ISA:

The processor uses 16-bit instruction format and includes eight general-purpose registers, each 16 bits wide. No status registers, such as condition flags, are utilized in this architecture. It features separate instruction memory (IMEM) and data memory (DMEM) to handle program and data storage independently. The data memory (DMEM) functions as a queue: all LOAD instructions retrieve data from the front (head) of the queue, while STORE instructions append data to the back (tail).

The supported instruction set comprises eight operations:
LOAD (fetch data from the front of DMEM), STORE (write data to the back of DMEM), ADD (register-to-register addition), ADDI (addition with an immediate value), MUL (register-to-register multiplication), SUB (register-to-register subtraction), SLL (shift left logical), and JRI (jump to an address calculated by adding a register value and an immediate offset).

LW

0000	Ra	

Load value from top of dmem into reg A.

SW:

0001	Ra	

Store value from reg A to back of dmem.

ADD:

0010	Ra	Rb	Rc	

Addition is done to the contents of reg B and reg C and result is stored in reg A

SUB:

0011	Ra	Rb	Rc	

Contents of reg C is subtracted from contents of reg B and result is stored in reg A
[Choose values where borrow is not required]

MUL:

0100	Ra	Rb	Rc	

Multiplication is done to the contents of reg B and reg C and result is stored in reg A
[only 16 bits of result is stored at destination]

ADDI:

0101	Ra	Rb	6 bit immediate

Add the content of reg B with imm and store the result in reg A.

SLL:

0110	Ra	Rb	Rc

Shift the contents of reg B by the amount specified in the least 4 significant bits of reg C and store the result in reg A.

JRI:

0111	Ra	9-bit immediate

Jump to the location [update the value of PC] given by reg A + Imm *2

NOTE : A testbench containing a set of test cases will be provided to assist in verifying and validating your implementation.