CSE301 – Computer Organization

Tutorial 2

Chapter 3: A Top-Level View of Computer Function and Interconnection

External Problem

A hypothetical computer has the following characteristics: The processor contains a single general-purpose register, called an accumulator (AC), a Program Counter (PC), a Memory Address Register (MAR), and a Memory Buffer Register (MBR). Both instructions and data are 16 bits long, so memory words are 16 bits.
The instruction format provides 4 bits for the opcode and 12 bits for the memory address.

Opcodes and Operations

Opcode (binary)	Operation
0000	Load AC from memory
0001	Store AC into memory
0010	Add to AC from memory, result in AC
0011	Subtract memory from AC, result in AC
0100	Multiply AC times memory, result in AC
0101	Divide AC by memory, result in AC
0110	Logical AND memory and AC, result in AC
0111	Logical OR memory and AC, result in AC

Memory Contents

Address	Contents
ABD	0E2E
ABE	3E2F
ABF	4E30
AC0	2E31
AC1	1E31
E2E	0009
E2F	0007
E30	0005
E31	0003

Questions

1. What is the maximum memory address space this processor can access?
2. Fill in the contents of every register and memory location **after the fetch cycle and after the execute cycle** of every instruction (values in hexadecimal).

Instruction	cycle	PC	MAR	MBR	AC	loc E2E	loc E2F	loc E30	loc E31
	Initially	0ABD	ABCD	F43A	50C8	0009	0007	0005	0003

3. Rewrite the program stored in memory using **MIPS architecture**.
4. Assume an **interrupt occurs** during execution of the instruction at location ABF. Describe the sequence of events to handle it.
5. How many **memory accesses** are needed to run the 5-instruction program?

CSE321a – Midterm Exam– Fall 2014

Consider a small hypothetical computer with four 16-bit general-purpose registers numbered from 0 to 3. Each machine contains 16 bits (X15-0). The six most-significant bits of the instruction (X15-10) represent an op-code. The following two bits (X9-8) represent a register number. The remaining bits (X7-0) may represent the value or the address of an operand. The following table contains some of the supported op-codes:

Mnemonic	Binary Meaning
LOAD 011100	Load register X9-8 from memory location X7-0.
STORE 011101	Store value of register X9-8 to memory location X7-0.
ADDLD 110001	Add value of memory location X7-0 to register X9-8.
ADDST 110011	Add value of register X9-8 to memory location X7-0.
DECBRNZ 111010	Decrement value of register X9-8 by 1, and if new value of register X9-8 is not 0, branch to instruction whose address is X7-0, else continue normally.

1. How many memory accesses are needed to fetch and execute instruction EA9B?
   (a) 0
   (b) 1
   (c) 2
   (d) 3
   (e) None of the above

2. Suppose the values of register 2 and location 3A are: 2D15 and 11B5 respectively. What would be their values after executing instruction 763A?
   (a) 2D15 and 11B5
   (b) 2D15 and 2D15
   (c) 11B5 and 11B5
   (d) 11B5 and 2D15
   (e) None of the above

3. Suppose the values of register 1 and location 75 are: 623E and 2935 respectively. What would be their values after executing instruction C575?
   (a) 623E and 9174
   (b) 623E and 2935
   (c) 9174 and 2935
   (d) 8B74 and 623E
   (e) None of the above

4. Suppose the values of register 3 and program counter (PC) are: 0001 and 005B respectively. Which of the following instructions will load the PC with 008F after being executed?
   (a) EB8F
   (b) 738F
   (c) EAF8
   (d) C78F
   (e) None of the above

Problem 3.14

A microprocessor has an increment memory direct instruction, which adds 1 to the value in a memory location. The instruction has five stages: fetch opcode (four bus clock cycles), fetch operand address (three cycles), fetch operand (three cycles), add 1 to operand (three cycles), and store operand (three cycles).

a. By what amount (in percent) will the duration of the instruction increase if we have to insert two bus wait states in each memory read and memory write operation?
b. Repeat assuming that the increment operation takes 13 cycles instead of 3 cycles.

Problem 3.18

The microprocessor of Problem 3.14 initiates the fetch operand stage of the increment memory direct instruction at the same time that a keyboard actives an interrupt request line. After how long does the processor enter the interrupt processing cycle? Assume a bus clocking rate of 10 MHz.

Interrupt handling

A given processor has eight interrupt lines (numbered 0-7), and a policy that low-numbered interrupts have priority over higher-numbered ones. The processor takes 10 units of time to handle an interrupt. Suppose the processor starts with no interrupts pending, and the following sequence of interrupts occurs:

Interrupt Number	2	5	1	4	0
Occurrence Time	3	7	11	13	25

Part 2: Questions with Answers

Chapter 3: A Top-Level View of Computer Function and Interconnection

External Problem

A hypothetical computer has the following characteristics: The processor contains a single general-purpose register, called an accumulator (AC), a Program Counter (PC), a Memory Address Register (MAR), and a Memory Buffer Register (MBR). Both instructions and data are 16 bits long, so memory words are 16 bits.
The instruction format provides 4 bits for the opcode and 12 bits for the memory address.

Opcodes and Operations

Opcode (binary)	Operation
0000	Load AC from memory
0001	Store AC into memory
0010	Add to AC from memory, result in AC
0011	Subtract memory from AC, result in AC
0100	Multiply AC times memory, result in AC
0101	Divide AC by memory, result in AC
0110	Logical AND memory and AC, result in AC
0111	Logical OR memory and AC, result in AC

Memory Contents

Address	Contents
ABD	0E2E
ABE	3E2F
ABF	4E30
AC0	2E31
AC1	1E31
E2E	0009
E2F	0007
E30	0005
E31	0003

Questions

1. What is the maximum memory address space this processor can access?
2. Fill in the contents of every register and memory location **after the fetch cycle and after the execute cycle** of every instruction (values in hexadecimal).

Instruction	cycle	PC	MAR	MBR	AC	loc E2E	loc E2F	loc E30	loc E31
	Initially	0ABD	ABCD	F43A	50C8	0009	0007	0005	0003

3. Rewrite the program stored in memory using **MIPS architecture**.
4. Assume an **interrupt occurs** during execution of the instruction at location ABF. Describe the sequence of events to handle it.
5. How many **memory accesses** are needed to run the 5-instruction program?

Solution

(1) As the instruction 16 bit (4bit opcode + 12 bit memory address), the max addressable space 2^{12} = 4KiB(4096 bytes)

(2)

Instruction	Cycle	PC	MAR	MBR	IR	AC	loc E2E	loc E2F	loc E30	loc E31
--	Initially	0ABD	ABCD	F43A	--	50C8	0009	0007	0005	0003
0E2E: load Ac <- loc 0E2E	Inst. Fetch	**0ABE**	**0ABD**	**0E2E**	**0E2E**	50C8	0009	0007	0005	0003
	Oper. Fetch	0ABE	**0E2E**	**0009**	0E2E	50C8	0009	0007	0005	0003
	Inst. Execute	0ABE	0E2E	0009	0E2E	**0009**	0009	0007	0005	0003
3E2F: Sub A, Ac <- Ac - A	Inst. Fetch	**0ABF**	**0ABE**	**3E2F**	**3E2F**	0009	0009	0007	0005	0003
	Oper. Fetch	0ABF	**0E2F**	**0007**	3E2F	0009	0009	0007	0005	0003
	Inst. Execute	0ABF	0E2F	0007	3E2F	**0002**	0009	0007	0005	0003
4E30: Mult A, Ac <- Ac * A	Inst. Fetch	**0AC0**	**0ABF**	**4E30**	**4E30**	0002	0009	0007	0005	0003
	Oper. Fetch	0AC0	**0E30**	**0005**	4E30	0002	0009	0007	0005	0003
	Inst. Execute	0AC0	0E30	0005	4E30	**000A**	0009	0007	0005	0003
2E31: Add A, Ac <- Ac + A	Inst. Fetch	**0AC1**	**0AC0**	**2E31**	**2E31**	000A	0009	0007	0005	0003

	Oper. Fetch	0AC1	**0E31**	**0003**	2E31	000A	0009	0007	0005	0003
	Inst. Execute	0AC1	0AC0	2E31	2E31	**000D**	0009	0007	0005	0003
1E31: Store A, A <- Ac	Inst. Fetch	**0AC2**	**0AC1**	**1E31**	**1E31**	000D	0009	0007	0005	0003
	Inst. Execute	0AC2	**0E31**	**000D**	1E31	000D	0009	0007	0005	**000D**

(3)

```
# MIPS program:
# a0 → accumulator (AC)
# a1 → temporary operand

.data
   E2E: .word 9
   E2F: .word 7
   E30: .word 5
   E31: .word 3

.text
   lw $a0, E2E           # AC       ← MEM[E2E]
   lw $a1, E2F           # temp     ← Mem[E2F]
   sub $a0, $a0, $a1     # AC       ← AC - Mem[E2F]
   lw $a1, E30           # temp     ← Mem[E30]
   mult $a0, $a1         # (HI,LO)  ← AC * Mem[E30]
   mflo $a0              # Ac       ← LO
   lw $a1, E31           # temp     ← MEM[E31]
   add $a0, $a0, $a1     # AC       ← AC + MEM[E31]
   sw $a0, E31           # MEM[E31] ← AC
```

(4) Sequence will be:

1. After the instruction at location ABF finishes execution, CPU checks interrupts.
2. CPU saves current contents of PC, which are AC0, and other relevant data.
3. CPU loads PC with the address of the first instruction of the interrupt service routine (ISR).
4. ISR executes.
5. CPU restores PC (AC0) and relevant data.

(5) Every instruction needs one memory access to fetch and one memory access to fetch or store the operand ➔ Total number of memory accesses = 2 + 2 + 2 + 2 + 2 = 10 memory access.

CSE321a – Midterm Exam– Fall 2014

Consider a small hypothetical computer with four 16-bit general-purpose registers numbered from 0 to 3. Each machine contains 16 bits (X15-0). The six most-significant bits of the instruction (X15-10) represent an op-code. The following two bits (X9-8) represent a register number. The remaining bits (X7-0) may represent the value or the address of an operand. The following table contains some of the supported op-codes:

Mnemonic	Binary Meaning
LOAD 011100	Load register X9-8 from memory location X7-0.
STORE 011101	Store value of register X9-8 to memory location X7-0.
ADDLD 110001	Add value of memory location X7-0 to register X9-8.
ADDST 110011	Add value of register X9-8 to memory location X7-0.

DECBRNZ 111010	Decrement value of register X9-8 by 1, and if new value of register X9-8 is not 0, branch to instruction whose address is X7-0, else continue normally.

1. How many memory accesses are needed to fetch and execute instruction EA9B?
   (a) 0
   (b) **1** ✅
   (c) 2
   (d) 3
   (e) None of the above

💡 *Explaination*:
Instruction fetch takes one clock cycle
Instruction execute EA9B = 1110 1010 1001 1011, op code = 111010 DECBRNZ, either register value is zero or not no need for extra memory access.

1. Suppose the values of register 2 and location 3A are: 2D15 and 11B5 respectively. What would be their values after executing instruction 763A?
   (a) 2D15 and 11B5
   (b) **2D15 and 2D15** ✅
   (c) 11B5 and 11B5
   (d) 11B5 and 2D15
   (e) None of the above

💡 *Explaination*:
763A = 0111 0110 0011 1010, opcode = 011101 STORE the value of register 2 into MEM[3A]
reg $2 = 2D15
MEM[3A] = 2D15

3. Suppose the values of register 1 and location 75 are: 623E and 2935 respectively. What would be their values after executing instruction C575?
   (a) 623E and 9174
   (b) 623E and 2935
   (c) 9174 and 2935
   (d) 8B74 and 623E
   (e) **None of the above** ✅

💡 *Explaination*:
C575 = 1100 0101 0111 0101, opcode = 110001 ADDLD $1, 75, $1 = $1 + MEM[75]
reg $1 = 8B73
MEM[75] = 2935

4. Suppose the values of register 3 and program counter (PC) are: 0001 and 005B respectively. Which of the following instructions will load the PC with 008F after being executed?
   (a) EB8F
   (b) 738F
   (c) EAF8
   (d) C78F
   (e) **None of the above** ✅

💡 *Explaination*: examine the instructions
EB8F: 1110 1011 1000 1111 - DECBRNZ $3, 8F
738F: 0111 0011 1000 1111 - LOAD $3, 8F
EAF8: 1110 1010 1111 1000 - DECBRNZ $2, F8
C78F: 1100 0111 1000 1111 - ADDLD $3, 8F

none of the above instruction and without even examining the instruction the value of register 3 is 1 if decreased by one it cannot branch.

Problem 3.14

A microprocessor has an increment memory direct instruction, which adds 1 to the value in a memory location. The instruction has five stages: fetch opcode (four bus clock cycles), fetch operand address (three cycles), fetch operand (three cycles), add 1 to operand (three cycles), and store operand (three cycles).

a. By what amount (in percent) will the duration of the instruction increase if we have to insert two bus wait states in each memory read and memory write operation?
b. Repeat assuming that the increment operation takes 13 cycles instead of 3 cycles.

Stage	Description	Cycles
1	Fetch opcode	4
2	Fetch operand address	3
3	Fetch operand	3
4	Add 1 to operand	3 *(or 13 in part b)*
5	Store operand	3

Total memory accesses = 4 (three reads + one write).
Each memory access incurs **2 additional wait states**.

(a)

Without wait states
T1 = 4 + 3 + 3 + 3 + 3 = 16 cycles

With wait states
Extra = 4 memory accesses × 2 wait states = 8 cycles
T2 = 8 + 16 cycles

Instruction duration increases by = (T2 - T1) / T1 × 100 = 8 / 16 × 100 = 50%

(b) Without wait states
T1 = 4 + 3 + 3 + 13 + 3 = 26 cycles

With wait states
Extra = 4 memory accesses × 2 wait states = 8 cycles
T2 = 8 + 26 cycles

Instruction duration increases by = (T2 - T1) / T1 × 100 = 8 / 26 × 100 = 30.77%

Problem 3.18

The microprocessor of Problem 3.14 initiates the fetch operand stage of the increment memory direct instruction at the same time that a keyboard actives an interrupt request line. After how long does the processor enter the interrupt processing cycle? Assume a bus clocking rate of 10 MHz.

The interrupt line is checked **only after** the current instruction finishes execution.
When the interrupt occurs, the processor has just started the **fetch operand** stage, so it must complete the remaining stages first:

Remaining Stage	Cycles
Fetch operand	3
Add 1 to operand	3
Store operand	3
Total	**9 cycles**

Bus clock frequency: **10 MHz**

$$T = \frac{1}{f} = \frac{1}{10 \text{ MHz}} = 0.1 \ \mu\text{s} = 100 \ \text{ns}$$

$$\text{Delay} = 9 \times 100 \ \text{ns} = 900 \ \text{ns} = 0.9 \ \mu\text{s}$$

Interrupt handling

A given processor has eight interrupt lines (numbered 0-7), and a policy that low-numbered interrupts have priority over higher-numbered ones. The processor takes 10 units of time to handle an interrupt. Suppose the processor starts with no interrupts pending, and the following sequence of interrupts occurs:

Interrupt Number	2	5	1	4	0
Occurrence Time	3	7	11	13	25

Solution

- disable interrupt method

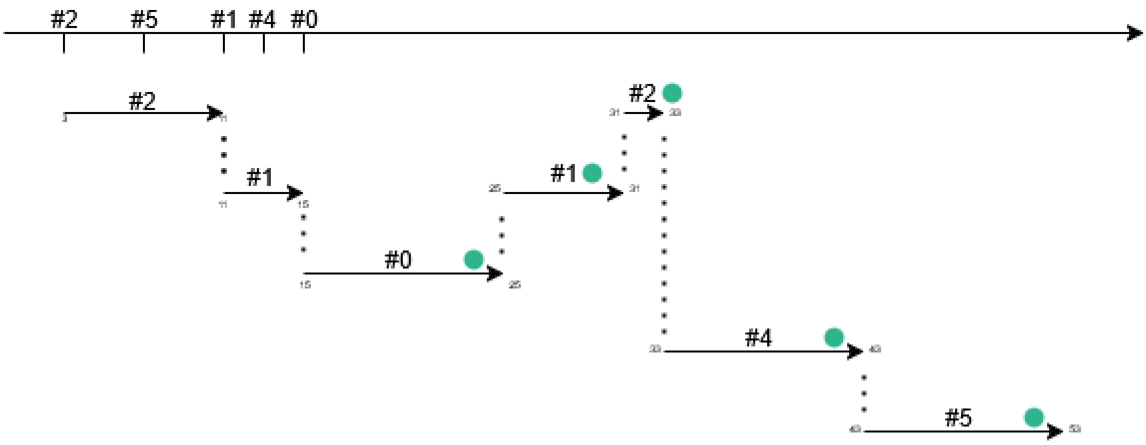
Service is non-preemptive and handled in arrival order:

Interrupt	Arrival	Start	Finish
2	3	3	13
5	7	13	23
1	11	23	33
4	13	33	43
0	15	43	53

- priority method

Service is preemptive, allow higher-priority (lower number) interrupts to preempt:

Interrupt	Arrival	Start	Finish
2	3	3	33
5	7	43	53
1	11	11	31
4	13	33	43
0	15	15	25


#2
#5
#1
#4
#0
#2
#1
#0
#1
#2
#4
#5