1. Describe the input fields of each pipeline register

1 IF/ID pipeline register
- 1.1 32 bits for PC+4
- 1.2 32 bits for current instruction

2 ID/EX pipeline register
- 2.1 32 bits for RSdata
- 2.2 32 bits for RTdata
- 2.3 21 bits for last 21 bits of current instruction
- 2.4 32 bits for PC+4
- 2.5 1 bit for RegWrite control line
- 2.6 1 bit for RegDst control line
- 2.7 1 bit for ALUsrc control line
- 2.8 1 bit for Branch control line
- 2.9 1 bit for MemRead control line
- 2.10 1 bit for MemWrite control line
- 2.11 1 bit for MemtoReg control line
- 2.12 2 bits for ALUOp control line
- 2.13 32 bits for SignExtendData

3 EX/MEM register
- 3.1 32 bits for ALUresult
- 3.2 32 bits for pc of branch instruction
- 3.3 5 bits for number of write register
- 3.4 32 bits for RTdata
- 3.5 1 bit for RegWrite control line
- 3.6 1 bit for Branch control line
- 3.7 1 bit for MemRead control line
- 3.8 1 bit for MemWrite control line
- 3.9 1 bit for MemtoReg control line
- 3.10 1 bit for zero output from ALU

4 MEM/WB register
- 4.1 32 bits for MemReadData
- 4.2 32 bits for ALUresult
- 4.3 5 bits for number of write register
- 4.4 1 bit for MemtoReg control line
- 4.5 1 bit for RegWrite control line

2. Explain your control signals in the sixth cycle(both test data test_1.txt and test_2.txt are needed)

1 test_1.txt
- 1.1 IF stage (slt $6, $2, $1)
  - 1.1.1 Branch (from instruction in MEM stage): 0 (not branch instruction)
- 1.2 ID stage (or $5, $1, $0)
  - 1.2.1 RegWrite (from instruction in WB stage): 1 (can write to register)
- 1.3 EX stage (and $3, $5, $6)
  - 1.3.1 RegDst: 1 (RD)
  - 1.3.2 ALUsrc: 0 (RTdata)
  - 1.3.3 ALUOp: 00 (R type)
- 1.4 MEM stage (addi $4, $0, 14)
  - 1.4.1 MemRead: 0 (can't read from data memory)
  - 1.4.2 MemWrite: 0 (can't write to data memory)
- 1.5 WB stage (addi $2, $0, 17)
  - 1.5.1 MemtoReg: 0 (ALUresult)

2 test_2.txt
- 2.1 IF stage (sw $3, 8($3))
  - 2.1.1 Branch (from instruction in MEM stage): 0 (not branch instruction)
- 2.2 ID stage (sw $2, 4($0))
  - 2.2.1 RegWrite (from instruction in WB stage): 1 (can write to register)
- 2.3 EX stage (addi $5, $0, 10)
  - 2.3.1 RegDst: 0 (RT)
  - 2.3.2 ALUsrc: 1 (SignExtendData)
  - 2.3.3 ALUOp: 01 (add)
- 2.4 MEM stage (addi $4, $0, 9)
  - 2.4.1 MemRead: 0 (can't read from data memory)
  - 2.4.2 MemWrite: 0 (can't write to data memory)
- 2.5 WB stage (addi $3, $0, 8)
  - 2.5.1 MemtoReg: 0 (ALUresult)

test_1.txt

010011 00000 00001 0000000000011111 -> addi $1, $0, 31

010011 00000 00010 0000000000010001 -> addi $2, $0, 17

010011 00000 00100 0000000000001110 -> addi $4, $0, 14

000000 00101 00110 00011 00000 011111 -> and $3, $5, $6

000000 00001 00000 00101 00000 101111 -> or $5, $1, $0

000000 00010 00001 00110 00000 010100 -> slt $6, $2, $1

test_2.txt

010011 00000 00010 0000000000000111 -> addi $2, $0, 7

010011 00000 00011 0000000000001000 -> addi $3, $0, 8

010011 00000 00100 0000000000001001 -> addi $4, $0, 9

010011 00000 00101 0000000000001010 -> addi $5, $0, 10

101000 00000 00010 0000000000000100 -> sw $2, 4($0)

101000 00011 00011 0000000000001000 -> sw $3, 8($3)