Simple AMP Running Linux and Bare-Metal System on Both Zynq SoC Processors

Based on Xilinx Application note xapp1078 for ZC702 board, adapted for Zedboard by vcesson.

Introduction

The Zynq-7000 AP SoC provides two Cortex-A9 processors that share common memory and peripherals. Asymmetric multiprocessing (AMP) is a mechanism that allows both processors to run their own operating systems or bare-metal applications with the possibility of loosely coupling those applications via shared resources. The reference design includes the hardware and software necessary to build a reference design that runs both Cortex-A9 processors in an AMP configuration. CPU0 runs Linux and CPU1 runs a bare-metal application. Care has been taken to prevent the CPUs from conflicting on shared hardware resources. This document also describes how to create a bootable solution and how to debug both CPUs.

Design Overview

Refer to the document provided by Xilinx; it can be followed step by step, except for small details. Just take note of the changes described after.
The main difference is that ZC702 board has 1GB of DDR while Zedboard has just 512MB. So CPU1 application is mapped at address 0x18000000 instead of 0x30000000. Also, configuration of U-boot and Linux is different, but perfectly supported now.

Implementation details

Note: SDCard-ready files are available at https://github.com/vcesson/xapp1078_z.git

Generating Hardware

XPS may propose to update cores. Updates can be done safely.

When updating the bitstream, the following error may occur:
"A problem was encountered attempting to get the ChipScopePro license."
A solution is to delete every ChipScope related IP from the design. To do so, go to "Bus Interface" tab and delete the three following instances using a right click=>Delete Instance=>Delete instance and all its connections.
chipscope_icon_0
chipscope_ila_0
chipscope_vio_0
ChipScope is not compulsory here.

Generating Applications

Creating Bare-Metal Application For CPU1:
The linker script must be modified to match the Zedboard address space. Open app_cpu1/src/lscript.ld and do the following changes:
ps7_ddr_0_S_AXI_BASEADDR Base Address = 0x18000000, Size = 0x08000000

Creating Linux Kernel

First of all, create U-Boot: refer to "Creating U-Boot" section.
Then follow steps described on this link: http://wiki.xilinx.com/zynq-linux

Creating Linux Device Tree

Use zynq-zed.dts instead of zynq-zc702.dts.
Reduce memory:

```
memory {
            device_type = "memory";
            reg = <0x00000000 0x18000000>;
      };
```

Creating U-Boot

Follow steps described on this link: http://wiki.xilinx.com/zynq-uboot but replace the configuration command by:

```
make zynq_zed_config
```

Once U-Boot is build, go back to Linux creation.

Acquiring Root File System

Generating Boot File

Copying Files to SD Card

Running the Design

The command to wake up the second cpu is:

```
/mnt/rwmem.elf 0xFFFFFFF0 0x18000000
```

Debugging the Design

If the program hangs during linux initialization, there may be a problem with the device tree. For instance, if it seems to hang during i2c initialization, you should remove the related contents from the dts file and regenerate the dtb.