Tutorial 4: Computer Architecture

December 8, 2019

1. Given the Figure 1 below.

   (a) Bus is a physical connection to transfer data inside computer systems.

   (b) Multipoint bus

   (c) Answer

      i. **Data bus:** Transports data between the memory CPU, memory, and input and output.

      ii. **Address bus:** Specifies data receipient, and identifies source and destination of data on data bus.

      iii. **Control bus:** Provide control for synchronization & control of bus & modules connected.

2.

Point-to-point bus	Multipoint bus
One sender, one receipient	Multiple sender, multiple receipient
Data is sent directly to the receipient	Data is broadcasted to all possible receipient

   (a) Diff 1: Bus carry from specific to specific. Multipoint: Bus connect several devices together

   (b) Bus lines: Data bus & control bus. Data bus, ADD bus, CTRL bus

   (c) Diagram: CRL U - AL

      i. CMP - CMP - CMP - CMP

   (d) Diagram

3. Answer

   (a) When attempting to retrieve data, MAR holds the address of the data to be used by CPU

   (b) MDR will retrieve and store a copy of data pointed by MAR inside the RAM for read access

   (c) MDR is responsible for storing data back into the RAM if write operation is carried out by the CPU

4. Answer

(a) Answer

i. IR stores the current instruction being executed by the CPU.

ii. PC stores the address of the next instruction to be executed by the CPU.

iii. A: A register used for holding data for processing by the CPU and data transfer

(b) Answer

i. They form the machine cycle inside the LMC. The LMC's fetch operation involves the MAR and MDR. Whereas the LMC's execute operation involves the IR, PC, and A.

(c) The ADD operation by the LMC, which involves the full fetch-execute cycle

i. The CPU copies the address pointed by the PC into the MAR to retrieve the next instruction.

ii. The MDR hold a copy of the data pointed to by the MAR for read-access.

iii. The IR retrieves the instruction stored in the MDR.

iv. The CPU decodes the instruction, and place the operand's address inside the MAR for copying purpose. A copy of the operand is returned to the MDR.

v. The CPU adds the adds the accumulator, A, and the MDR together, and stores the end result in the accumulator, A.

vi. The program counter is incremented to the next instruction.

5. Answer

(a) $2^{36} = 6.87194767 \times 10^{10} bytes$

6. Answer **(ASK Teacher, is this hex or decimal?)**

(a) (Table not acceptable in final exam, must list down all the steps)

i.

Instruction	IR	PC	MAR	MDR	A
20 (LOAD)	550	21	50	422	422
21 (ADD)	151	22	51	008	430
22 (STORE) Check with teacher	350	23	50	430	430

ii. Steps:

A.

Steps	Result
550 (Load)	
$PC \rightarrow MAR$	$MAR = 20$
$MDR \rightarrow IR$	$IR = 550$
$IR\,[address] \rightarrow MAR$	$MAR = 50$
$MDR \rightarrow A$	$A = 422$
$PC + 1 \rightarrow PC$	$PC = 21$
151 (MUL)	
$PC \rightarrow MAR$	$MAR = 21$
$MDR \rightarrow IR$	$IR = 151$
$IR\,[address] \rightarrow MAR$	$MAR = 51$
$A \cdot MDR \rightarrow A$	$A = 23_{16} \cdot 5_{16} = 430$
$PC + 1 \rightarrow PC$	$PC = 22$
350 (STORE)	
$PC \rightarrow MAR$	MAR=22
$MDR \rightarrow IR$	$IR = 350$
$IR\,[Address] \rightarrow MAR$	$MAR = 50$
$A \rightarrow MDR$	$MDR = 430$
$PC + 1 \rightarrow PC$	$PC = 23$

7. Answer

(a)

Instruction	IR	PC	MAR	MDR	A
20 (LOAD)	550	21	50	23_{16}	23_{16}
21 (MUL)	151	22	51	5_{16}	$23_{16} * 5_{16} = AF_{16}$
22 (STORE)	350	1	50	$23_{16} * 5_{16} = AF_{16}$	$23_{16} * 5_{16} = AF_{16}$

(b) Steps:

Steps	Result
550 (Load)	
$PC \rightarrow MAR$	$MAR = 20$
$MDR \rightarrow IR$	$IR = 550$
$IR\,[address] \rightarrow MAR$	$MAR = 50$
$MDR \rightarrow A$	$A = 23_{16}$
$PC + 1 \rightarrow PC$	$PC = 21$
$PC \rightarrow MAR$	$MAR = 21$
$MDR \rightarrow IR$	$IR = 151$
$IR\,[address] \rightarrow MAR$	$MAR = 51$
$A \cdot MDR \rightarrow A$	$A = 23_{16} \cdot 5_{16} = AF_{16}$
$PC + 1 \rightarrow PC$	$PC = 22$
Continue	
$PC \rightarrow MAR$	MAR=22
$MDR \rightarrow IR$	$IR = 350$
...	

8. Answer
    (a) RISC (Reduced instruction set computer)
    (b) Advantages of arthictecture (4 is enough, 1 extra):
        i. **Limited and simple instruction set.** Execute at a faster clock speed, does not require complex hardware.
        ii. **Registers-oriented instructions.** Reduces memory access, use registers to operate/hold frequently used instruction.
        iii. **Fixed length/format instruction word.** Easy to identify, can be fetched and decodd independently -> pipelining.
        iv. **Limited addressing mode.** Provide single address mode, speed up instruction executions.
        v. **Large bank of registers.** Registers applied widely, reduce memory access.
9. Extra exercise
    (a) PC -> MAR: 40
    (b) MDR -> IR: 160
    (c) IR[address] -> MAR: 60
    (d) A: 150 + 25 = 175
    (e) PC = 41
    (f) A = 175
    (g) MDr =
10.
    (a) PC: 68
    (b) MAR: 90
    (c) MDR: 333
    (d) IR: 390
    (e) A: 333
    (f) TraceLog
        i. First part
            A. PC to MAR: MAR = 65
            B. MDR to IR = 590
            C. IR [address] to MAR. MAR = 90
            D. MDR to A. A = 111
            E. PC + 1 = PC. PC = 66
        ii. Second part
            A. PC to MAR. MAR = 66

B. MDR to IR. IR = 192
C. IR[address] to MAR. MAR = 92
D. A + MDR = A. A = 333
E. PC + 1 = PC. PC = 67

iii. Third part
A. PC to MAR. MAR = 67
B. MDR to IR. IR = 390
C. IR[address] to MAR. MAR = 90
D. A to MDR. MDR = 333
E. PC + 1 -> PC. PC = 68

1 Notes

1. For LMC remember:
   (a) MAR
   (b) MDR
   (c) PC
   (d) IR
   (e) A → results
2. FETCH (Must have this one first, then follow by one of those below)
   (a) PC to MAR
   (b) MDR to IR
   (c) IR[address] to MAR
3. LOAD
   (a) MDR -> A
   (b) PC = PC + 1
4. Calculation
   (a) MDR -> A
   (b) PC + 1 -> PC
5. Store
   (a) A -> MDR
   (b) PC + 1 -> PC