CLASS EXERCISES

Topic Goals

Main Goals

- To know the basic characteristics of an assembly language and its corresponding machine language.
- To know the instruction set of a real processor and the methodologies for low level programming with this set.

Specific Goals

- List the characteristics of an Instruction Set
- Calculate the location of a data for a given addressing mode
- Write assembly language instruction sequences for different machines
- Explain the different fields that appear in the machine format of a given instruction
- Translate Z80 assembly language instructions to machine language and vice versa
- Interpret the meaning and simulate the execution of Z80 assembler programs
- State the differences between CISC and RISC approaches to instruction set architecture design
- Classify an instruction set as CISC or RISC

BIBLIOGRAPHY

- Chapters 10 and 11 from "Computer Architecture and Organization", W. Stallings
- Chapter 3 from "Modern Computer Architecture", J. Ledin
- Chapter 6 from "Fundamentos de Computadores", R. Hermida
- Z80 Material available in Canvas, including "Programming the Z80"

Exercises

Instruction Set Comparison

1. Compare 0, 1, 2 and 3 address computers by writing a program to calculate the expression Z = (A+B*C) / (D-E*F-G*H), all representing memory locations, given the following instructions (Where M is a memory address, X, Y are memory addresses or registers and R_n are registers):

0-Address	1-Address	2-Address	3-Address
PUSH M	LOAD M	MOVE X, Y	ADD R_i, R_j, R_k
POP M	STORE M	ADD X, Y	SUB R_i, R_j, R_k
ADD	ADD M	SUB X, Y	MUL R_i, R_j, R_k
SUB	SUB M	MUL X, Y	DIV R_i, R_j, R_k
MUL	MUL M	DIV X, Y	LOAD R, M
DIV	DIV M		STORE M, R

Addressing Modes

2. The address field of an instruction contains the decimal value 14. Indicate where the corresponding operand is located for each of the following addressing modes:

   a) immediate or literal

   b) direct

   c) indirect

   d) register

   e) register indirect

3. A computer has the following contents in registers and memory:

   R1 = 98 R2 = 2

   Memory:

Address	96	97	98	99	100	101	102	103	104	198	199	200	201	202
Content	99	102	101	104	102	106	107	109	110	45	100	34	96	201

   What would be the value of the operand with the following addressing modes?

   a) Direct, with the corresponding field of the instruction containing the value 100

   b) Register, with the corresponding field of the instruction containing the value 1

   c) Indirect register, with the corresponding field of the instruction containing the value 1

   d) Indirect (memory), with the corresponding field of the instruction containing the value 100

   e) Base plus displacement, with the corresponding fields of the instruction containing the values 1 (for the register) and 100 (for the offset).

   f) Base plus displacement, with the corresponding fields of the instruction containing the values 2 (for the register) and 100 (for the offset).

How many bits would it take to encode the above addressing modes if we assume that the processor has 16 general purpose registers and 16MB of main memory?

If we assume that T_{MEM} is the time to access memory for reading, T_{REG} is the time to access a register for reading and T_+ is the time needed to perform a sum, how long does it take to get the data with each of the modes above?

4. Given the following memory values, and assuming a machine with 1-address instructions, and with an accumulator, what values load the following instructions into the accumulator?

   MEM[20] = 40 MEM[30] = 50 MEM[40] = 60 MEM[50] = 70

   a) Immediate or literal 20

   b) Direct 20

   c) Indirect 20

   d) Immediate or literal 30

   e) Direct 30

   f) Indirect 30

Assembly Programming

5. Write a Z80 assembly program that adds two integers (X and Y) and leaves the result in Z. X, Y and Z correspond to the memory addresses 0xC000, 0xC001 and 0xC002 respectively.
6. Try the above program with different addressing modes: Register, Direct, Indirect Register, Base plus displacement.
7. Write a Z80 assembly language program that adds two integer vectors (A and B) and leaves the result in another vector, C. The size of the vectors is 3. A, B and C arrays are located at, 9000h, 9003h and 9006h.
8. Write a Z80 assembly program that, given two integers, A and B, determines which is the larger of the two and which is the smaller.
9. Write an algorithm in pseudo-code that computes the product of two integers using sums. Translate the algorithm into Z80 assembly language.
10. Write an algorithm, using pseudo-code, that performs the addition of two 10-component vectors using a FOR loop. Translate it to Z80 assembly language.
11. Write an algorithm in pseudo-code that computes the sum of the first 10 elements of a vector. Translate it to Z80 assembly language.
12. Write an algorithm, using pseudo-code, that finds the maximum value in a 10-component vector. Translate it to Z80 assembly language.

Assembly-Machine Code translation

13. Translate the following instructions, written in Z80 assembly, into binary and hexadecimal notation:

    a) ADD A, (IX-15) b) LD A, ($4000) c) DJNZ -2

    d) JR Z, -24 e) AND C f) XOR $FE

    (Note: take advantage of the exercise to take a look at the undocumented instructions (in red in the table) of the Z80)

14. Consider the following instructions in Machine Code

    a) 0111 1110 b) 1101 1101 0111 1110

    c) 1111 1101 0010 0011 d) 1101 1101 0010 0011

    e) 0010 0011 f) 1100 0010 0000 0000 1000 0000

    What Z80 instructions do they represent? Translate them to assembler format.

    Note whether there are prefixes and their relationship to the instructions without prefixes

    Don’t forget that Z80 is LITTLE ENDIAN