AS · Cambridge (CIE) · Computer Science

32 mins 7 questions

Exam Questions

Assembly Language

Assembly language basics / Assembly process / Instruction groups / Addressing modes

1 The following table shows part of the instruction set for a processor. The processor has two registers, the Accumulator (ACC) and the Index Register (IX).

Instruction		Explanation
Opcode	**Operand**	
LDM	#n	Immediate addressing. Load the number n to ACC
LDD	<address>	Direct addressing. Load the contents of the location at the given address to ACC
LDI	<address>	Indirect addressing. The address to be used is at the given address. Load the contents of this second address to ACC
LDX	<address>	Indexed addressing. Form the address from <address> + the contents of the index register. Copy the contents of this calculated address to ACC
LDR	#n	Immediate addressing. Load the number n to IX
ADD	#n/Bn/&n	Add the number n to the ACC
ADD	<address>	Add the contents of the given address to the ACC
SUB	#n/Bn/&n	Subtract the number n from the ACC
SUB	<address>	Subtract the contents of the given address from the ACC
INC	<register>	Add 1 to the contents of the register (ACC or IX)
DEC	<register>	Subtract 1 from the contents of the register (ACC or IX)
<address> can be an absolute or a symbolic address # denotes a denary number, e.g. #123 B		

denotes a binary number, e.g. B01001010
& denotes a hexadecimal number, e.g. &4A

The current contents of memory are shown:

Address	Data
19	25
20	23
21	2
22	4
23	15
24	50
25	22

The current contents of the ACC and IX are shown:

ACC	50
IX	20

Complete the table by writing the content of the ACC and the IX after each set of instructions has run.

	Instructions	ACC content	IX content
1	LDM #19 DEC ACC		
2	LDD 23 ADD 19		
3	LDI 25 INC ACC		
4	LDR #21 LDX 2		

Answer

Mark Scheme and Guidance

1 mark for each correct content of the ACC (4)
1 mark for correct IX column

	Instructions	ACC content	IX content
1	LDM #19 DEC ACC	18	20
2	LDD 23 ADD 19	40	20
3	LDI 25 INC ACC	5	20
4	LDR #21 LDX 2	15	21

(5 marks)

2 The following table shows part of the instruction set for a processor. The processor has two registers, the Accumulator (ACC) and the Index Register (IX).

Instruction		Explanation
Opcode	**Operand**	
LDM	#n	Immediate addressing. Load the number n to ACC
LDD	<address>	Direct addressing. Load the contents of the location at the given address to ACC
LDI	<address>	Indirect addressing. The address to be used is at the given address. Load the contents of this second address to ACC
LDX	<address>	Indexed addressing. Form the address from <address> + the contents of the index register. Copy the contents of this calculated address to ACC
LDR	#n	Immediate addressing. Load the number n to IX
ADD	#n/Bn/&n	Add the number n to the ACC
ADD	<address>	Add the contents of the given address to the ACC
SUB	#n/Bn/&n	Subtract the number n from the ACC
SUB	<address>	Subtract the contents of the given address from the ACC
INC	<register>	Add 1 to the contents of the register (ACC or IX)
DEC	<register>	Subtract 1 from the contents of the register (ACC or IX)
<address> can be an absolute or a symbolic address # denotes a denary number, e.g. #123 B		

denotes a binary number, e.g. B01001010
& denotes a hexadecimal number, e.g. &4A

The current contents of memory are shown:

Address	Data
50	54
51	55
52	50
53	52
54	100
55	25
56	50

The current contents of the ACC and IX are shown:

ACC	50
IX	45

Complete the table by writing the content of the ACC after each program has run.

	Instructions	ACC content
1	LDD 50 ADD #4 ADD 54	
2	LDI 53 DEC ACC ADD 56	
3	LDM #55 SUB #5	

Answer

Mark Scheme and Guidance

1 mark each:

	Instructions	ACC content
1	LDD 50 ADD #4 ADD 54	**158**
2	LDI 53 DEC ACC ADD 56	**99**
3	LDM #55 SUB #5	**50**

(3 marks)

3 The following table shows part of the instruction set for a processor. The processor has two registers: the Accumulator (ACC) and an Index Register (IX).

Instruction		Explanation
Opcode	**Operand**	
LDM	#n	Immediate addressing. Load the number n to ACC
LDD	<address>	Direct addressing. Load the contents of the location at the given address to ACC
LDI	<address>	Indirect addressing. The address to be used is at the given address. Load the contents of this second address to ACC
LDX	<address>	Indexed addressing. Form the address from <address> + the contents of the index register. Copy the contents of this calculated address to ACC
LDR	#n	Immediate addressing. Load the number n to IX
ADD	#n/Bn/&n	Add the number n to the ACC
ADD	<address>	Add the contents of the given address to the ACC
SUB	#n/Bn/&n	Subtract the number n from the ACC
SUB	<address>	Subtract the contents of the given address from the ACC
INC	<register>	Add 1 to the contents of the register (ACC or IX)
<address> can be an absolute or a symbolic address # denotes a denary number, e.g. #123 B denotes a binary number, e.g. B01001010 & denotes a hexadecimal number, e.g. &4A		

The current contents of memory are shown:

Address	Data
19	24
20	2
21	1
22	3
23	5
24	4
25	22

The current contents of the ACC and IX are shown:

ACC	12
IX	1

Complete the table by writing the content of the ACC after each program has run.

Program number	Instructions	ACC content
1	LDD 20 ADD #2	
2	LDX 22	
3	LDI 25 INC ACC SUB 22	
4	LDD 19 LDM #5 LDM #25	

Answer

Mark Scheme and Guidance

1 mark for each correct answer:

Program number	Instructions	ACC content
1	LDD 20 ADD #2	**4**
2	LDX 22	**5**
3	LDI 25 INC ACC SUB 22	**1**
4	LDD 19 LDM #5 LDM #25	**25**

(4 marks)

4 The following table shows part of the instruction set for a processor. The processor has two registers: the Accumulator (ACC) and an Index Register (IX).

Instruction		Explanation
Opcode	**Operand**	
LDM	#n	Immediate addressing. Load the number n to ACC
LDD	<address>	Direct addressing. Load the contents of the location at the given address to ACC
LDI	<address>	Indirect addressing. The address to be used is at the given address. Load the contents of this second address to ACC
LDX	<address>	Indexed addressing. Form the address from <address> + the contents of the index register. Copy the contents of this calculated address to ACC
LDR	#n	Immediate addressing. Load the number n to IX
ADD	#n/Bn/&n	Add the number n to the ACC
ADD	<address>	Add the contents of the given address to the ACC
SUB	#n/Bn/&n	Subtract the number n from the ACC
SUB	<address>	Subtract the contents of the given address from the ACC
INC	<register>	Add 1 to the contents of the register (ACC or IX)
<address> can be an absolute or a symbolic address # denotes a denary number, e.g. #123 B denotes a binary number, e.g. B01001010 & denotes a hexadecimal number, e.g. &4A		

The current contents of memory are shown:

Address	Data
10	1
11	3
12	5
13	11
14	10
15	16
16	12

The current contents of the ACC and IX are shown:

ACC	10
IX	0

Complete the table by writing the content of the ACC after each program has run.

Program number	Instructions	ACC content
1	LDI 15 SUB #1	
2	LDD 14 ADD 11	
3	LDM #11 ADD #3 SUB 16	
4	LDR #2 LDX 14 ADD #2	

Answer

Mark Scheme and Guidance

1 mark for each correct answer:

Program number	Instructions	ACC content
1	LDI 15 SUB #1	**11**
2	LDD 14 ADD 11	**13**
3	LDM #11 ADD #3 SUB 16	**2**
4	LDR #2 LDX 14 ADD #2	**14**

(4 marks)

5 The following table shows part of the instruction set for a processor. The processor has two registers: the Accumulator (ACC) and an Index Register (IX).

Instruction		Explanation
Opcode	**Operand**	
LDM	#n	Immediate addressing. Load the number n to ACC
LDD	<address>	Direct addressing. Load the contents of the location at the given address to ACC
LDI	<address>	Indirect addressing. The address to be used is at the given address. Load the contents of this second address to ACC
LDX	<address>	Indexed addressing. Form the address from <address> + the contents of the index register. Copy the contents of this calculated address to ACC
LDR	#n	Immediate addressing. Load the number n to IX
ADD	#n/Bn/&n	Add the number n to the ACC
ADD	<address>	Add the contents of the given address to the ACC
SUB	#n/Bn/&n	Subtract the number n from the ACC
SUB	<address>	Subtract the contents of the given address from the ACC
INC	<register>	Add 1 to the contents of the register (ACC or IX)
<address> can be an absolute or a symbolic address # denotes a denary number, e.g. #123 B denotes a binary number, e.g. B01001010 & denotes a hexadecimal number, e.g. &4A		

The current contents of memory are shown:

Address	Data
48	51
49	6
50	48
51	50
52	49
53	50
54	6

The current contents of the ACC and IX are shown:

ACC	2
IX	50

Complete the table by writing the content of the ACC after each program has run.

Program number	Instructions	ACC content
1	LDM #50 INC ACC SUB #1	
2	LDI 51 ADD 52	
3	LDR #2 LDX 50 DEC ACC	
4	LDD 52 SUB 54 INC ACC	

Answer

Mark Scheme and Guidance

1 mark for each correct answer:

Program number	Instructions	ACC content
1	LDM #50 INC ACC SUB #1	**50**
2	LDI 51 ADD 52	**97**
3	LDR #2 LDX 50 DEC ACC	**48**
4	LDD 52 SUB 54 INC ACC	**44**

(4 marks)

6 (a) Identify the purpose of the first pass of a two-pass assembler.

Answer

Mark Scheme and Guidance

1 mark for:

- To create a symbol table

(1 mark)

(b) The following table shows part of the instruction set for a processor. The processor has two registers, the Accumulator (ACC) and the Index Register (IX).

Instruction		Explanation
Opcode	**Operand**	
LDR	#n	Immediate addressing. Load the number n to IX
STO	<address>	Store contents of ACC at the given address
ADD	<address>	Add the contents of the given address to the ACC
INC	<register>	Add 1 to the contents of the register (ACC or IX)
CMP	#n	Compare the contents of ACC with number n
JPE	<address>	Following a compare instruction, jump to <address> if the compare was True
OUT		Output to the screen the character whose ASCII value is stored in ACC
<address> can be an absolute or symbolic address # denotes a denary number, e.g. #123		

(i) Give one example of an instruction that belongs to each of the following instruction groups.

Only use the instructions given in the table. Each instruction must have a suitable operand.

Data movement .....

Arithmetic operation .....

Conditional instruction .....

(3)

(ii) The instruction LDR #2 uses immediate addressing.

Give **one** similarity and **one** difference between direct addressing and indexed addressing.

(2)

(iii) Identify **one** other mode of addressing.

(1)

Answer

Mark Scheme and Guidance

(i) **1 mark** for each bullet point

- Data movement: e.g. LDR #50 // STO 201
- Arithmetic operation: e.g. ADD 100 // INC IX
- Conditional instruction: e.g. JPE 96

(ii) **1 mark** for each bullet point (**max 2**)

Similarity:

- both load the contents of an address into the Accumulator

Difference:

- direct accesses the address given by the operand whereas indexed adds the contents of IX to the operand and accesses the data at that calculated address

(iii) **1 mark** for

- Indirect (addressing)
- Relative (addressing)

(6 marks)

7 (a) The following table shows part of the instruction set for a processor. The processor has two registers, the Accumulator (ACC) and the Index Register (IX).

Instruction		Explanation
Opcode	**Operand**	
LDD	<address>	Direct addressing. Load the contents of the location at the given address to ACC
LDX	<address>	Indexed addressing. Form the address from <address> + the contents of the index register. Copy the contents of this calculated address to ACC
LDR	#n	Immediate addressing. Load the number n to IX
STO	<address>	Store the contents of ACC at the given address
ADD	#n	Add the denary number n to the ACC
JMP	<address>	Jump to the given address
INC	<register>	Add 1 to the contents of the register (ACC or IX)
CMP	<address>	Compare the contents of ACC with the contents of <address>
CMI	<address>	Indirect addressing. The address to be used is at the given address. Compare the contents of ACC with the contents of this second address
JPE	<address>	Following a compare instruction, jump to <address> if the compare was True
IN		Key in a character and store its ASCII value in ACC

OUT		Output to the screen the character whose ASCII value is stored in ACC
END		Return control to the operating system
<address> can be an absolute or a symbolic address # denotes a denary number, e.g. #123		

The instructions in the processor's instruction set can be grouped according to their function.

Identify the instruction group for each of the following opcodes.

IN .....

ADD .....

JPE .....

CMI .....

Answer

Mark Scheme and Guidance

1 mark for each bullet point

- IN - Input **and** output of data
- ADD - Arithmetic operations
- JPE - Unconditional and conditional instructions
- CMI - Compare instructions

(4 marks)

(b) The current contents of main memory and selected values from the ASCII character set are given on page 15.

Trace the program currently in memory using the trace table when the input is '1'.

Address	Instruction
10	LDR #0
11	IN
12	STO 101
13	LDX 110
14	CMP 100
15	JPE 21
16	LDD 101
17	ADD #16
18	INC IX
19	STO 100
20	JMP 13
21	OUT
22	END
...	
100	0
101	0
...	
110	51
111	65

ASCII value	Character
49	1
50	2
51	3
52	4
...	
65	A
66	B
67	C
68	D

Instruction address	ACC	IX	Memory address				Output
			100	101	110	111	
			0	0	51	65	

Answer

Mark Scheme and Guidance

1 mark for each set of shaded rows

Instruction address	ACC	IX	Memory address				Output
			100	101	110	111	
			0	0	51	65	
10		0					
11	49						
12				49			
13	51						
14							
15							
16	49						
17	65						
18		1					
19			65				
20							
13	65						
14							
15							
21							A
22							

(1 mark)