

Instruction Sets

EVALUATION SHEET

1. In Part A below, circle the *two* letters that identify the names of instruction word fields.

In Part B, write the letter of the field being defined in the space provided.

Part A

The two fields of an instruction word are:

- a. Instruction field
- b. Operand field
- c. Program field
- d. Word-size field
- e. Operation field
- f. Mnemonic field

Part B

1. The b tells the CPU where to find the data that are to be processed.
2. The e holds a binary code that tells the CPU exactly what to perform next.

2. Match each of the terms below with its definition.

Term	Definition
Operand	<u>b</u>
Op Code	<u>d</u>
Instruction Mnemonic	<u>a</u>
Accumulator	<u>c</u>

Definitions

- A 3- or 4-letter abbreviation that programmers use in place of the binary operation code.
- An item of data to be acted upon by an instruction.
- A special storage area contained in the CPU.
- A predefined binary code that tells the CPU what operation it is to perform.

3. Match each of the following instruction formats with its description.

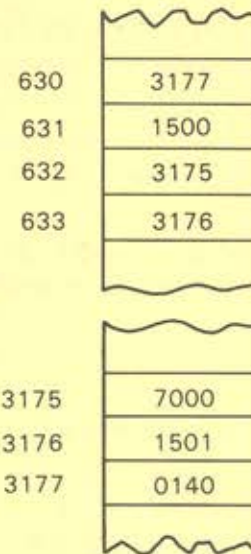
- | | | |
|-------------------------|----------|--|
| a. No operand field | <u>c</u> | If the instruction calls for an addition, the sum is placed in a memory location formerly occupied by one of the operands. |
| b. Single operand field | <u>a</u> | Memory is not referenced; the instruction operates directly on the contents of the AC. |
| c. Two operand fields | <u>b</u> | If the instruction calls for an addition, the contents of a memory location are added to the contents of the AC. |
| d. Three operand fields | <u>d</u> | Not widely used in mini-computers because it requires a large word size. |

4. The computer word size places a limit on the maximum number of memory locations that can be directly addressed. Three techniques that may be used to overcome this addressing limitation are: multiple-word instructions (MW), special registers (SR), and memory pages (MP).

Match each of the descriptions below with the technique it describes by writing the correct abbreviation in the space provided.

Description	Technique
Part of the CPU.	<u>SR</u>
Sometimes used for purposes other than addressing main memory.	<u>SR</u>
Segments of main memory.	<u>MP</u>
Used in place of single-word instructions.	<u>MW</u>
Size of each is chosen so that CPU can address any location by using the available address bit in the instruction.	<u>MP</u>

5. The diagram at the right specifies the contents of various memory locations. The table below lists several instructions that reference these memory locations. For each instruction, specify the addressing method (direct or indirect), the operand address and the operand.



Instruction	Addressing Method	Operand Address	Operand
ISZ I 632	Indirect	3175	7000
ADD 631	Direct	631	1500
ADD I 630	Indirect	3177	0140
ADD I 633	Indirect	3176	1501
ISZ 630	Direct	630	3177

6. Write a simple program that adds A, B, and C and then stores the answer (X) in memory location 333.

Known Factors: A is stored in location 330.
 B is stored in location 331.
 C is stored in location 332.

Restrictions: Use only the instructions defined in the lesson "Typical Instruction Set." Use 200 as the starting address of the program.

Address	Instruction or Data	Explanation*
200	CLA	Clear AC to all zeros.
201	ADD 330	Add A to contents of AC (zero).
202	ADD 331	Add B to A
203	ADD 332	Add C to (B + A)
204	STR 333	Store answer in location 333.
205	HLT	Stop.
.	.	
.	.	
.	.	
330	A	} Operands
331	B	
332	C	
333	X	Answer

* This is supplementary information. It is not expected to be part of the student's answer.

7. Write a program that solves the mathematical expression $Y = A - (B * 2)$ and then stores the answer (Y) in memory location 352.

Known Factors: A is stored in location 350.
 B is stored in location 351.

Restrictions: Use only the instructions defined in the lesson "Typical Instruction Set." Use 200 as the starting address of the program.

Address	Instruction or Data	Explanation*
200	CLA	Clear AC to all zeros.
201	ADD 351	Add B to contents of AC (zero).
202	ADD 351	Add B to B ($B * 2$).
203	CMA	Convert ($B * 2$) to 1's complement.
204	IAC	Convert ($B * 2$) to 2's complement.
205	ADD 350	Add A to $-(B * 2)$.
206	STR 352	Store answer in location 352.
207	HLT	Stop.
.	.	
.	.	
.	.	
350	A	} Operands
351	B	
352	Y	

* This is supplementary information. It is not expected to be part of the student's answer.

8. Write a program that multiplies 150_8 by 75_8 and then stores the answer in memory location 332. Use a program loop in your solution.

Known Factors: The operand 150_8 is stored in memory location 330; the operand 75_8 is stored in location 331.

Restrictions: Use only the instructions defined in the lesson "Typical Instruction Set." Use 200 as the starting address of your program.

Address	Instruction or Data	Explanation*
200	CLA	Clear AC to all zeros.
201	ADD 331	Load 75 into AC.
202	CMA	Convert 75 to 1's complement.
203	IAC	Convert 75 to 2's complement.
204	STR 331	Store -75 (7703) in location 331; clear AC.
205	ADD 330	Add 150 to AC.
206	ISZ 331	Increment tally (-75); skip if 0.
207	JMP 205	Otherwise, go back to ADD.
210	STR 332	Store answer in location 332.
211	HLT	Stop.
.	.	
.	.	
.	.	
330	0150	Operand.
331	0075	Tally (-75 or 7703).
332		Answer.

* This is supplementary information. It is not expected to be part of the student's answer.

- 9 Circle the *letter* of the mathematical expression that is solved by the following program.

200	CLA
201	ADD 216
202	CMA
203	IAC
204	STR 216
205	ADD 215
206	ISZ 216
207	JMP 205
210	STR 215
211	ADD 215
212	ADD 215
213	ADD 217
214	HLT
215	A
216	B
217	C

Answers

- a. $(2 + A * B) + C$
- b. $(2 * A + C) + B$
- c. $(2 * B * C) + A$
- d. $(2 * A * B) + C$
- e. $(2 * A + B) + C$

10. Indicate whether each of the following statements refers to a conditional instruction (C) or an unconditional instruction (U) by writing the correct letter in the space provided.

Statement	Instruction Type
If y is negative, branch to 277.	<u> C </u>
Branch to location 215 if x = 0.	<u> C </u>
Skip the next instruction in the sequence if A is positive.	<u> C </u>
Skip the next instruction in the sequence.	<u> U </u>
Jump to location 307.	<u> U </u>