

GETTING STARTED WITH MACHINEKIT ON SoC+FPGA

HOW TO CREATE AND CUSTOMIZE A WORKING SYSTEM

CHARLES STEINKUEHLER

Agenda

- Quick Start Guide
- What are all the pieces
- Customizing the software
- Customizing the hardware
- Questions

Quick Start Guide

Using a development board and uSD image

- Start with a development board
 - Terasic DE0-Nano-SoC (Cyclone-V)
 - MYIR Z-turn (Zynq)
 - Zedboard MicroZed (Zynq)
- Download and burn an OS image to uSD
 - <http://deb.mah.priv.at/uploads/de0-nano/>
 - <http://deb.mah.priv.at/uploads/zynq/>
- Enjoy!
- More details online (see references)

Component Pieces

- Boot Loader (U-Boot)
- Kernel (Linux)
- OS (Debian Jessie)
- Application (Machinekit)
- FPGA Hardware
- Interface Hardware

Software Customization

- U-Boot:
 - Vendor provided snapshots available
 - Mainline supports Altera and Xilinx parts
- Linux:
 - Mainline supports Altera and Xilinx parts
 - Vendor provided branches available
- OS (Debian):
 - apt get install
 - Tweak image generation scripts (based on debootstrap and RCN's scripts)
- Application (Machinekit)
 - Install from source or packages
 - Edit code (source) or send a PR (packages)

Hardware Customization (FPGA)

The Basics

- The hostmot2 VHDL is very flexible and can be configured at compile time to support nearly endless combinations of features
- Some features are defined by generics passed to the hostmot2 instance
- Card ID and physical details are defined in a VHDL package (card file)
- Logic features and I/O specifics are contained in a separate VHDL package (PIN file)
- The appropriate card and PIN packages are referenced via “use work.<package>.all”
- The HostMot2 component is instantiated and passed details from the selected libraries as generics

Hardware Customization (FPGA)

Making Changes

- Create a new “PIN” package file describing the logic and I/O
 - ModuleID controls what logic gets instantiated
 - Add and remove module types
 - Change the number of instances
 - PinDesc controls how signals are connected to I/O pins
 - Available functions depend on the modules defined by ModuleID
- Use the existing PIN files as examples
- **LOTS** more examples available from Mesanet.com (*.zip files for the various FPGA cards)
- No automated tools exist for generating PIN files
- User is responsible for correctness and consistency

Hardware Customization (Interface Hardware)

- Example Interface Board: DE0-Nano-DB25:
https://github.com/cdsteinkuehler/bobc_hardware/tree/CRAMPS/DE0-Nano_DB25
- Details will depend on the end use application
- Standard schematic/PCB design/layout task

Hardware Customization (New Platform)

- Design the board or obtain details for the new platform (dev-kit)
- Create a new “card” package file describing the hardware
 - Clock Frequencies!
 - Board Name
- Create a new “PIN” package file describing the logic and I/O
- Create a new Quartus/Vivado project using your new packages
- Update the HPS/PS (ARM system) if needed using the FPGA tools

Build configurations used for uSD images

- These are provided for reference, it is recommended you build using a current mainline source tree or a recent vendor branch, however referring to the existing build scripts will likely make it much easier to get started.
- U-Boot:
<https://jenkins.machinekit.io/job/u-boot-socfpga/>
<https://jenkins.machinekit.io/job/u-boot-xilinx/>
- Linux:
<https://jenkins.machinekit.io/job/socfpga-kernel-4.1-ltsi-rt/>
<https://jenkins.machinekit.io/job/xilinx-kernel-4.4-rt/>
- Debian rootfs (contains RIP Machinekit build):
<https://jenkins.machinekit.io/job/oib-cyclonev/>
<https://jenkins.machinekit.io/job/oib-zynq/>
- FPGA bit files:
<https://jenkins.machinekit.io/job/mksocfpga-quartus/>
<https://jenkins.machinekit.io/job/mksocfpga-vivado/>
- FPGA packages:
<https://jenkins.machinekit.io/job/mksocfpga-packaging-quartus/>
<https://jenkins.machinekit.io/job/mksocfpga-packaging-vivado/>
- uSD Image:
<https://jenkins.machinekit.io/job/oib-cyclonev-sd-build/>
<https://jenkins.machinekit.io/job/oib-zynq-sd-build/>

References

- M. Haberler DE0-Nano-SoC instructions
<https://gist.github.com/mhaberler/89a813dc70688e35d8848e8e467a1337>
- Mesa Electronics:
<http://www.mesanet.com/>
- Machinekit build infrastructure:
<http://www.machinekit.io/docs/infra/repositories/>

License

Copyright (C) 2017 Charles Steinkuehler.

Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.3 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts. A copy of the license may be obtained from the Free Software Foundation:

<http://www.gnu.org/licenses/fdl.html>