SHAKTI BOOT SEQUENCE

User Manual

developed by: SHAKTI Development Team @ iitm '19

shakti.org.in

contact @ shakti[dot]iitm[@]gmail[dot]com

Table of Contents

SECTION 1

Introduction

In Stand alone mode, the Aardonyx board runs autonomously. The application is no longer downloaded from the PC through a debugger and run. Instead, it is stored in the Flash memory. When the system starts, the boot loader loads the application from the Flash to the physical memory. Then the control transfers to the application residing in RAM. This mode of running the application is usually used for standalone systems.

This document describes:

- The overall boot process.
- An overview of the boot options available on the SHAKTI Arty35T.
- Bare-metal boot examples that can be run on the SHAKTI Arty 35T.

There are two components to running an application in standalone mode. There should be a boot process, which ensures the application is run after reset i.e. without any external intervention. Secondly, the bare metal application should be present in a non volatile memory. This will ensure the bare metal application will run autonomously even after several resets. The Figure 2 describes the boot process.

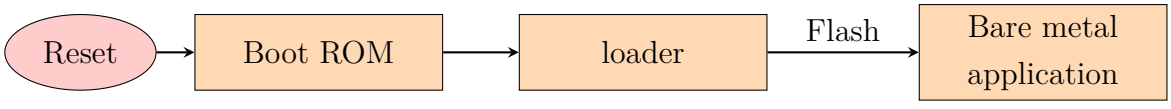

Figure 1: SHAKTI Boot flow

1.0.1 Prerequisite

In order to run the example application in standalone mode, the following are required:

- Digilent Artix 35T board with SHAKTI E class programmed.
- Host PC running Ubuntu 16.04.
- SHAKTI tool chain installed.

SECTION 2

Boot Overview

This section presents an overview of the different boot options and capabilities available for booting applications on SHAKTI for Arty35T boards.
The SHAKTI boot process starts when the processor is released from reset, and jumps to the Boot ROM address space. Typically, the main steps in boot flow are:

- Detect the selected boot source.
- Perform necessary initialization for pre-loader to start.
- Load the bare metal application from Flash to physical memory and jump to it.

The behavior of the Boot ROM is influenced by the pin options.

2.1 Boot Sources

The boot happens in one of the following modes:

- SPI Flash
- Debugger

Boot from SPI
When booting from SPI, the loader images are always located in the Boot ROM. The loader is present following the boot initialization code. The bare metal application is present in the flash memory, inside a 64KB sector. The loader loads the contents in flash from the location 0x000B0000 inside flash memory. Usually the contents inside

the flash is the bare metal application.

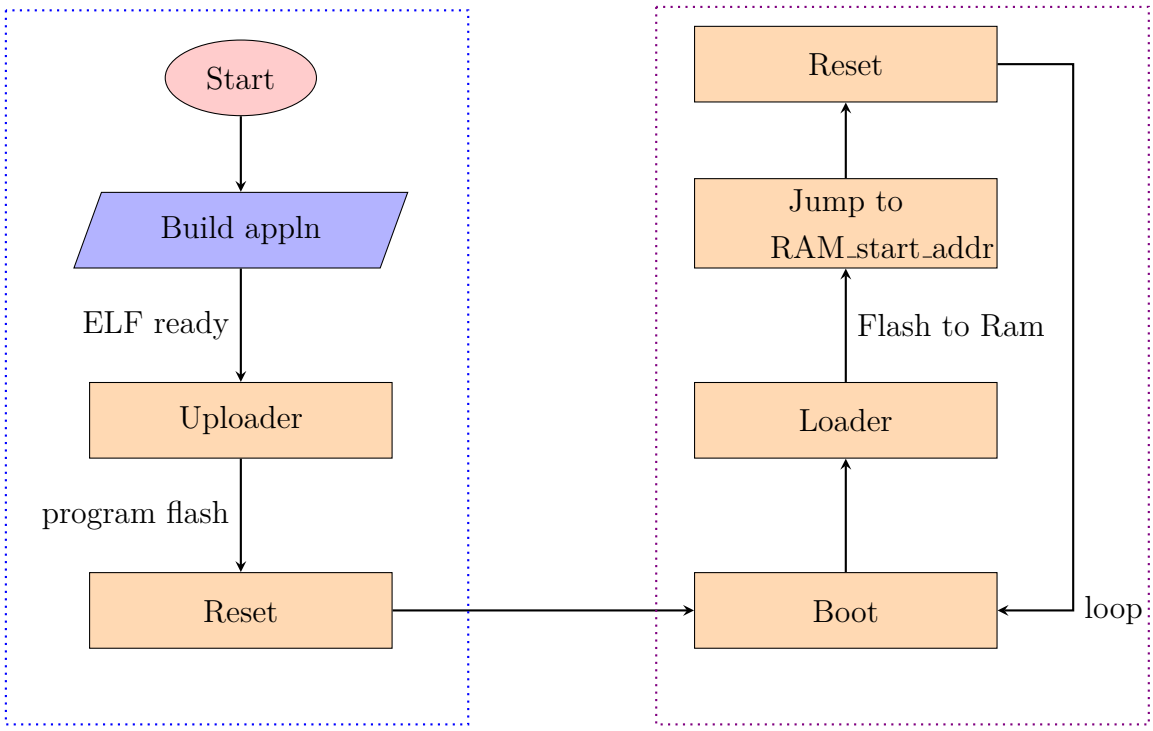

Figure 2: High level working diagram of Standalone mode

Boot Duration
In time critical applications, the duration of the boot process is very critical, and it needs to meet a certain constraint. The SPI boot takes less than 1 sec to boot. And the boot code size is 2048 X 32 bits.

2.2 Steps to generate standalone build

The make upload command is used to build and upload the application to the flash. This enables the application to run in stand alone mode after reset. The shakti-sdk has a Uploader tool to load the content to flash, after building the image.

```
$ cd shakti-sdk
$ make upload PROGRAM= bare metal appln TARGET=artix7_35t
```

Interpreting above commands:

- PROGRAM is the new one created. It is listed by typing ”make list_applns”.
- TARGET= artix7_35t or artix7_100t.
- Default TARGET is artix7_35t.