**Note:**  (Late: -15% Penalty).

1. (5 pts) Fill in the following table to show how the given integers are represented, assuming 16-bits are used to store values and the machine uses 2’s complement notation.

|  |  |  |  |
| --- | --- | --- | --- |
| Integer | Binary | Hex | 4 Byte Little Endian  (hex value as seen in memory) |
| 28 | 0000000000011100 | 001C | 0x1C00 |
| 2216 | 0000100010101000 | 08A8 | 0xA808 |
| -18675 | 1011011100001101 | B70D | 0xODB7 |
| -12 | 1111111111110100 | FFF4 | 0xF4FF |
| 31456 | 0111101011100000 | 7AE0 | 0xE07A |

1. (5 pts) Convert the following expressions from infix to reverse Polish (postfix) notation.
   1. (8–6)/2

Answer: 86-2/

* 1. (2+3)\*8/10

Answer: 23+8\*10/

* 1. (5×(4+3)×2–6)

Answer: 43+5×2×6-

1. (5 pts) Explain how a stack is used to evaluate the RPN (reverse polish notation) expression 3 5 7 + 2 1 - \* 1 + +

Answer is 16.

1. (5 pts) Define:
   1. Immediate addressing

The operand is next to the opcode in the instruction

* 1. Direct addressing

The address of the operand is next to the opcode in the instruction.

* 1. Indirect addressing

The address of the address that holds the pointer to the operand is in the instruction next to the opcode.

* 1. Indexed addressing

The rear part address portion in the instruction is the index.

The portion in the front is the base. When you put together the base and the index. You get the address to the operand.

1. (5 pts) A nonpipelined system takes 200ns to process a task. The same task can be processed in a 5-segment pipeline with a clock cycle of 40ns. Determine the speedup ratio of the pipeline for 200 tasks. What is the maximum speedup that could be achieved with the pipeline unit over the nonpipelined unit?

Max speed up 4.87

Work Shown below.

1. (5 pts) Suppose we have the instruction “Load1000”. Given memory and register R1 contain the values below, and assuming R1 is implied in the indexed addressing mode, determine the actual value loaded into the accumulator and fill in the table below:

|  |  |
| --- | --- |
| Memory | |
| Address | Data |
| 0x1000 | 0x1400 |
|  |  |
| 0x1100 | 0x400 |
|  |  |
| 0x1200 | 0x1000 |
|  |  |
| 0x1300 | 0x1100 |
|  |  |
| 0x1400 | 0x1300 |

|  |
| --- |
| R1 |
| 0x200 |

|  |  |
| --- | --- |
| Mode | Value Loaded into AC |
| Immediate | 0x1000 |
| Direct | 0x1400 |
| Indirect | 0x1300 |
| Indexed | 0x1000 |

1. Assuming the same stages as in Example 5.11, explain the potential pipeline hazards (if any) in each of the following code segments.

Check Work on the back

1. (15 pts) Write working assembly code that successfully executes the following stack reverse Polish (postfix) based code.
   1. Your code must include the use of sub-routines Push, Pop, Subtract, Add

Push A

Push B

Push C

Add

Push D

Push F

Subtract

Subtract

Push E

Add

Add

Pop X

A, 3

B, 5

C, 7

D, 2

E, 4

F, 1

* 1. What is the final value for X? X=18 Work is on the back.