We have implemented six command-

1. add- It is in R-format with opcode=”000000” funct=”100000” shamt=”00000”. Destination address in rd, first and second operant in rs and and rt respectively.
   reg(rd) = reg(rs) + reg(rt)
2. sub- opcode=”000000” funct=”100010” in R-format with shamt=”00000”.
   reg(rd) = reg(rs) - reg(rt)
3. sll- Again in R-formate. Shamt stores shift amount in binary. Opcode=”000000” funct=”000000”. rs}=”00000”.
   reg(rd) = shift_left [reg(rt)] , shamt
4. srl- Same as of sll with funct=”000010”
   reg(rd) = shift_right [reg(rt)] , shamt
5. lw- I-format with opcode=”100011” offset value is stored in address in binary.
   reg(rt) = mem(reg(rs) + offset)
6. sw- I-format with opcode=”101011” as in lw offset is stored in address in binary.
   mem(reg(rs) + offset) = reg(rt)

We read file at specified location to load the program in the memory which is array (0 to 4095) of std_logic_vector (31 downto 0).
We stop execution when we read “00000000000000000000000000000000”.
In order to read file we created a impure function which takes file name in string value as an argument. Open a file and until it is read completely, we read it line by line and each line give std_logic_vector which is stored in memory. We have a internal data structure for register which is array (0 to 31) of std_logic_vector (31 downto 0).