Introduction into SoC: Building an FPGA-based System by Integration of MIPSfpga Processor Core with Memory and Peripheral Devices Using AHB-Lite Bus

Dr.T. Paltashev, S.Zhelnio[1]
ITMO University, Laboratory of architectures and design techniques for embedded systems and systems on chip, 14, Birjevaja liniya V.O., Saint Petersburg, 199034, Russia

The solutions that use FPGA-based System on Chip (SoC) as a central element have a number of advantages when compared to the devices based on microcontrollers (AVR, STM, PIC etc) and microcomputers (Raspberry Pi and analogues). Their favorable features are the following:

– scalability and variability: possibility to change the quantity and the composition of the implemented digital interfaces;

– wide range of possibilities in terms of digital signal processing. For instance, appropriate algorithms can be implemented in the HDL;

– possibility of further development as an ASIC solution (when a number of conditions are met).

The emergence of FPGA-chips with a built-in ADC module (for example, Altera MAX 10 series) and FPGA-based budget-friendly debug boards (Terasic DE10-Lite) on the market makes SoC an ever more convenient platform for academic projects.

MIPSfpga [1] CPU core, which could be used as a central element of such SoCs, has the following distinguishing features:

– it is a variation of the industrial core MIPS microAptiv UP specifically oriented to the FPGA-based implementation. It is used by multiple licentiates for the creation of ASIC (such as platform IoT Samsung Artik 1 and Microchip PIC32MZ);

– it corresponds to the MIPS32 architecture and can be used for running Linux;

– it does not use Xilinx- or Altera-specific blocks;

– it provides free license subscription to the Universities and supplies with source codes (Verilog).

To assemble SoC on the basis of MIPSfpga the developer has to provide the integration of the processor core and peripheral devices using the AHB-Lite bus.

An engineer will facilitate his/her job by familiarizing him/herself not only with the CPU core documentation but with the following features:

– minimal list of signals necessary for a correct performance of the bus;

– device connection to the address decoder;

– overall plan of the finite state machine for the external (ported) interface module bus connection;

– operating procedures with bytes and halfwords;

– processing of the Read-after-Write scenario;

– sequence of testing and identification of errors that seldom manifest themselves.

An example of MIPSfpga-based SoC where the issues of integration of the heterogeneous peripheral devices are resolved is the project MIPSfpga-plus [2]. It can also be used as a basis for the development of a more complex system.

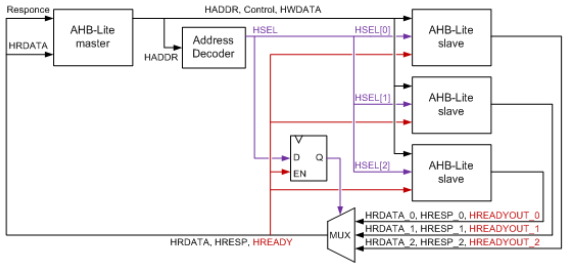

Fig 1: Assembly of peripheral devices with the CPU core using AHB-Lite bus (part of the signals is not shown)

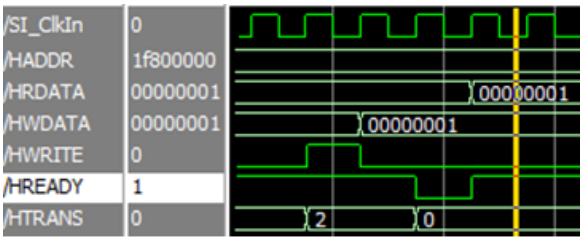

Fig 2: Read-after-Write scenario signal diagram

[1] How to start working with MIPSfpga (Rus) (https://habrahabr.ru/post/275215/)
[2] Github. MIPSfpga-plus (https://github.com/MIPSfpga/mipsfpga-plus/)

[1] Email: zhelniosl@yandex.ru