AS · Cambridge (CIE) · Computer Science

🕓 21 mins ❓ 7 questions

Exam Questions

Bit Manipulation

Bitwise operations

1 The instruction set also includes these bit manipulation instructions:

Instruction		**Explanation**
Opcode	**Operand**	
AND	#n/Bn/&n	BitwiseAND operation of the contents ofACC with the operand
AND	<address>	Bitwise AND operation of the contents of ACC with the contents of <address>
XOR	#n/Bn/&n	Bitwise XOR operation of the contents ofACC with the operand
XOR	<address>	Bitwise XOR operation of the contents of ACC with the contents of <address>
OR	#n/Bn/&n	Bitwise OR operation of the contents ofACC with the operand
OR	<address>	Bitwise OR operation of the contents of ACC with the contents of <address>
LSL	#n	Bits in ACC are shifted logically n places to the left. Zeros are introduced on the right-hand end
LSR	#n	Bits in ACC are shifted logically n places to the right. Zeros are introduced on the left-hand end
<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 content of the ACC is shown:

ACC	1	0	0	1	1	0	1	0

The table has three sets of instructions. The binary number 10011010 is reloaded into the ACC before each set of instructions is run.

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

	Instructions	ACC content
1	LSL #2	
2	ADD #5 AND #30	
3	OR B11110010 INC ACC	

(3)

(ii) Explain how bit manipulation can be used to test whether the binary number stored in the ACC represents an odd denary number.

Write the bit manipulation instruction that will be used.

(3)

Answer

Mark Scheme and Guidance

(i) **1 mark** each correct content of ACC

	Instructions	ACC content
1	LSL #2	**0110 1000**
2	ADD #5 AND #30	**0001 1110**
3	OR B11110010 INC ACC	**1111 1011**

(ii) **1 mark** each to **max 2** for explanation

- An odd binary number will have a 1 in the Least Significant Bit (LSB)
- A bit manipulation operation is required to access/mask only the LSB and clear all the others
- Compare the result of the masking with denary 1
- ... the result of the comparison will be true if the number is odd

1 mark for correct instruction

- AND B00000001 // AND #1 // AND &01

(6 marks)

2 The instruction set also includes these bit manipulation instructions:

Instruction		Explanation
Opcode	Operand	
AND	#n/Bn/&n	Bitwise AND operation of the contents of ACC with the operand
AND	<address>	Bitwise AND operation of the contents of ACC with the contents of <address>
XOR	#n/Bn/&n	Bitwise XOR operation of the contents of ACC with the operand
XOR	<address>	Bitwise XOR operation of the contents of ACC with the contents of <address>
OR	#n/Bn/&n	Bitwise OR operation of the contents of ACC with the operand
OR	<address>	Bitwise OR operation of the contents of ACC with the contents of <address>
<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		

Explain how bit manipulation can be used to clear the data in an 8-bit register.

Write the bit manipulation instruction that will be used.

Answer

Mark Scheme and Guidance

> **1 mark** for each bullet point for the explanation
>
> **1 mark** for correct instruction
>
> - A bit manipulation operation is required to set all the bits to zero
> - Compare the result of the masking with 0
> - … the result of comparison will be true if the register is cleared
> - AND B00000000 / #00 / &00

(3 marks)

3 Describe the difference between a right logical binary shift and a right arithmetic binary shift.

Answer

1 mark each:

- A logical shift moves all bits to the right and inserts zeros in the appropriate leftmost bits

- An arithmetic shift moves all bits to the right but copies the sign bit into the Most Significant Bit (MSB)

(2 marks)

4 The processor includes these bit manipulation instructions:

Instruction		Explanation
Opcode	Operand	
AND	#n/Bn/&n	Bitwise AND operation of the contents of ACC with the operand
AND	<address>	Bitwise AND operation of the contents of ACC with the contents of <address>
XOR	#n/Bn/&n	Bitwise XOR operation of the contents of ACC with the operand
XOR	<address>	Bitwise XOR operation of the contents of ACC with the contents of <address>
OR	#n/Bn/&n	Bitwise OR operation of the contents of ACC with the operand
OR	<address>	Bitwise OR operation of the contents of ACC with the contents of <address>
<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
30	01110101
31	11111111
32	00000000
33	11001100
34	10101010

The current content of the ACC is shown:

1	0	0	1	1	0	1	0

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

The binary number 10011010 is reloaded into the ACC before each program is run.

Program number	Code	ACC content
1	AND 31	
2	XOR B01001111	
3	OR #30	

Answer

Mark Scheme and Guidance

1 mark for each correct answer:

Program number	Code	ACC content
1	AND 31	**1001 1010**
2	XOR B01001111	**1101 0101**
3	OR #30	**1001 1110**

(3 marks)

5 The processor includes these bit manipulation instructions:

Instruction		Explanation
Opcode	**Operand**	
AND	#n/Bn/&n	Bitwise AND operation of the contents of ACC with the operand
AND	<address>	Bitwise AND operation of the contents of ACC with the contents of <address>
XOR	#n/Bn/&n	Bitwise XOR operation of the contents of ACC with the operand
XOR	<address>	Bitwise XOR operation of the contents of ACC with the contents of <address>
OR	#n/Bn/&n	Bitwise OR operation of the contents of ACC with the operand
OR	<address>	Bitwise OR operation of the contents of ACC with the contents of <address>
<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
25	11000110
26	11100001
27	10000001
28	11001101
29	00001111

The current content of the ACC is shown:

0	1	0	0	0	1	1	0

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

The binary number 01000110 is reloaded into the ACC before each program is run.

Program number	Code	ACC content
1	XOR 29	
2	AND #29	
3	OR B11111111	

Answer

Mark Scheme and Guidance

1 mark for each correct answer:

Program number	Code	ACC content
1	XOR 29	0100 1001
2	AND #29	0000 0100
3	OR B11111111	1111 1111

(3 marks)

6 The processor includes these bit manipulation instructions:

Instruction		Explanation
Opcode	Operand	
AND	#n / Bn / &n	Bitwise AND operation of the contents of ACC with the operand
AND	<address>	Bitwise AND operation of the contents of ACC with the contents of <address>
XOR	#n / Bn / &n	Bitwise XOR operation of the contents of ACC with the operand
XOR	<address>	Bitwise XOR operation of the contents of ACC with the contents of <address>
OR	#n / Bn / &n	Bitwise OR operation of the contents of ACC with the operand
OR	<address>	Bitwise OR operation of the contents of ACC with the contents of <address>
LSL	#n	Bits in ACC are shifted logically n places to the left. Zeros are introduced on the right-hand end
LSR	#n	Bits in ACC are shifted logically n places to the right. Zeros are introduced on the left-hand end
<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
100	00001101
101	10111110
102	11110011
103	00110111
104	00000000

The current content of the ACC is shown:

1	1	1	1	1	1	1	1

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

The binary number 11111111 is reloaded into the ACC before each instruction is run.

Instruction number	Instruction	ACC content
1	LSL #2	
2	XOR 100	
3	AND 103	

Answer

Mark Scheme and Guidance

1 mark for each correct answer:

Instruction number	Instruction	ACC content
1	LSL #2	**1111 1100**
2	XOR 100	**1111 0010**
3	AND 103	**0011 0111**

(1 mark)

7 The following table shows another part of the instruction set for the same processor.

Instruction		Explanation
Opcode	**Operand**	
AND	Bn	Bitwise AND operation of the contents of ACC with the operand
XOR	Bn	Bitwise XOR operation of the contents of ACC with the operand
LSR	#n	Bits in ACC are shifted logically n places to the right. Zeros are introduced on the left hand end
# denotes a denary number, e.g. #123 B denotes a binary number, e.g. B01001101		

(i) The current contents of the ACC are:

0	1	0	0	1	1	1	1

Show the contents of the ACC after the execution of the following instruction.

AND B10100101

(1)

(ii) The current contents of the ACC are:

0	0	0	1	0	1	1	1

Show the contents of the ACC after the execution of the following instruction.

LSR #3

(1)

(iii) The current contents of the ACC are:

1	1	1	1	0	1	1	1

Show the contents of the ACC after the execution of the following instruction.

XOR B00100101

(1)

Answer

Mark Scheme and Guidance

(i) 0000 0101

(ii) 0000 0010

(iii) 1101 0010

(3 marks)