Team members: Sylvia Le, Irakli Svanidze , Maylin Martinez

**a) Explain the function of the data path. How does it work?**

- A bunch of registers connect to ALU, over which the data flow. Basic operation:
  + Selecting one or two registers.
  + Having the ALU operate on them (for example, adding them together).
  + Storing the result back in some register.

**b) What do the following register names stand for? SP, TOS, LV, CPP, and H.**

- SP: Stack pointer register.
- TOS: Top-of-stack register.
- LV: Local variable register.
- CPP: Constant Pool Pointer.
- H: holding register.

**c) Is it possible to add the contents of the registers SP and LV in one clock cycle?**

- No, because it takes one cycle to put SP or LV on to H.

**d) How many cycles does it take to add the content of H and OPC and write back to OPC?**

- According to the timing graph, it's possible to do it within one cycle. The output value will be written back to OPC near the end of the cycle, at the rising edge of the clock.

**e) How many bits does each register hold (except MBR)? How about MBR?**

- 32-bit for all registers other than MBR. MBR has 8 bits.

**f) Can multiple registers drive the B bus? How/Why?**

- No, might cause physical damage. Imagine register A is having the 1-bit value 1, register B is having 0, if both manage to drive B bus at the same time, how does it know which value to choose?

**g) Can the data on the C bus be written into multiple registers in the same clock cycle?**

- Yes. On the rising edge of the clock near the end of a cycle, registers (notice the s) will be loaded from C bus and memory.

**h) What do the N and Z outputs of the ALU represent?**

- N and Z are ALU status flags.
  + N is set when the result of the operation was Negative.
  + Z is set when the result of the operation was Zero.

**i) The ALU doesn't have a subtractor, how does it do subtraction (e.g. B-A)?**

- Suppose the number representation used is the two complement's, the ALU will flip the sign of A to -A, then perform addition between B and -A.

**j) What are the two functions of the shifter?**

- The basic function of shifters is to double or divide numbers by 2.
- SLL8 shifts the content left by 1 byte, filling the 8 least significant bits with 0.
- SRA1 shifts the contents right by 1 bit, leaving the most significant bit unchanged.

**k) Explain how the MDR and MAR registers work as a pair.**

- MAR/MDR combination is used to read and write ISA-level data.
- MAR contains address of word (word 0, word 1, . . . ) to be read into MDR.

**l) Explain how the MBR and PC registers work as a pair.**

- PC/MBR combination is used to read executable ISA-level program, which consist of a byte stream.
- PC contains addresses of byte (byte 0, byte 1, . . . ) to be read into low-order 8 bits of MBR.

**m) Explain the process of sign extension while MBR is being copied onto the B bus.**

- Data read from memory through the 8-bit memory port are returned in MBR, an 8-bit register. MBR can be gated (i.e., copied) onto the B bus in one of two ways: unsigned or signed.

  + If unsigned value is needed:

    - 32-bit word put onto the B bus contains: the MBR value in the low-order 8 bits, and zero in the upper 24 bits.
    - Unsigned are useful for indexing (table), or when 16-bit int has to be assembled from 2 consecutive unsigned byte in the instruction stream.

  + The other option for converting the 8-bit MBR to a 32-bit word is to treat it as a signed int.

    - Between -128 and +127.
    - Use this value to generate a 32-bit word, with same numerical value.
    - Duplicate the MBR sign bit into the upper 24-bit position of the B bus (sign extension).
    - Rest are for actual MBR bits.

**n) Describe the functions of MPC and MIR.**

- The control store needs its own MAR and MDR, since it's functionally a read-only memory. They have special name:

  + Control store's memory address register: MPC (MicroProgram Counter).

+ Control store’s memory data register: MIR (MicroInstruction Register).

- Hold the current microinstruction, whose bits drive the control signals that operate the data path.
- The MIR contains same 6 groups in the format discussed in Fig. 4-5 in the book.

**o) What is the function of the 'next address' field in the MIR instruction format?**

- Hold the address of the next microinstruction to carry out. Note that this might not be the actual instruction that gets to be executed next, since the JAM field has to be inspected.

**p) Explain the function of the JAM bits.**

- JAM bits determines how the next microinstruction is selected.
- When it’s time to determine what is the next instruction.

  + The NEXT_ADDRESS is copied to the MPC.

  + JAM is then inspected.

  + The value of JAM will determine what is the final next instruction.

  - If JAM = 000, then the next instruction to be executed is located at NEXT_ADDRESS.
  - If any bit of JAM is 1

    > JAMN=1: 1-bit N flip-flop is ORed into high-order bit of MPC.

    > JAMZ=1: 1-bit Z flip-flop is ORed into high-order bit of MPC.

    > If both JAMN and JAMZ is set, both bits are ORed there.

    > MPC[8] := (JAMZ AND Z) OR (JAMN AND N) OR NEXT ADDRESS[8].

**q) Explain the function of WRITE, READ and FETCH bits.**

- These are the three bits in Mem field of the microinstruction format, specifies the memory operation.

  + WRITE: write to memory, via MAR, MDR.

  + READ: read what is in the memory, via MAR/MDR.

  + FETCH: fetch the data from the memory, via PC/MBR.