SDSoC Hardware Platform ARTY-Z7

1.0

Generated by Doxygen 1.8.13

Contents

Chapter 1

Main Page

1.1 Introduction

This HW platform is a basis for building Vivado SDSoC applications running on an Arty-Z7 board connected to a TEC0053 board that implement the Field-Oriented Control (FOC).

It consists of the following:

1. A base Vivado 2017.1 project, containing a block design augmented with input/output points for SDSoC functions.
2. A template SDSoC application implementing FOC which can be used as a starting point for more complex applications; see *FOC SDSoC* for more information.

1.2 Block diagram

The block diagram of the HW platform is shown on Figure 1.1. See the tables 1.1, 1.2, 1.3 and 1.4 for the lists of IP-cores.

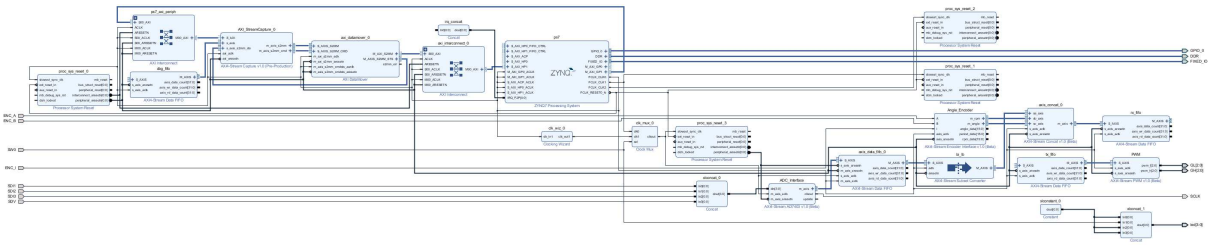

Figure 1.1 Block Diagram of the HW platform.

Table 1.1 List of IP cores on the input path.

Name	Type	Description
xlconcat_0	Standard	Concat input signals from 4 external ADCs to vector
ADC_Interface	Custom	Decode ADC signals and generate AXI4-Stream with Currents and voltage from TEC0053

Name	Type	Description
ADC_data_split	Custom	Split ADC data stream to 2 streams, FOC processing and raw data capture
Angle_Encoder	Custom	Decode signals from rotor angle encoder and generate AXI4-Streams for current angle and rotation speed
axis_concat_0	Standard	Assemble AXI4-Stream with data from ADC and Encoder interfaces
rx_fifo	Standard	Buffer FIFO with data for FOC processing. M_AXIS port of this FIFO is entry point of hardware function pf_read_stream
raw_fifo	Standard	Buffer FIFO with raw data from ADC. M_AXIS port of this FIFO is entry point of hardware function pf_read_adc_stream

Table 1.2 List of IP cores on the output path.

Name	Type	Description
tx_fifo	Standard	Buffer FIFO for FOC processing output stream. S_AXIS port of this FIFO is entry point of hardware function pf_write_stream
PWM	Custom	Genarate bipolar PWM signal for motor control

Table 1.3 List of IP cores on the debug stream path

dbg_fifo	Standard	Buffer FIFO for the debug stream; S_AXIS port of this FIFO is entry point of hardware function dbg_write_stream
AXI DataMover	Standard	Copy data from a stream to main memory through a master AXI Bus interface
AXI StreamCapture	Custom	Capture data from a stream to main memory by using the standard IP core "AXI DataMover"

Table 1.4 List of system IP cores

Name	Type	Description
ps7	Standard	Zynq Processing System block.
proc_sys_reset_0, proc_sys_reset_1, proc_sys_reset↩ _2	Standard	Reset blocks for system clocks

1.3 Interface

The table 1.5 lists HW functions available for defining an SDSoC application by calling them from C code.

Table 1.5 List of HW functions available to an SDSoC application

Function	Description	Data format
pf_read_adc_stream	Read raw data from the ADC stream	16-bit signed samples from 4 channels in parallel
pf_write_stream	Write to the PWM	16-bit signed values for 3 channels in parallel, the fourth channel unused for a total of 64-bits.
pf_read_stream	Read current data and positions	See the table 1.6
dbg_write_stream	Write to the debug stream	64-bit value

Table 1.6 Format of the data read by pf_read_stream()

Bits	Description
63 ... 48	Angle, in encoder steps

Bits	Description
32 ... 47	Speed, in RPM
16 ... 31	Second channel current, I_b, signed integer
0 ... 15	First channel current, I_a, signed integer

1.4 Directory structure

The directory structure is listed in the table 1.7.

Table 1.7 Directory structure

Directory	Description
hw	Hardware part of platform (Vivado project and configuration file)
samples	Example software project
src	Platform functions source files and build configuration
sw	Software project includes, libraries and prebuilt parts

1.5 Tools

The tools required are listed in the table 1.8.

For the development, only Vivado SDSoC is needed.

For the documentation Doxygen is used.

Table 1.8 Tools

Tool	Version	Notes
Vivado SDSoC	2017.1	SDSoC Development Environment
Doxygen	1.8.11	Documentation extraction
MiKTeX	2.9	PDF generation

1.6 Quick start

To create a new SDSoC application based on the FOC template, do as follows:

1. On the Start Menu, locate SDx IDE 2017.1 and click on it. The Vivado SDSoc 2017.1 starts.
2. Select workspace, if necessary.
3. On the main menu, select File → New → Xilinx SDx Project. The New Project wizard dialogue appears.
4. Enter "my_foc" as the project name and click Next. The Choose Hardware Platform page appears.
5. Click Add Custom Platform. A directory selection dialogue appears. Navigate either to the directory "arty_↩ z7_10" or "arty_z7_20" and click OK. The Choose Software Platform and Target CPU page appears.
6. Don't change anything and click Next. The Templates dialogue appears.

7. In the list Available Templates, select Field Oriented Control (see Figure 1.2). Click OK. A new project will be created.
8. Optional: Modify the source code of the template application to match your preferences.
9. On the main menu, select Project → Build Project. It will take about 15 minutes to build the project; the actual time depends on the computer.

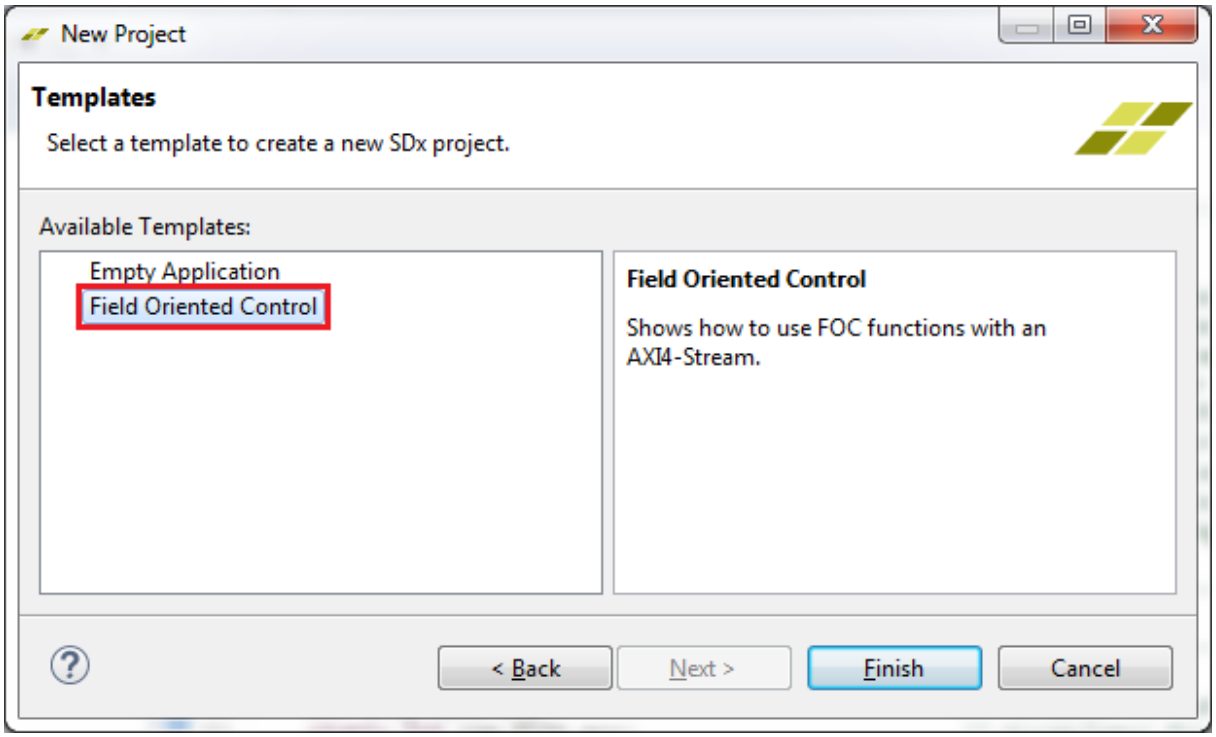

Figure 1.2 Wizard page Available Templates with the sample application Field Oriented Control.

To run the newly built application on the Arty-Z7 board connected to the TEC0053, do as follows:

1. Copy the files from the directory Debug/sd_card to your SD card.
2. Put the card into the Arty-Z7 board.
3. Connect the Arty-Z7 USB port named "PROG/UART" to the computer. If necessary, download and install drivers from the http://www.ftdichip.com/Drivers/VCP.htm .
4. Power up the Arty-Z7 and TEC0053 boards. The FPGA will be configured and Linux will boot.
5. Open the USB serial port with a terminal program. In the case nothing appears on the screen for about 5 seconds, press Enter once to bring up a serial console login.
6. Login with the username "root" and password "root".
7. Enter command "/run/media/mmcblk0p1/my_foc.elf" to run your application.