Building the First Stage Bootloader and Vivado projects for the ADMVPX39Z2

Introduction

This guide describes how to build the First Stage Bootloader and Vivado designs for the ADM-VPX3-9Z2.

Requirements

Xilinx SDK 2017.4

Xilinx Vivado 2017.4 with ES parts enabled.

Vivado Projects

The Vivado project contains the processing system configuration, which is required by the First Stage Bootloader.

Sources

The sources can be downloaded from https://support.alpha-data.com

Generating the Vivado project

ES1 parts must be enabled in Vivado's init.tcl file (See Xilinx Answer Record #53090 for info about init.tcl). Add the following line to init.tcl to enable ES1 parts:

```
enable_beta_device*
```

To generate the Vivado project:

1: Open Vivado 2017.4, and in the tcl console navigate to the fpga/vivado/ directory, e.g:

```
cd C:/ZynqMPSoc/fpga/vivado
```

2: Source the tcl script to generate the project:

```
source ./simple-9z2.tcl
```

This will generate the Vvado project in the fpga/vivado/simple-9z2 directory. A bitstream can then be generated in the usual way.

After bistream generation, the hardware description file can be generated for use with XSDK. Click File->Export->Export Hardware, then click Ok.

Outputs

After building has completed, the following files can be found:

admvpx39z2_zu15_simple.bit	This file is the FPGA bitstream.
simple_9z2_wrapper.hdf	This is the hardware description file to be loaded by the First Stage Bootloader.

Customising the Processor Configuration

The processor can be configured to us different peripherals, such as the EMMC or SD card, or different configurations of the GTR transceivers. If any modifications are made to the processing system, the block design should be generated by clicking "Generate Block Design" in the left bar, and then exporting the hdf file again. After this, the First Stage Bootloader should be re-generated with the new hdf file.

First Stage Bootloader

The FSBL (First Stage Boot Loader) configures the processor with the settings from Vivado before loading U-Boot.

Sources

The sources can be downloaded from https://support.alpha-data.com

Building the FSBL

1: Open a command line interface and source the XSDK setup script. e.g for Windows:

```
call C:\Xilinx\SDK\2017.4\settings64.bat
```

2: Change directory to the github repository downloaded above, and start XSDK in batch mode, using build_fsbl_2017_4.tcl

```
xsdk -batch -source build_fsbl_2017_4.tcl
```

This will build all the required projects, and generate the BOOT.bin file required to boot.

The BOOT.bin file will include the bitstream file located in the "bit" folder, and the u-boot.elf file located in xsdk/build_bootimg folder.

Outputs

After building has completed, the following files can be found in the build directory:

9z2_fsbl.elf	This file is the first stage bootloader executable.
BOOT.bin	This is the boot file that the Zynq process will first try to load for booting. It contains first stage bootloader
9z2_workspace	This directory contains the workspace and all required projects to build the 9z2 FSBL.

Customising the FSBL and Processor Configuration

The FSBL can be customised after generating the output files. Open Xilinx SDK 2017.4 and set the workspace to the "9z2_workspace" folder that was created when running the fsbl build script. The project called **9z2_fsbl** will contain the first stage bootloader source files, which can be modified as needed.

The project will use a pre-built hdf file that configures the processor to use DisplayPort, USB and the SD Card. This can be changed by right clicking on the 9z2_hw project, and selecting "Change Hardware Platform Specification", click yes to the warning, navigate to the new hdf file and click OK. This will re-generate the first stage bootloader with the updated processor configuration.

After generating the FSBL, a BOOT.bin file must be generated manually:

1 Within Xilinx SDK, Select Xilinx->Create Boot Image.
2 Select Import from existing BIF file
3 In the "Import BIF file path" section, select "browse" and select to "build_bootimg/mkimg-uboot_bitstream.bif"
4 Click "Create Image"

The updated BOOT.bin file will be genreatred in the xsdk/build_bootimg directory.

Revision History

Revision	Date	Description of Change
1.0	15th June 2018	Initial draft