About this package

This is a package of example codes and projects for the book:

System-on-Chip Design with Arm(R) Cortex(R)-M Processors

Reference Book

by Joseph Yiu, 2019 (first edition), ISBN: 978-1-911531-19-7

Published by Arm Education Media (www.armedumedia.com)

Here you will find:

- An example Cortex-M3 system design based on Arm Cortex-M3 DesignStart Eval.
- Simulation setup for the example system.
- FPGA project setup for the example system, for Digilent Arty-S7-50T FPGA board and Xilinx Vivado 2019.1.

The directory structure is as follows:

Directory	Descriptions
simenv	Simulation environment.
simenv\rtl_sim	Execution directory for simulation.
simenv\testcodes	Testcodes for simulation with compilation setup, supports Arm Compiler 5, Arm Compiler 6 and gcc.
simenv\example_software_compilations	Simplified software compilation setup for Arm Compiler 5, Arm Compiler 6 and gcc. Not used for simulation.
fpga	FPGA project environment.
fpga\Arty_S7_fpga	FPGA project for Arty S7.
fpga\Arty_S7_fpga\cortex_m3_designstart_eval	Placeholder for Cortex-M3 DesignStart Eval (you need to download the Cortex-M3 IP from Arm DesignStart website).
fpga\Arty_S7_fpga\fpga_top	Top-level Verilog files for the FPGA project.
fpga\Arty_S7_fpga\mcu_system	Verilog design for system-level design, bus interconnect components and peripherals.
fpga\Arty_S7_fpga\project_1	Vivado project.
fpga\testcodes_fpga_blinky_keil_mdk	Example blinky project for FPGA with Keil MDK (Book section 11.3.3 – 11.3.9).
fpga\testcodes_fpga_blinky_rtx_keil_mdk	Example blinky project with RTX for FPGA with Keil MDK (Book section 11.4.2).
fpga\testcodes_fpga_demo_keil_mdk	Demo project for FPGA for Keil MDK.
fpga\testcodes_fpga_demo_ac5_ac6_gcc	Demo project for FPGA for compilation with command lines with makefile in a Linux environment.

IMPORTANT: The Cortex-M3 DesignStart Eval (obfuscated version) files are not included.

Before you start, you need to download the Cortex-M3 DesignStart Eval from Arm website. Please visit https://developer.arm.com/ip-products/designstart to register and download the file. After downloading the packages, please copy the files as follows:

Please copy **the following files** from source (in Cortex-M3 DesignStart Eval) to the Destination (in this example package) locations
Copy **cm3_code_mux.v** from AT421-MN-80001-r0p0-02rel0\m3designstart\logical\cortexm3integration_ds\verilog To fpga\Arty_S7_fpga\cortex_m3_designstart_eval\cortexm3integration_ds\verilog
Copy **cortexm3ds_logic.v** from AT421-MN-80001-r0p0-02rel0\m3designstart\logical\cortexm3integration_ds_obs\verilog To fpga\Arty_S7_fpga\cortex_m3_designstart_eval\cortexm3integration_ds_obs\verilog
Copy **CORTEXM3INTEGRATIONDS.v** from AT421-MN-80001-r0p0-02rel0\m3designstart\logical\cortexm3integration_ds_obs\verilog To fpga\Arty_S7_fpga\cortex_m3_designstart_eval\cortexm3integration_ds_obs\verilog

The simulation setup is based on a Linux environment using makefiles. You also need access to Modelsim / Questasim (I have also included experimental support for Icarus Verilog in the makefile, but have not added any interactive simulation support for it.)

To run a simulation, first, you need to configure two makefiles:

Directory	**Descriptions**
simenv\rtl_sim\makefile	Simulator type: SIMULATOR = mti / iverilog
simenv\testcodes\makefile	Toolchain type: TOOL_CHAIN = ac5 / ac6 / gcc Test name: TESTNAME = hello / timer_test / uart_test

Simulation

To compile the RTL:

$> cd simenv/rtl_sim

$> make compile

After compilation, you can run the simulation:

$> make run

The makefile automatically invokes the software compilation makefile to compile the test software.

If the simulation setup works correctly, the simulation will execute the code and terminate the simulation automatically. At the end of the simulation, the software sends a special character 0x04 to the UART that is captured and detected by the UART monitor in the testbench, which then stops the simulation.

FPGA project

The FPGA project is based on the same Cortex-M3 example system, but with some modifications:

- The memory size for program memory (CODE region) is set to 32KB.
- The memory size for data memory (CODE region) is set to 16KB.
- There are additional FPGA I/O for Arty S7's LEDs, switches, buttons. (The System Control register peripheral at memory address 0x40007000 is extended for this).
- Additional clock and reset control (e.g., Clock Wizard generated by Vivado).

The Verilog RTL sources contain a pre-processing macro called FPGA_CONFIG to allow the same source files to be used for simulation and FPGA. However, the design includes a system definition file – there are two copies: one for simulation and one for FPGA:

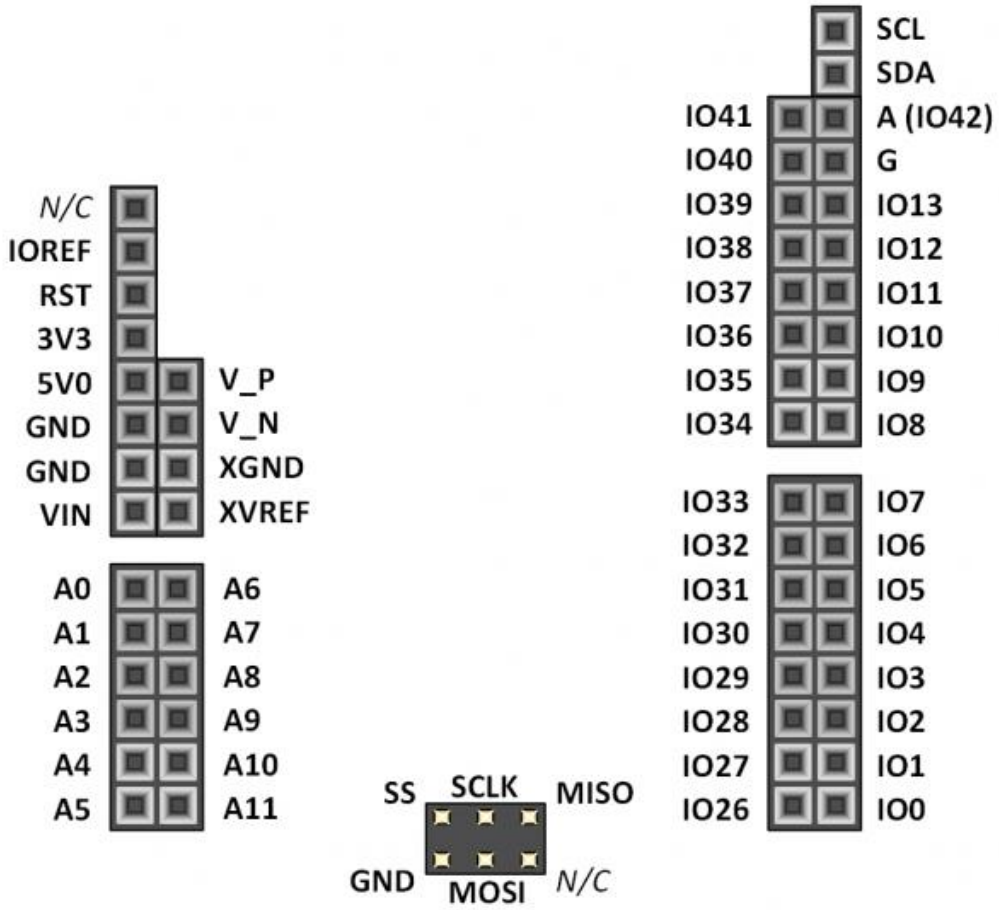
MCU system definition file	**Purpose**
simenv/rtl_sim/mcu_sys_defs.v	For simulation - FPGA_CONFIG not set.
fpga/Arty_S7_fpga/mcu_system/mcu_sys_defs.v	For FPGA - FPGA_CONFIG set.

The debug interface (Serial Wire Debug only) is connected to the shield connector of the Arty S7 and is compatible with the debug connection arrangements in the Arm V2C-DAPLINK.

https://developer.arm.com/tools-and-software/development-boards/designstart-daplink-board

(Details of this board can be found in Arm Cortex-M3 DesignStart FPGA-Xilinx edition User Guide:

https://developer.arm.com/docs/101483/latest/v2c-daplink-board)



(Image from Arty S7 reference manual https://reference.digilentinc.com/reference/programmable-logic/arty-s7/reference-manual)

Shield	FPGA pin	PMOD	V2C-DAPLIN
IO41	U15	jc[0]	SWCLK (SWD debug) input
IO40	V16	jc[1]	SWDIO (SWD data) inout
IO39	U17	jc[2]	auxiliary reset, active low (Not used)
IO38	U18	jc[3]	UART_TX (Not used)
IO37	U16	jc[4]	UART_RX (Not used)
IO36	P13	jc[5]	QSPI_nS (Not used)
IO35	R13	jc[6]	QSPI_CLK clock (Not used)
IO34	V14	jc[7]	DAPLINK board detect, active low (Not used)
IO33	V15	jd[0]	QSPI_Q3 (Not used)
IO32	U12	jd[1]	QSPI_Q2 (Not used)
IO31	V13	jd[2]	QSPI_Q1 (Not used)
IO30	T12	jd[3]	QSPI_Q0 (Not used)
IO29	T13	jd[4]	SD_CLK (SPI clock) (Not used)
IO28	R11	jd[5]	SD_MOSI (SPI MOSI) (Not used)
IO27	T11	jd[6]	SD_MISO (SPI MISO) (Not used)
IO26	U11	jd[7]	SD_nSS (SPI CS) (Not used)

The other shield IOs are not used.

The clocks used in the projects include:

Clocks	Descriptions
CLK12MHz	Not used in the current design.
CLK100MHz	Used by clock wizard.
Clock wizard generated 40MHz clock (internal only, not available at top level)	Used by the processor system.
SWCLK	Serial Wire debug clock, 20MHz.
Slow virtual clock	For setting up timing constraints for slow I/O pins, for FPGA tools only (Not a real clock signal).

The processor reset button is used to generate full reset for the processor system.

Limitations

SWV (Serial Wire Viewer) is not supported on this FPGA image (the V2C-DAPLINK board does not have SWV support). To use the example in Section 11.3.10 “Using ITM for test message output (printf),” you need to use a hardware platform that is capable of handling SWV feature.