VLSI SYSTEM DESIGN
(GRADUATE LEVEL)

N26F300

Fall 2022
Rev 1

Project Guidelines

Project Description

*** Please read through the description carefully. Further revision will be posted on Moodle if the description is not clear enough or adjust as needed.

In this term project, you will work as a team (with 4 ~ 6 members) to design an application-specific embedded system that have one pipelined processor, developed or extended the design in this class, can communicate with extended processing unit (EPU) (an accelerator or another processor for a specific application.) via AXI.

Specifically, the chip shall have at least the following IPs: the processor core, L1 caches, instruction memory (IM) & data memory (DM), AXI bus, and an extra processing unit (either application-specific or general-purpose processor with the same class). The main memory (DRAM) and Read-only-memory (ROM) are located outside of the IC via bus. The embedded system starts from a booting ROM followed by the execution of the μ-processor system. All data are assumed residing in main memory first and later are read via interconnection into IM or DM or memory space of the extra processing unit.

The technology specified in class, U18, is provided as the basis of this project. Detailed requirements will be described later.

Detailed requirements are described as followed.

Basic requirements (70%)

- The processor must be a at least 5-stage pipelined structure and compatible to the target processor (RISC-V) supported by the course. Note that it is NOT allowed to use any IP generator for a processor or bus. If found, your project will receive zero point on this part.
    - The processor must be compatible with a compiler and a debugger of the target processor.
- The processor core operating speed is targeting at least 80 MHz for post-synthesized netlist.
- Its data bit width shall match the specification of the target processor

and shall be at least, i.e., 32 bits.

- Its instruction set shall have at least 49 instructions, including branch, I/O instructions.
- The address space for I/O controller registers is part of the same addressing space as data memory address, i.e., I/O registers are memory mapped.
- Some constraints are imposed for on the area of the IPs. The processor core itself shall be within 2 mm^2; IM + DM cannot be larger than 320KB and the silicon area of the CPU+ICache+DCache+IM+DM shall be confined within 110 mm^2 in total. The bus shall be synthesized without area constraints. EPU shall be synthesized and constrained within 3 mm^2. The kernel of the chip shall be less than 120 mm^2.
- Caches, IM or DM shall utilize the SRAM macro generated by Memory Compiler like the one given in class homework. The size of the given macro is 2KB. You can use several of them to construct a bigger one that already detailed in homework. The simulation model will be given.
- All IPs shall use a P&R tool to perform APR, separately and combined together to form a chip. Area shall be obtained by the layout of the P&R tool and NOT the synthesis results.
- The processor core itself must implement and verify L1 Cache along with resolving data hazard by Forwarding. The size of Cache is determined by you. However, you shall avoid using too large cache due to area constraints imposed.
- Must use non-ideal latency models for on-chip and off-chip memories as specified. The on-chip memory is defined as storage modules that are outside the processor itself, such as data memory and instruction memory, but are still inside an IC. Main memory (implemented as DRAM given as one module), on the other hand, is outside the IC and, therefore, considered as off-chip, however, you still need to design a wrapper to connect to the bus.
  - On-chip memory is implemented with given macros. Therefore, the timing will follow the specification of the generated memory.
  - The read/write access time of an off-chip not-synthesized memory, usually DRAM, is 50ns. The main memory only has one read/write port with bit width of 32.
  - If you want to use a custom off-chip memory, such as interleaved or pipelined that closed to commercial products, you shall design your memory wrapper with explanations and shall highlight in

demo presentation and the final presentation.

- The processor must have interrupt mechanism and interrupt service routine for handling requests from other devices, such as sending data and control signals to DMA controller, or receiving data and control signals from DMA controller.
- The processor must provide performance counters, as those specified in homework, inside the processor.
- Must complete code style analysis by superLint and reach at least 99% error free.
- All IPs must perform in either direct test or constrained-random test (CRT). The CPU shall at least have verification schemes specified in all homeworks. At least one bus interface shall be verified by using JasperGold. The more number of IPs adapts CRT, the more points will one obtain. All these shall be specified clearly in both demo and final presentation. The following are the minimum requirements.
    - ■ Subsystem without EPU
        - ◆ Must verify every instruction individually with testbenches, specifically capability of self-verification with error locations and error messages.
        - ◆ Must perform verification schemes in all homeworks specified by TAs
        - ◆ Must verify by running the benchmark which provided by TAs and list the RTL-simulation time in demo check list.
        - ◆ Must verify at least 2 hardware interrupt and interrupt service routine by receiving data from external devices.
    - ■ EPU
        - ◆ Must perform verification for the targeting functionality
            - With at least 10 meaningful and significant benchmarks by using direct test
            - With CRT by using JasperGold. Note that you are required to clearly explain the intention of SVA.
    - ■ Full system including EPU
        - ◆ Must verify by running at least 3 benchmarks whose number of instructions shall be larger than 1000 lines of code w/o space lines and perform a meaningful function based on desired target applications. A meaningful function is like perform NxN matrix multiplication, various NN models with significant amount of inference data, etc. If you target 3D-graph applications, a meaningful function would be to compute

pixel position for a cube when rotating 60 degrees horizontally.

- The Watchdog timer operating speed is targeting at 10 MHz for post-synthesized netlist.

Advanced features (60%)

- Run and pass a set of advanced benchmarks specified by TA later. (+)
- Add, synthesize, and verify at least 10 more instructions other than those in basic requirements and include instructions facilitating 64-bit addition/subtraction & store/load. (+)
- Add, synthesize, and verify direct-memory access (DMA) block. (+)
- Use VIP to verify the synthesized AXI (++) much more than the basics, i.e., those taught in the homework
- Add, synthesize, and verify another levels of Cache, such as L2 or L3. (++)
- Add, synthesize, and verify stack or other mechanisms to facilitate function calls or recursive function. (++)
- Add, synthesize, and verify dynamic branch prediction. (++)
- Validate the full system running an FPGA board after verifying using simulations. (+++++)
- Use CRT to verify for more than two IPs and provide coverage results. (+++)
- The full chip is laid out and verify with I/O PADs with confirmed post-layout simulation. (++)
- Add, synthesize, and verify floating-point co-processor. (++++)
- Make the full system bootable by an operating system, such as Linux, Android, or RTOS. (+++++)
- Other special features are welcomed, but need to discuss with TAs or instructors and fully explain as well as verify with adequate testbenches.