
If you intend to debug code after that point you need to configure the CORE0 for SMP debugging:

```
CORE0 = CYCLONE_VA_SMP, 0, 0x4BA00477
```

Here is the CFG file – https://download.ronetix.at/peedi/cfg_examples/cortex-a/cyclone5_smp.cfg

6. AMP Debugging

For asymmetric multiprocessing (AMP) debugging, you need to set the CORE0 to:

```
CORE0 = iMX6A, 2, 0xBA00477
```

This way PEEDI opens a gdb remote listening port for each core so you can start an eclipse/gdb session for each core with the corresponding ELF file.

Share on LinkedinShare on FacebookShare on Reddit on TwitterShare Via EmailShare on Whatsapp

Share