1DT301: Computer Technology

Homework 1

Due in class on October 10, 2018

Adell Tatrous

at222ux

1. What are the five functional units of a computer?

Input Unit, Output Unit, Memory Unit, Control Unit and Arithmetic and Logic Unit (ALU).

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

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.

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. It can be used to represent constants.

Indirect Mode:

Effective Address of the operand is the contents of a register or a memory location whose address appears in the instruction.

Indexing Mode:

Effective Address of the operand is generated by adding a constant value to the contents of the register.

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. When the instruction is being executed, the PC holds the address of the next instruction in the program. It is useful for specifying target addresses in branch instructions.

Autoincrement Mode:

Effective address of the operand is the contents of a register specified in the instruction. After accessing the operand, the contents of this register are automatically incremented to point to the next consecutive memory location.

Autodecrement Mode:

Effective address of the operand is the contents of a register specified in the instruction. Before accessing the operand, the contents of this register are automatically decremented to point to the previous consecutive memory location.

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

R1 is the “index register” and contains e.g. 1000 and the operand is at address 1020. Offset 20 is added to the contents of R1 to generate the address 1020.

Contents of R1 do not change in the process of generating the address.

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.

.def numberOfElements = r16

.def sum = r17

.def currentValue = r18

.equ N = 100

.equ address = #1000

ldi numberOfElements, N

mov currentValue, address

clr sum

loop:

add sum, currentValue

add currentValue, #4

dec numberOfElements

cpi numberOfElements, 1

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.

Complex Instruction Set Computer (CISC) refers to processors using the Two-word instruction format where a second word is included as a part of the instruction to specify the address of a memory location or an immediate operand.

Reduced Instruction Set Computers (RISC) refers to processors using the Three-operand instruction format where a restriction might occur that an instruction must occupy only one word. Manipulation of data must be performed on operands already in processor registers. Restriction may require additional instructions for tasks.

8. Describe what an edge triggered D flipflop is.

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 HIGH on the D input when a clock pulse is applied, the flip-flop SETs and stores a 1.

If there is a LOW on the D input when a clock pulse is applied, the flip-flop RESETs and stores a 0.

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”.

1. Place the contents of register R1 into the Y register in the first clock cycle.

2. Place the contents of register R2 onto the bus in the second clock cycle. Both

inputs to the ALU are now valid. Select register Y, and assert the ALU command F=A+B.

3. In the third clock cycle, Z register has latched the output of the ALU. Thus the

contents of the Z register can be copied into register R3.

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

