Tutorial #8

CSE 321a: Computer Organization (I)
Third Year, Computer and Systems Engineering

Question (4) – Final –Fall 2014
Consider a simple computer with a **big-endian byte-addressable** memory and a processor that has six 16-bit registers: a program counter (PC), a stack pointer (SP), and four general-purpose registers (R0, R1, R2 and R3). The processor supports a mix of two-, one- and zero-operand instructions. Instructions can operate on either 16-bit data (**words**) or 8-bit data (**half words**). The supported instruction set has a **variable-length** format described as follows:
• Two-operand instructions (**24-bit)**: 8-bit opcode followed by two 8-bit operands.
• One-operand instructions (**16-bit)**: 8-bit opcode followed by one 8-bit operand.
• Zero-operand instructions (**8-bit)**: 8-bit opcode.

➢ Operands (8-bit): 2-bit addressing mode (AM) followed by 6-bit value/address (VA).
➢ Addressing-mode (2-bit): 00 → immediate, 01 → direct, 10 → register, 11 → base register
(where register number is encoded by the least significant two bits of VA).

1. What is the total number of opcodes supported by this variable-length instruction format, and what is the maximum effective address that can be specified using the direct addressing mode?

Total number of opcodes = 28 = 256
Maximum effective address = 26 – 1 = 63

2. What is the machine code that corresponds to each of the following assembly instructions, given that their opcodes (in **hexadecimal**) are 2F, 94, DF, 9D, and A8 respectively?

(a) SKIP_WORD
Zero-operand ==> 1 byte ==> 2F

(b) PUSH_WORD R1
One-operand ==> 2 byte ==> 94(10XXXX01)b ==> 94{8, 9, A, B}{1, 5, 9, D}

(c) ADD_WORD R1(2),#2E
One-operand ==> 3 byte ==> DF(11001001)b(00101110)b ==> DFC92E

(d) EXECUTE_HALF_WORD 1C
One-operand ==> 2 byte ==> 9D(01011100)b ==> 9D5C
(e) ARITH_SHIFT_RIGHT_WORD R2
One-operand ==> 2 byte ==> A8(10XXXX10)b ==> A8{8, 9, A, B}{2, 6, A, E}

3. Suppose **before fetching** each of the instructions in part 2, the (only relevant) contents of the registers and memory (in **hexadecimal**) were:
• Registers: [PC] = 3B8A, [SP] = 25E0, [R1] = 001A, [R2] = A5A5
• Memory: [001C] = 2F, [001D] = 93, [25DE] = 11, [25DF] = 27
Show the contents of the registers and memory **after executing** each of those instructions.

(a) SKIP_WORD
[PC] = 3B8D

(b) PUSH_WORD R1
[PC] = 3B8C, [SP] = 25DE, [25DE] = 00, [25DF] = 1A

(c) ADD_WORD R1(2),#2E
[PC] = 3B8D, [001C] = 2F, [001D] = C1

(d) EXECUTE_HALF_WORD 1C
[PC] = 3B8E

(e) ARITH_SHIFT_RIGHT_WORD R2
[PC] = 3B8C, [R2] = D2D2

External problem:
Design Variable length opcode where all instructions encoded in 36 bits.
Instruction format can be one of three categories:

1. Instructions with Two operands of 15 bits memory address and one operand of 3 bits represent register number.
2. Instructions with one operand of 15 bits memory address and one operand of 3 bits represent register number.
3. Zero operand instructions.

There are many method of design one of them is as shown below:

<u>For three operand instructions:</u>

Opcode	operand1	operand2	operand3
3 bits	15 bits	15 bits	3 bits

Where opcodes:

$$\left.\begin{array}{l} 000 \\ 001 \\ \cdots \\ 110 \end{array}\right\} \text{Represent opcodes of three operand instructions}$$

$$111 \rightarrow \text{Indicate that \# of operand less than 3}$$

# of three operand opcodes = $2^3 - 1 = 7$ opcodes

<u>For two operand instructions:</u>

Opcode	operand1	operand2
111 18 bits	15 bits	3 bits

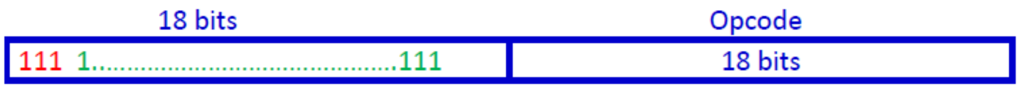
Where opcodes:

$$\left.\begin{array}{l} 111\ \overbrace{0\ldots000}^{15\text{ bits}} \\ 111\ 0\ldots001 \\ \cdots\cdots\cdots\cdots \\ 111\ 1\ldots110 \end{array}\right\} \text{Represent opcodes of two operand instructions}$$

$$111\ 1\ldots111 \rightarrow \text{Indicate that instruction is zero operand instruction}$$

# of two operands opcodes = $2^{15} - 1$ opcodes

For zero operand instructions:



# of zero operand opcodes = 2^{18} opcodes

Prob. 11.16 {Text book}

Assume an instruction set that uses a fixed length instruction of 16 bit. Operand specifiers are 6 bits in length. There are K two operand instructions and L zero operand instructions. What is the maximum # of one operand instructions that can be supported by that machine?

Instruction length = 16 bits → Total # of different form of instructions = 2^{16}

# of 2-operand instruction = $K * 2^6 * 2^6$

# of 0-operand instruction = L

Assume there is J instructions of one operand

# of 1-operand instruction = $J * 2^6$

:. $2^{16} = K * 2^6 * 2^6 + J * 2^6 + L$

:. $J = (2^{16} - K * 2^{12} - L)/ 26$

:. $J\text{max} = 2^{16} / 2^6 = 2^{10}$