MIPS Simulator

Description: You will develop an instruction-by-instruction simulator for the MIPS architecture. The simulator will execute instructions sequentially (non-pipelined) and output the contents of all registers and memory (the state of the processor and memory) after each instruction. You do not have to implement exception/interrupt handling.

Your simulator must support the following MIPS instructions. Refer to the Appendix in the textbook for the exact semantics of each instruction.

```
J, BEQ,
ADD, ADDI, SUB
SW, LW
SLL, SRL
MUL,
AND, OR,
NOP
```

Input/Output: Input to your program is going to be a valid assembly program in MIPS. The input program can be written by hand or generated by a compiler. In either case, it will contain only the instructions listed above.

The final output of your simulator will be the contents of the register file and memory. These two components represent the *state of the machine* after completing the execution of the program. The simulator should also support a debug mode that prints the contents of the register file and simulator after executing each instruction. You are free to choose the exact formatting of this output but you should make any attempt to make it intuitive.