- ## Developer's instructions

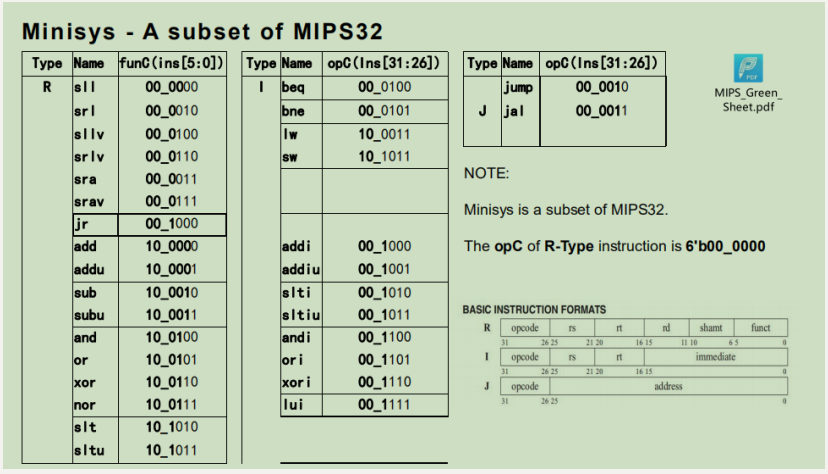
NAME	SID	CONTRIBUTION RATIO	WORK
Ruixiang Jiang	12111611	33%	ALU, IFetch, CPUTop, Led, Switch, Seg, Report
Yujing Zhang	12111944	33%	
Yilun Qiu	12013006	33%	

Version modification record

- v1.0(05-14): Basic modules completed
- v1.1(05-20): Top module completed

CPU architecture design specification

- CPU Features
  - Instruction set architecture
    Registers: number = 32, width = 32
    Exception handling:



  - Address space design
    - Architecture: Harvard architecture
    - Addressing unit: Byte

        - Instruction space: $0x00000000$ ~ $0xFFFFFFFF$
        - Data space: $0x00000000$ ~ $0xFFFFFC00$
    - External I/O devices: none
    - CPI: single cycle CPU
- CPU interface
    - Clock: Built-in clock interface of EGO1 development board
    - Reset: R1 of EGO1 development board
    - Uart:
    - Switch
    - 16-bit-width Led
    - 8-bit-width Seg
- Internal structure of CPU
    - External interfaces of CPU
      (add pictures)
    - Interface Connections among submodules within CPU
      (add pictures)
    - Submodules
        - Instruction Fetch
          The instruction fetch stage involves retrieving the instruction from the memory. The program counter (PC) is responsible for providing the memory address of the current instruction. This address is sent to the instruction memory, which fetches the instruction stored at that address and transfers it to the instruction register (IR). Additionally, the program counter is incremented to point to the subsequent instruction in memory.
          (add pictures)

- Instruction Decode
  During the instruction decode stage, the received instruction is analyzed and decoded. The control unit examines the opcode of the instruction and determines the required control signals for subsequent stages. By interpreting the opcode, the control unit configures the CPU components accordingly, enabling the appropriate data paths and control signals necessary for executing the instruction.
  (add pictures)
- ALU
  The ALU carries out arithmetic and logical operations on the data within the CPU. Depending on the specific instruction, the ALU performs operations such as addition, subtraction, logical AND, logical OR, or other specified computations. The ALU takes inputs from the general-purpose registers and applies the operation indicated by the control signals received from the control unit.
  (add pictures)
- Controller
  The controller is responsible for generating the required control signals to coordinate the activities of the CPU components. It examines the decoded instruction and produces control signals to enable or disable specific registers, select the appropriate source and destination registers, activate the ALU, and govern memory operations. The controller plays a crucial role in orchestrating the overall execution of the instruction.
  (add pictures)

- Data memory
  The data memory stage involves accessing the memory unit for read or write operations. For load or store instructions, the memory address for the data transfer is calculated and stored in the memory address register (MAR). Subsequently, the data memory unit reads data from the specified address (in the case of a load operation) or writes data to the specified address (in the case of a store operation). The data is temporarily stored in the data memory register (MDR) before further processing.
  (add pictures)
- MemOrIO
  During the MemOrIO stage, the CPU interacts with external memory or I/O devices to either read data from memory or write data to memory or I/O devices. This stage involves transferring data between the CPU and the memory or I/O devices, as well as handling any necessary address calculations or data transfers.
  It contains the following operations:
    - Memory Read: If the instruction requires reading data from memory, the memory address calculated in earlier stages (typically stored in the Memory Address Register, MAR) is used to fetch the data from the memory. The fetched data is then temporarily stored in the Memory Data Register (MDR) within the CPU.
    - Memory Write: If the instruction involves writing data to memory, the memory address and data to be written (typically stored in the MAR

and MDR, respectively) are transferred from the CPU to the memory. The memory then stores the data at the specified memory address.

- I/O Operations: In some cases, the MemOrIO stage can involve input/output operations instead of or in addition to memory operations. This includes communication with peripheral devices such as keyboards, displays, or storage devices. The CPU may send or receive data to or from these devices during this stage.

The specific operations and data transfers during the MemOrIO stage depend on the instruction being executed and the specific design of the CPU. The MemOrIO stage is an integral part of the single-cycle CPU architecture, ensuring that memory and I/O operations are properly handled within the CPU's instruction execution process.
(add pictures)

- Switch Driver
  The dip switches on the EGO1 development board are physical switches that can be toggled on or off to provide input signals to the board. The switch driver module would typically include the necessary circuitry and logic to read the state of the dip switches and provide the corresponding digital signals to the other components or modules on the board.

The switch driver module enables the EGO1 board to read the positions of the dip switches and use that information for various purposes, such as what we need to do in the basic test.
(add pictures)

- LED Driver
  The LED driver is an essential module responsible for controlling the LEDs (Light Emitting Diodes) on the board. Receiving input signals from various sources, such as the microcontroller or other modules on the board, it provides the necessary circuitry and logic to control the illumination of the LEDs based on the desired patterns or states, typically including the LED patterns, sequences, or behaviors.

  With the LED driver module, the EGO1 development board can effectively control the illumination and behavior of the LEDs, providing visual feedback or indicators for various purposes, such as status indication, user interaction, or debugging information.
  (add pictures)

- Segment Driver
  The segment driver is responsible for controlling the segments or tubes used to display numbers or alphanumeric characters. It takes input signals, such as digital data representing the number or character to be displayed, and generates the appropriate signals to activate the specific segments required to form the desired pattern.

  By controlling the activation and deactivation of the segments, the seg driver module enables the EGO1 board to display numbers or characters on the 8-segment

display. It can be programmed or configured to update the display in real-time, showing dynamic information, or to show static values based on the input provided.

Special attention should be paid to the value of the clock cycle for the 8-segment display on the EGO1 development board.
(add pictures)

Test instructions

- Test for vivado

METHOD	TYPE	DETAIL	RESULT
Simulation	Unit	Test the 5 basic modules on OJ	Accepted

- Test for mips

Issues and summary