ECE4810J
System-on-Chip Design

LAB 0

September 23, 2021

Yihua Liu 518021910998

---

1. (10%) What are the benefits of having Programmable logic on Zynq?

Programmable logic offers an ideal resource for implementing algorithms inherently parallel in nature; for example in signal and image processing, where mathematical operations are performed on a large number of samples or pixels simultaneously. Besides, the Zynq PL has the ability to easily instantiate multiple IP blocks which can act as AXI masters. It can couple with the processor (PS) to allow the differing properties of these two resources to be maximally exploited when partitioning the system into software and hardware elements. It provides a perfect platform in which to create co-processing cores due to the ability to perform parallel execution.

2. (20%) How many cores does Zynq-7000 have? What are the names of those cores? How does the memory hierarchy look like? What is the main CPU frequency?

Zynq-700 has 2 cores. Those cores are ARM Cortex-A9 with NEON and FPU extensions. The memory hierarchy consists of DDR memory, I-cache, D-cache, and processor cache: Dynamic RAM (DRAM), Static RAM (SRAM), Level 1 (L1) Cache, Level 2 (L2) Cache, Level (L3) Cache.

Main CPU frequency for Z-7010, Z-7015, and Z-7020 (the PS on the Artix-7 based devices) is 866 MHz, and 1 GHz for Z-7030, Z-7045, and Z-7100 (the Kintex-based devices) (processor type: ARM Cortex-A9). Arty Z7 integrating ARM Cortex-A9 processor has a CPU frequency of 650 MHz.

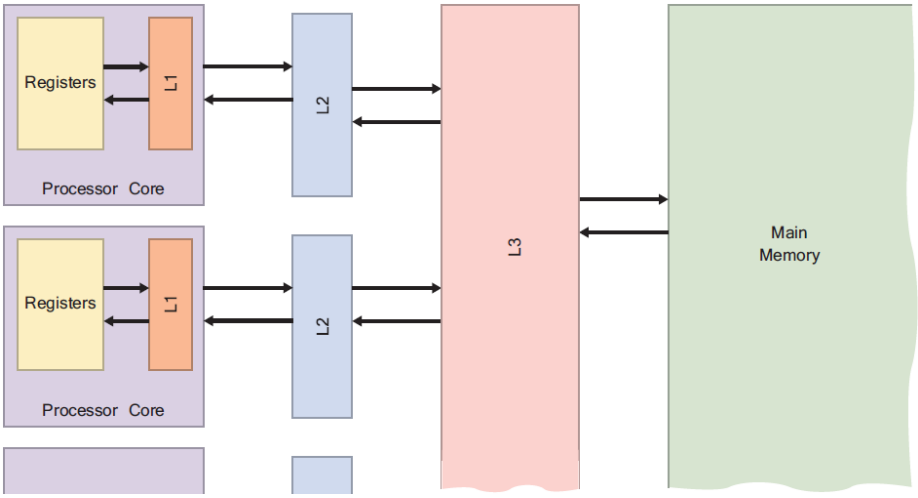

Figure 1: Cache levels and their locations relative to the processor cores and main memory.

3. (15%) Please list the main interfaces of the Zynq-7000 AP SoC device.
PS-PL AXI Interfaces: M_AXI_GP0, M_AXI_GP1, S_AXI_GP0, S_AXI_GP1, S_AXI_ACP, S_AXI_HP0, S_AXI_HP1, S_AXI_HP2, S_AXI_HP3.
PL Co-Processing Interfaces: Accelerator Coherency Port (ACP) Interface.
Interrupt Interfaces.
Memory Interfaces: 1. Dynamic Memory Interface; 2. Static Memory Interface: NAND flash, Parallel (SRAM/NOR).
Programmers model interface to the GIC, Distributor and CPU interfaces.
4. (5%) How is the Arty Z7 board powered?

It is powered from USB or any 7V-15V external power source. It is powered from the Digilent USB-JTAG-UART port (J14) or from some other type of power source such as a battery or external power supply. An external power supply (e.g. wall wart) can be used by plugging it into the power jack (J18) and setting jumper JP5 to "REG". Similar to using an external power supply, a battery can be used to power the Arty Z7 by attaching it to the shield connector and setting jumper JP5 to "REG".

Supply	Circuits	Current (max/typical)
3.3V	FPGA I/O, USB ports, Clocks, Ethernet, SD slot, Flash, HDMI	1.6A/0.1A to 1.5A
1.0V	FPGA, Ethernet Core	2.6A/0.2A to 2.1A
1.5V	DDR3	1.8A/0.1A to 1.2A
1.8V	FPGA Auxiliary, Ethernet I/O, USB Controller	1.8A/0.1A to 0.6A

Table 1. Arty Z7 power supplies.

5. (10%) What is the Quad SPI Flash? What is the usage of it on this board?

Quad SPI Flash, i.e., Quad SPI serial NOR flash, is a Multi-I/O SPI Flash memory that is used to provide non-volatile code and data storage and initialize the PS subsystem as well as configure the PL subsystem.

6. (10%) What is DDR Memory? What is the usage of it?

DDR Memory, i.e., the multi-protocol Double Data Rate memory controller, consists of three major modules: a core memory controller and scheduler (DDRC), an AXI memory port interface (DDRI) and a digital PHY and controller (DDRP). It can be configured to operate in either 16-bit or 32-bit mode, offering access to a 1 GB address space with a single rank DRAM memory configuration of 8-, 16- or 32-bit.

7. (20%) What are the main custom IP blocks creation methods Xilinx provides? Implemented in the PL:

- The Xilinx library
- Third Party sources
- Designed by the User, or by colleagues within the same organisation ('in-house')

IP Core Design Methods/IP Creation Method (tools which enable the creation of custom IP blocks for use in your own embedded system designs):

- HDL (Hardware description languages, such as VHDL and Verilog)
- System Generator
- HDL Coder (MathWorks HDL Coder)
- Vivado High-Level Synthesis: IP-XACT, IP Core, SysGen (Xilinx Vivado HLS)

8. (10%) What is High-Level Synthesis (HLS)? What are the motivations behind HLS?

High-level synthesis means synthesising the high-level C, C++ or SystemC code into an HDL description, which would thereafter undergo logic synthesis to obtain a netlist. Here, 'synthesis' refers to logic synthesis, i.e. the process of analysing and interpreting HDL code and forming a corresponding netlist.

Motivation: 'productivity'

- With a high-level representation abstracting low-level detail, the implication is that the description of the circuit becomes simpler. The result is that designs can be generated much more rapidly than using more traditional methods. Variations on the architecture can be created quickly by applying appropriate directives to the HLS process.
- High-level synthesis from software languages is convenient for many designers, as these languages are widely adopted for developing algorithms and system-level descriptions. The facility to rapidly convert from software to hardware design languages therefore constitutes a simplification of the design process.
- HLS is also beneficial in terms of systems development and software/hardware partitioning, as there is a common language for targeting both elements of the system; this makes it easy to adapt and retarget parts of the design as it is iterated to completion.