Fall 2024
Rev 3.1

Project Guidelines

Project Description

*** Please read through the description carefully. Further revision will be posted on Moodle if the description is unclear or needs to be adjusted as needed.

In this term project, you will work as a team (with 4 ~ 5 members) to design an application-specific embedded system that has 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 IC 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 outside the IC and can only be accessed via bus. The embedded system starts from a booting ROM and is followed by the execution of the micro-processor system. All data are assumed to reside in the EXTERNAL main memory first and later are read via interconnection into IM or DM or memory space of the extra processing unit.

This project is based on the technology specified in class, ADFP16. Detailed requirements will be described later. BASE Requirement is for the processor-based subsystem. PLUS Requirement is for the EPU. Advanced Requirements will be counted once you complete Requirements BASE and PLUS. The detailed requirements are as follows.

Processor-based Subsystem, BASE Requirement (35%)

- The processor must have at least a 5-stage pipeline structure and be compatible with 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 points on this part.
- The processor must be compatible with the target processor's compiler and debugger.
- The processor core operating speed targets at least 800 MHz for a post-synthesized netlist.

- Its data bit-width shall match the specification of the target processor and shall be at least 32 bits.
- Its instruction set shall have at least 65 instructions, including branch and I/O instructions.
- The address space for I/O controller registers is part of the same addressing space as the data memory address, i.e., I/O registers are memory mapped.
- The AXI operating speed targets at least 400 MHz for a post-synthesized netlist.
- Caches, IM, or DM shall utilize the SRAM macro as specified by TA. The size of the given macro is fixed. You can use several of them to construct a bigger one already detailed in the homework.
- All IPs shall use an APR tool to perform P&R separately and combine them to form a chip with IO pads. The area shall be obtained by the layout with IO Pads using an APR tool, and NOT the synthesis results. Note that the pin count will affect the size of the chip area. Therefore, you must use the minimum number of pins possible. The silicon area will reflect the cost of your design and need to be controlled within **a specified budget** constrained by the project.
- The processor core must implement and verify the various levels of Cache access (at least L1) and resolve any data hazards it may cause. Due to the imposed area constraints, you should avoid using a large cache when determining its size.
- As specified, non-ideal latency designated models must be used to access on-chip and off-chip memories. On-chip memory is a storage module outside the processor, such as data and instruction memory, but it is still inside the chip. On the other hand, the main memory (implemented as DRAM given as one module) is outside the IC and, therefore, considered off-chip. However, designing a wrapper to connect to the bus would be best.
- 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, DRAM, is 25ns. The main memory only has one read/write port with a bit width of 32.
- The watchdog timer (WDT) will operate at the speed of 20 MHz.
- Suppose you use a customized off-chip memory, such as interleaved or

pipelined, close to commercial products. In that case, you should design your memory wrapper with explanations and highlight them in the demo and final presentations.

- The processor must have an interrupt mechanism and interrupt service routine for handling requests from other devices, such as sending data and control signals to or receiving data and control signals from the DMA controller.
- As specified in the homework, the processor must provide performance counters inside the processor.
- Must complete code style analysis by superLint and reach at least 99% error-free.
- All IPs must perform either direct or constrained random tests (CRT). The CPU shall at least have verification schemes specified in all homework. At least one bus interface shall be verified using JasperGold. The more IPs adapt CRT, the more points one obtains. All these shall be specified in the demo and the final presentation.
- Must verify every instruction individually with test benches, specifically the capability of self-verification with error locations and error messages.
- Must perform verification schemes in all homework specified by TAs
- Verify by running the benchmark provided by TAs and listing the RTL simulation time on the demo checklist.
- Verify with at least two hardware interrupts and service routines by receiving data from external devices.
- Need to pass the cross-domain clocking test using the EDA tool as you were taught in the class.

EPU, PLUS Requirement (35%)

- The difficulty level of EPU will set the tone of your project. Therefore, choose your application wisely. The lower challenging levels will receive lower ranks on starting points. Suppose you design a half-tone processing unit with 64 x 64 pixels; it is a simple case. The maximum score you may obtain is about 10 points. This is because an undergraduate student can finish it with trained skills in Verilog design as one homework.
- An EPU must verify the targeting functionality against golden models developed in C/C++/Java/Python or MATLAB.

- With at least ten meaningful benchmarks by using direct tests.
  - With CRT by using JasperGold. Note that you are required to explain the intention of SVA clearly.
- The system, including EPU, must verify by running at least seven benchmarks whose number of instructions shall be more significant than 1000 lines of code w/o space lines per benchmark and perform a meaningful function based on the desired target application.
  - A meaningful function is like performing *N* x *N* matrix multiplication, various neural network models with a significant amount of inference data, etc. If you target 3D-graph applications, a meaningful function would be to compute the Pixel position for a cube when rotating 60 degrees horizontally.
- Some constraints are imposed on the chip area. The total area, including the IO pads (around 208 pins), shall be within 1.75 mm^2. The budget cost is about NT$ 3,298,750, based on TSRI information. Any price above the budget shall be justified, and the cost incurred shall be calculated based on the TSRI formula and announced separately. If not justified, i.e., consider wasted in the silicone space. Your rank will be reduced.

Advanced features (30% maximum)

Before considering points of the advanced features, you must complete the BASE and the PLUS requirements. For example, if you did not complete Requirement BASE, additional points will not be given to the "ten more instructions" add-on item in the advanced features. All features added shall also be verifiable by running the system together, not just individually run by itself. Failure to follow these rules will result in no credit on this part.

- Run and pass a set of advanced benchmarks specified by TA later. (+)
- Add, synthesize, and verify at least ten more instructions other than those in BASE 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 completed in the homework
- Add, synthesize, and verify Caches after L1, such as L2 or L3. (++)
- Add, synthesize, and verify stack or other mechanisms to facilitate function calls or recursive functions. (++)

- Add, synthesize, and verify dynamic branch prediction. (++)
  - Validate the full system running an FPGA board after verifying using simulations. (+++++)
    - Use CRT to verify two IPs and provide coverage results. (+++)
- The full chip is laid out and verified with I/O PADs with confirmed post-layout simulation. (+)
- Add, synthesize, and verify floating-point co-processors. (++++)
- Make the full system bootable by an operating system, such as Linux, Android, or RTOS. (+++++)
- Other special features are welcomed but must be discussed with TAs or instructors, fully explained, and verified with adequate test benches.