Homework

Author: Amata ANANTAPRAYOON,
Student ID: aa224iu,
Semester: VT2018
Course: Computer Technology 1
Course code: 1DT301

Contents

1 What are the five functional units of a computer?

- Input Unit: Transfers this information to the memory or processor.
- Output Unit: Convert the information in binary form to a form understood by an output device.
- Memory Unit: Stores instructions and data.
- Control Unit:
  - Accepts information from the input units (Input unit).
  - Stores the information (Memory).
  - Processes the information (ALU).
  - Provides processed results through the output units (Output unit).
- Arithmetic and Logic Unit: Arithmetic operations such as addition, subtraction, Logic operations such as comparison of numbers.

2 In a byte addressable memory with a 64-bit address, what is the maximum size of memory?

$2^{64} = 1.8 * 10^{19}$ Bytes.

3 Describe the addressing modes you have learned.

- Register Mode:
  - Operand is the contents of a processor register
  - Address of the register is given in the instruction.
  - For instance clear R1
- Absolute Mode (Direct Mode):
  - Operand is in a memory location.
  - address of the memory location is given explicitly in the instruction.
- Immediate Mode: Operand is given explicitly in the instruction. For instance MOVE 200,R0
- Indirect Mode: Effective Address of the operand is the contents of a register or a memory location whose address appears in the instruction. For instance ADD (R1),R0 and ADD (A),R0. We called R1 and A "pointers"
- Indexing Mode: Effective Address of the operand is generated by adding a constant value to the contents of the register.For instance "ADD 20(R1), R0. Read more information about how this mode in question 4

- Relative Mode:

  - Effective Address of the operand is generated by adding a constant value to the contents of the Program Counter (PC)

  - Variation of the Indexing Mode, where the index register is the PC instead of a general-purpose register

  - It is useful for specifying target addresses in branch instructions.

- Autoincrement Mode: After accessing the operand, the contents of this register are automatically incremented to point to the **next consecutive memory location**. For instance (R1)+. This mode are useful for implementing "LIFO" (Last-In-First-Out) data structure.

- Autodecrement Mode: Before accessing the operand, the contents of this register are automatically decremented to point to the **previous consecutive memory location**. For instance -(R1). This mode are useful for implementing "LIFO"

4 Describe what the code “Add 20(R1), R0” computes?

In this case R1 is an "index register". Assume that R1 contains 100. 20(R1) means offsets 20 is added to the contents of R1 to generate the address 120. during process of generating process, content of R1 is not change. Operand is at address 120. This addressing mode call "indexing mode".

5 Write an assembly language code which adds 100 numbers. Assume that it is a 32-bit machine and all the numbers are put in an array with the starting address of 1000.

```
.equ num = 100
.equ currentAddress = 1000
mov num, R16
mov currentAddress, R17


loop:
add (R17), R16
add #4, R17
dec R16
cpi R16, 1
; if R16 > 1 then loop else no operation (already added100 numbers)
brne loop



nop
```

6 Describe what the code “Move (SP)+, A” computes.

Removing the top item at the address A from the stack and increment the address of the stack pointer SP with 4 bytes automatically.

7 Describe the difference between a RISC machine and a CISC machine.

- **Reduced Instruction Set Computer RISC**: Only use simple instructions that can be executed within one clock cycle. That means, a programmer would need to code more lines of assembly too be able perform the exact series of steps as CISC machine.
- **Complex Instruction Set Computer CISC**: The primary goal of CISC architecture is to complete a task in as few lines of assembly as possible. This is achieved by building processor hardware that is capable of understanding and executing a series of operations.

Differences between RISC and CISC:

RISC:	**CISC**:
Emphasis on software.	Emphasis on hardware
Single-clock,reduced instruction only	Includes multi-clock,complex instructions
Register to register: "LOAD" and "STORE" are independent instructions	Memory-to-memory:"LOAD" and "STORE" incorporated in instructions
Large code sizesLow cycles per second	Small code sizes, high cycles per second
Spends more transistors on memory registers	Transistors used for storing complex instructions

8 Describe what an edge triggered D flipflop is.

The flip flop is a basic building block of sequential logic circuits. It is a circuit that has two stable states and can store one bit of state information. D flipflop has a D (data) input and a clock input and outputs Q and $\bar{Q}$ (the inverse of Q).

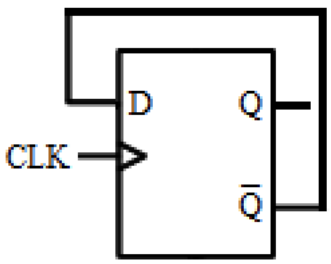

Figure 1: D Flip Flop

Edge-triggered D flip-flop has only one input addition to the clock. It is very useful when a single data bit (0 or 1) is to be stored.
If there is a **LOW** on the D input when a clock pulse is applied, the flip-flop RESETs and stores a 0. In another hand,if there is a **HIGH** on the D input when a clock pulse is applied, the flip-flop SETs and stores a 1. The negative edge-triggered flip-flop works the same except that the falling edge of the clock pulse is the triggering edge.

9 Describe the 3-clock cycle computation process for the code "Add R1, R2, R3"

- **First clock cycle**: Place the contents of register R1 into the Y register ($R1_{out} \rightarrow Y_{in}$).
- **Second clock cycle**: Place the contents of register R2 onto the bus, then place it in input B ($R2_{out} \rightarrow$B). Select register Y in MUX then place Y in input A in($Y \rightarrow A$). From this point, both inputs to the ALU are valid. Assert ALU command ADD and place the result in register Z (A+B $\rightarrow Z_{in}$).
- **Third clock cycle**: Place the content of Z register into Register R3 ($Z_{out} \rightarrow R3_{in}$).

10 Draw the diagram to compute A-B assuming that they are 8-bit numbers