MEDGAR EVERS COLLEGE
The City University of New York

Project:
Processor Program

Directions:

For your computer program, you will need to create a processor which is the processing unit of your computer. A processor has a control unit for fetching and decoding instructions, an arithmetic logic unit (ALU) for executing arithmetic and logic operations, registers, an internal bus for transferring content between the units of the processor and an external bus for transferring content to modules external to the processor. For your processor all content will be in boolean, which means you will be performing bitwise operations. Likewise, its units (methods) will all be void and take no parameters unless specified otherwise. They will receive inputs and produce outputs through the registers and internal bus. Furthermore, other than bool objects, int variables can be included for counters and indices only.

You will be creating a **Processor** class that

- □ must have a private **Bus** field for the internal bus.
- □ must have a private **Bus** pointer field for the external bus.
- □ must have private bool arrays for registers.
- □ must have a private **Logger** pointer field for logs.
- □ must have a private method named `Control()`. It must fetch instructions, decode them, and delegate control to other methods to perform executions. For simple assigns to registers, it can perform those tasks itself, but you can delegate it to another method. Ultimately, it should modify the buses, the program counter and call methods.
- □ must have a private method named `ALU()`. It must perform all arithmetic and logic operations. It should know which operations to perform based on the control bus of the internal bus. You can delegate certain operations to other methods.
- □ must have a default constructor that assigns NULL to the **Bus** pointer object (external bus) and zeroes to all the elements of the registers.
- □ must make the copy constructor and assignment operator private. As before, only their prototypes need to be declared.
- □ may have overloaded constructors that assigns the **Bus** object (external bus) and the **Logger**.
- □ must have a public set methods for the **Bus** object (external).
- □ must have a public set methods for the **Logger** object.
- □ must have a private method named `Valid()`. It must assign the control bus of the internal bus a success signal that indicates that the processor can process if a **Bus** object is assigned to the external bus; otherwise, it assigns a failure signal that indicates that the processor cannot process. The success signal should be all zeroes.
- □ must have a public method named `Process()`. It calls `Valid()`; and then, calls `Control()` if the success signal is read. This means it interprets the control bus of the internal bus to determine if a signal is generated

When creating your class, you need to add documentation that states what the failure signal is (the documentation may be a comment in the header file). If a **Logger** object is not assigned, the processor should still be able to operate; however, you would not know what it is doing because it does (must) not produce displays. If the **Logger** object is included, a log should be produce any time the processor modifies a register (only include final modifications of the register; that is, if a register is an operand of an operation, its final result should be logged, but not the intermediate changes). You can also include logs of the control bus of the internal bus for debugging purposes.

Last, besides creating the class, you must write a test program that creates a **Bus** object, a **Processor** object, a **Memory** object and a **Logger** object. Next, it should perform at least two arithmetic and logic operations each. Additionally, provide the memory file that is being modified by the program and the log file.