Linux Porting Guide for AM571x/AM570x Speed Grades

Contents

Overview

The AM57x Family of Processors includes a wide range of operating performance to meet the needs of a number of broad applications. Among these options are a variety of speed grades to meet different performance points. These devices have a number of specialized cores to provide applications specific computation capabilities. These cores can be run at different speeds to fine tune the processor to the needs of the application, power budget, thermal characteristics, etc.

The Processor SDK for Linux is a software development package provided to speed development by providing a software reference. This package now includes support for then entire AM57x family of processors which can be broken down into the AM572x, AM571x, and AM570x sets of devices or sub-familes. Most of the devices in this family are supported by the Processor SDK for Linux right out of the box. This support is tested and validated on TI designed EVMs. These EVMs use the highest performance devices in the family in order to allow users to evaluate the entire spectrum of performance.

In order to move the Processor Linux SDK from a TI EVM to another customized board, a "board port" of Linux must occur to adapt the software to any of the changes between the TI EVMs and the custom board. Since Linux requires a bootloader, that must be "board ported" as well. The Processor Linux SDK used U-Boot as the bootloader. So, U-Boot and Linux need to be adapted to the custom board to account for any differences between it and the EVMs that TI provides. One of these differences might be a different speed grade of the processor. In that case, changes may be necessary as part of the board port to allow for this new speed grade. This document highlights the changes that need to be considered in this board port effort to other speed grades of the AM57xx family of devices.

For example, the AM571x Industrial Development Kit is designed with the AM5718xXx device. This device, which has an "X" speed grade, is rated at 1 GHz operation for the ARM Cortex-A15 core with the ability to go to 1.5 GHz under certain conditions (Please see the AM5718 Data Sheet (http://www.ti.com/lit/ds/symlink/am5718.pdf), section 9.1 Device Nomenclature, for more information). The Processor SDK for Linux includes code in U-Boot and Linux Kernel to set the processor up with this configuration. If a custom board wanted to use a different speed or configuration, the software would need to be changed accordingly as part of the board port.

The AM571x and AM570x series include "D" speed grade parts that are characterized at 500 MHz operation for the ARM Cortex-A15 core. The AM570x series also includes a "J" speed grade that is different than the "X" speed grades for the AM572/1x. If one of these devices is being used on the custom board, U-Boot and Linux need to be changed to account for this difference. If this change is not made, the device could be running out of specification. These changes may reach across other cores and clocks on the device as well, depending on what speeds they need to operate at. This document is not an exhuastive list of all the changes needed for a proper board port as it focused on the changes needed to enable different speed grades.

U-Boot Changes to Consider

U-Boot is the bootloader used with the Processor SDK for Linux. Much of the clock configuration is done one time within U-Boot to configure the processor for correct operation.

Add 500 MHz for MPU Operation

When using "D" rated parts that run at 500 MHz, a clock table entry for this speed needs to be added to the U-Boot sources in arch/arm/cpu/armv7/omap5/hw_data.c.

```
static const struct dpll_params mpu_dpll_params_500mhz[NUM_SYS_CLKS] = {
        {-1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1},       /* 12 MHz   */
        {200, 7, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1},       /* 20 MHz   */
        {-1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1},       /* 16.8 MHz */
        {365, 13, -1, -1, -1, -1, -1, -1, -1, -1, -1},          /* 19.2 MHz */
        {-1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1},       /* 26 MHz   */
        {500, 26, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1},      /* 27 MHz   */
        {-1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1},       /* 38.4 MHz */
};
```

Change Structures to Use New Clock Table Entry

In order to use this new entry for the MPU, the dplls structure that is being used needs to be modified. The file that configures this for the IDK is board/ti/am57xx/board.c. This particular file is where most of the changes for a board port of U-Boot to a new board should exist. As shipped, this file uses the dra7xx_dplls structure back in hw_data.c. This structure should be modified to use the 500 MHz dpll_params structure added above.

To boot at 500MHz, modify the the dplls structure to use 500MHz instead of 1GHz:

```
.mpu = mpu_dpll_params_500mhz,
```

This structure is also used to set several other clocks. Take time to make sure that these settings also comply with the part that is being used on the custom board.

Configure DDR Interfaces

An important one to consider is the speed of the DDR memory. The clock for the DDR is selected using the same dplls structure. Some higher speed grade parts support a 667 MHz DDR clock, but some of the lower speed grade parts only support a 533 MHz DDR3 clock. Make sure to choose the appropriate DDR clock for the device on the custom board.

Over in the board/ti/am57xx/board.c file, make sure that the EMIF is being configured correctly for the appropriate speed, and that the appropriate number of EMIFs is being selected to match the part being used. The Processor SDK for Linux includes board detection logic along with a defined EEPROM structure included on TI boards to make executing the same U-Boot executable on multiple TI boards possible. For example, one U-Boot binary will execute correctly on an AM572x GP-EVM, AM572x IDK, and AM571x IDK even though an AM572x part has 2 DDR interfaces running at 533 MHz and the AM571x (and AM570x) only have one running at 667 MHz. This code can be kept or removed by the board port. As changes are made, the code must make sure to configure the new board correctly, with the appropriate number of DDR interfaces and speed configuration.

For AM571x and AM570x, make sure to use the code for the AM571x IDK in board/ti/am57xx/board.c to select 1 EMIF:

```
.dmm_lisa_map_3 = 0x80640100,
```

Set Speeds for Other Processing Cores

There are a number of cores on these devices and each has a configurable clock. This clock configuration is done in arch/arm/cpu/armv7/omap5/fdt.c. In that file, there are voltage domain configurations for the DSP, IVA, and GPU cores. Here's an example of the DSP's:

```
u32 dra7_opp_dsp_clk_rates[NUM_OPPS][OPP_DSP_CLK_NUM] = {
    {}, /*OPP_LOW */
    {600000000, 600000000, 400000000}, /* OPP_NOM */
    {700000000, 700000000, 466666667}, /* OPP_OD */
    {750000000, 750000000, 500000000}, /* OPP_HIGH */
};
```

Here there are settings for 3 different performance points (NOM, OD, and HIGH), usually coordinated with the higher speed grade parts. In the case of the "D" speed grade parts, which specify the DSP is supposed to run at 500 MHz and there is not a correctly defined entry. This structure should be modified to either use the OPP_LOW entry, or simply modifiy one of the other entries (ex. OPP_NOM) with the correct entry for 500 MHz operation.

The OPP is chosen via U-Boot build configuration by defining the OPP that U-Boot should use. The configuration file for building U-Boot for the AM57x boards should serve as a great reference. It is located in configs/am57xx_evm_defconfig. A quick review of this file will reveal that the Processor SDK sets these values to OPP_HIGH by default. A board using a different processor that does not support this speed should change this to a more appropriate value for each core by setting the appropriate configuration (CONFIG_DRA7_GPU_OPP_LOW=y, for example).

Linux Changes to Consider

In this case, the Linux board port is managed by the Device Tree Source, or dts file. The dts files provided with the SDK that are used for the EVM use a variety of include (or, dtsi) files that overlay on top of each other to form the complete dts. For example, the am571x-idk.dts includes the dra7.dtsi file. The dra7.dtsi file defines nodes like these to specify the OPPs that will be used by Linux:

```
cpu0_opp_table: opp_table0 {
        compatible = "operating-points-v2";
        opp-shared;

        opp_nom@1000000000 {
            opp-hz = /bits/ 64 <1000000000>;
            opp-microvolt = <1060000 850000 1150000>;
            opp-supported-hw = <0xFF 0x01>;
            opp-suspend;
        };

        opp_od@1176000000 {
            opp-hz = /bits/ 64 <1176000000>;
            opp-microvolt = <1160000 885000 1160000>;
            opp-supported-hw = <0xFF 0x02>;
        };

        opp_high@1500000000 {
            opp-hz = /bits/ 64 <1500000000>;
            opp-microvolt = <1210000 950000 1250000>;
            opp-supported-hw = <0xFF 0x04>;
        };
};
```

These are the appropriate settings for the speed grade of device found on the EVMs. Since they are not correct for a "D" or "J" speed grade parts, the correct value can easily be overlaid using the new board's dts file. The build procedure for a device tree file will use the last value that touches a node. Here's an example of changing OPP_NOM to 500 MHz by adding this entry to the boards dts file (ex. am57xx-idk.dts):

```
&cpu0_opp_table {
        opp_nom@500000000 {
                opp-hz = /bits/ 64 <500000000>;
                opp-microvolt = <1060000 850000 1150000>;
                opp-supported-hw = <0xFF 0x01>;
                opp-suspend;
        };

        /delete-node/ opp_od@1176000000;
```

```
        /delete-node/ opp_high@1500000000;
        /delete-node/ opp_nom@1000000000;
};
```

This will create a new 500 MHz node and delete the other three nodes that were originally created above in dra7.dtsi.

Something similar could be done for a "J" speed grade part that runs at 1 GHz and doesn't support any other OPPs for MPU:

```
&cpu0_opp_table {
        /delete-node/ opp_od@1176000000;
        /delete-node/ opp_high@1500000000;
};
```

Keystone=

{{

1. switchcategory:MultiCore=

- For technical support on MultiCore devices, please post your questions in the C6000 MultiCore Forum
- For questions related to the BIOS MultiCore SDK (MCSDK), please use the BIOS Forum

Please post only comments related to the article **Linux Porting Guide for AM571x/AM570x Speed Grades** here.

- For technical support on MultiCore devices, please post your questions in the C6000 MultiCore Forum
- For questions related to the BIOS MultiCore SDK (MCSDK), please use the BIOS Forum

Please post only comments related to the article **Linux Porting Guide for AM571x/AM570x Speed Grades** here.

C2000=*For technical support on the C2000 please post your questions on The C2000 Forum. Please post only comments about the article **Linux Porting Guide for AM571x/AM570x Speed Grades** here.*

DaVinci=*For technical support on DaVinciplease post your questions on The DaVinci Forum. Please post only comments about the article **Linux Porting Guide for AM571x/AM570x Speed Grades** here.*

MSP430=*For technical support on MSP430 please post your questions on The MSP430 Forum. Please post only comments about the article **Linux Porting Guide for AM571x/AM570x Speed Grades** here.*

OMAP35x=*For technical support on OMAP please post your questions on The OMAP Forum. Please post only comments about the article **Linux Porting Guide for AM571x/AM570x Speed Grades** here.*

OMAPL1=*For technical support on OMAP please post your questions on The OMAP Forum. Please post only comments about the article **Linux Porting Guide for AM571x/AM570x Speed Grades** here.*

MAVRK=*technical on MAVR please po questions MAVRK Forum. F post only commen the article Porting (for AM571x/ Speed G here.*

Links

Amplifiers & Linear
Audio
Broadband RF/IF & Digital Radio
Clocks & Timers
Data Converters

DLP & MEMS
High-Reliability
Interface
Logic
Power Management

Processors

- ARM Processors
- Digital Signal Processors (DSP)
- Microcontrollers (MCU)
- OMAP Applications Processors

Switches & Multiplexers
Temperature Sensors & Control ICs
Wireless Connectivity

Retrieved from "https://processors.wiki.ti.com/index.php?title=Linux_Porting_Guide_for_AM571x/AM570x_Speed_Grades&oldid=225278"

This page was last edited on 7 March 2017, at 10:59.

Content is available under Creative Commons Attribution-ShareAlike unless otherwise noted.