Institute of Technology of Cambodia

Telecommunication and Network

Computer Architecture
Group 10

TP 04

Name	ID	Score
Neath Morokot	e20220575	…….
Chhit Chantola	e20221273	…….
Kith Sereyvibol	e20221328	…….
An Vanneath	e20220208	…….

Lecture: Course: Mr.CHUN Thavorac
TP: Mr.OL Phearun

Academic Year: 2023-2024

Question:

1. What is the main function of CPU?
2. What does the control unit do?
3. What purpose does a datapath serve?
4. Compare CISC machine to RISC machine.
5. Explain the steps of fetch-decode-execute cycle.
6. Explain the difference between register-to-register, register-to-memory, and memory-to-memory instructions.
7. What is a big and little endian?

Exercise:

1.How many bits would you need to address a 3M×32 memory if

a. The memory is byte-addressable?
b. The memory is word-addressable?

2. Write the following code segment in MARIE assembly language:

```
if X = Y then
      X := Y x 2;
else if X > Y then
      X := X – 2;
else
    Y := X -Y;
```

(https://marie.js.org/?addition)

3.Assume you have a machine that uses 32-bit integers and you are storing the hex value 1234 at address 0:

a. Show how this is stored on a big endian machine.
b. Show how this is stored on a little endian machine.
c. If you wanted to increase the hex value to 123456, which byte assignment would be more efficient, big or little endian? Explain your answer.

4.The memory unit of a computer has 256K words of 32 bits each. The computer has an instruction format with 4 fields: an opcode field; a mode field to specify 1 of 7 addressing modes; a register address field to specify 1 of 60 registers; and a memory address field. Assume an instruction is 32 bits long. Answer the following:

a. How large must the mode field be?
b. How large must the register field be?
c. How large must the address field be?
d. How large is the opcode field?

Answer1:

1.The main function of CPU is to executes the instructions and delivers the results to the associated output.

2. Control unit determines the sequence of operations, directs the flow of data, and ensures proper coordination among different components.

3. The purpose of a datapath is to performs data processing operations on input data using arithmetic logic unit, buses, multiplexers, and registers.

4. Compare CISC machine to RISC machine:

CISC (Complex Instruction Set Computer) architectures feature a large set of complex, variable-length instructions that can execute multiple operations per instruction, leading to compact code but requiring more complex hardware and higher cycles per instruction (CPI). They are ideal for general-purpose computing with extensive backward compatibility, exemplified by the x86 architecture.

RISC (Reduced Instruction Set Computer) architectures, in contrast, use a smaller set of simple, fixed-length instructions that typically execute in a single clock cycle, resulting in simpler hardware, lower CPI, and efficient pipelining. This makes RISC well-suited for high-performance and embedded systems, with ARM being a notable example. RISC tends to produce larger code size but achieves higher performance and energy efficiency.

5. Explain the steps of fetch-decode-execute cycle:
_ **Fetch**: This is the first step where the CPU 'fetches' an instruction from the primary memory (RAM).
_ **Decode**: Right after fetching the instruction, the CPU 'decodes' it or translates it into a series of actions it can understand.
_ **Execute**: Finally, the CPU 'executes' the instruction decoded in the previous step. It carries out the actions dictated by the instruction.

6. Explain the difference between register-to-register, register-to-memory, and memory-to-memory instructions:

1. **Register-to-Register Instructions**
   Definition: These instructions operate exclusively on CPU registers. Both source and destination operands are located in the registers.
   Performance: Generally the fastest type of instruction because accessing registers is much quicker than accessing memory.

2. **Register-to-Memory Instructions**
   Definition: These instructions involve at least one operand in a register and at least one operand in memory. The operation might read from a memory location, perform the operation with a register, and then store the result back in a register or memory.

Performance: Slower than register-to-register instructions due to the need to access memory, which takes more time compared to register access.

3. Mmory-to-Memory Instructions

Definition: These instructions operate directly on memory locations, with both the source and destination operands being in memory.

Performance: Typically the slowest type of instruction because it involves multiple memory accesses, which are significantly slower than register accesses.

7. A big-endian system stores the most significant byte of a word at the smallest memory address and the least significant byte at the largest. A little-endian system, in contrast, stores the least-significant byte at the smallest address.

Answer2:

1. How many bits would you need to address a 3M×32 memory if

a) The memory is byte-addressable?

$\frac{32}{8}=4\ bits$

$3\times2^{20}\times2^{2}=3\times2^{22}=\log_2 3+22\ bits$

Therefore, you need $\log_2 3+22\ bits$.

b) The memory is word-addressable?

$3\times 2^{20}=\log_2 3+20\ bits$

Therefore, you need $\log_2 3+20\ bits$.

2. Write the following code segment in MARIE assembly language:

If X = Y then

X := Y x 2;

else if X > Y then

X := X – 2;

else

Y := X -Y;

```
If,      Load X
         Subt Y
         Skipcond 400 //Skip (Jump ElseIf) if (AC = 0)
         Jump ElseIf //Jump to (ElseIf) if (X-Y != 0)
Then,    Load Y //Calculation of If
         Add Y
         Store Y
         Jump End //Jump to output

ElseIf, Load X
         Subt Y
         Skipcond 800 //Skip (Jump Else) if AC > 0
         Jump Else
ThenIf, Load X //Calculation of ElseIf
         Subt Minus
         Store X
         Jump End //Jump to output
```

```

//Else condition calculation
Else,    Load X
         Subt Y
         Store X
         Jump End //Jump to output

End,     Output
Halt
X, DEC 0 /Any number
Y, DEC 0 /Any number
Minus, DEC 2
```

3. Assume you have a machine that uses 32-bit integers and you are storing the hex value 1234 at address 0:
a) Show how this is stored on a big endian machine.

0× 00000000	**12**
0× 00000001	**34**

b) Show how this is stored on a little endian machine.

0× 00000000	**34**
0× 00000001	**12**

c) If you wanted to increase the hex value to 123456, which byte assignment would be more efficient, big or little endian? Explain your answer.

• Big endian

0× 00000000	**12**
0× 00000001	**34**
0× 00000002	**56**

• Little endian

0× 00000000	**56**
0× 00000000	**34**
0× 00000000	**12**

• I will use Little endian assignment because it might be able to help finding the designated number faster.

4. The memory unit of a computer has 256K words of 32 bits each. The computer has an instruction format with 4 fields: an opcode field; a mode field to specify 1 of 7 addressing modes; a register address field to specify 1 of 60 registers; and a memory address field. Assume an instruction is 32 bits long. Answer the following:
a) How large must the mode field be?
$7\ addresses=8=23=3\ bits$

Therefore, the mode field must be 3 bits large.
b) How large must the register field be?
$60\ registers = 64 = 2^6 = 6\ bits$
Therefore, the register field must be 6 bits large.

c) How large must the address field be?
$256K \times 32 = 256 \times 1024 \times 32 = 2^8 \times 2^{10} \times 2^5 = 2^{23} = 23\ bits$
Therefore, the address field must be 22 bits large.

d) How large is the opcode field
$Opcode\ field = 32 - 23 - 6 - 3 = 0$
There the opcode field is 0 bit large. Hence, it means that the total instruction length of 32 bits is not sufficient to accommodate all the required fields.