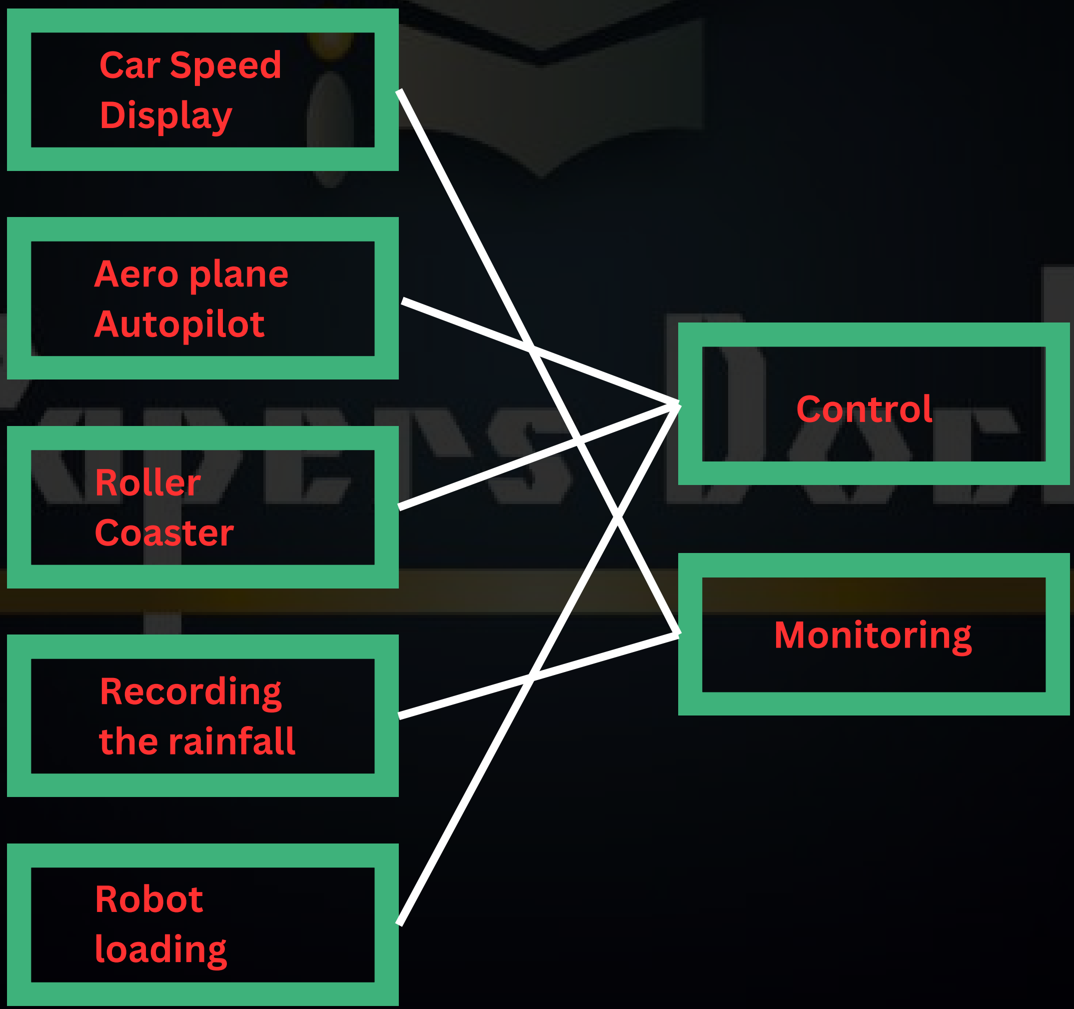
Monitoring And Control System And Bit Manipulation

COMPUTER SCIENCE 9618 PAPER 1

Monitoring And Control System

What is the difference between monitoring and control systems?

- Monitoring only gathers information, but control system also performs action.
- Output in a control system can affect the input.
- There is no feedback in a monitoring system, but a control system relies on feedback.



Sensors

- A physical quantity such as temperature, speed, moisture, oxygen level are measured by sensors.
- ADC (Analogue to Digital Converter)

Actuators

- An actuator is a component of a machine that is responsible for moving and controlling a mechanism.
- It receives a signal from the processor and adjusts a controlling device.
- DAC (Digital to Analogue Converter)

Types of Sensors and Their Uses

- Temperature: Measures and records temperature in particular surroundings.
- Pressure: Detects pressure and measures it, can be used in lifts or factories.
- Infra-Red: Detects motion, can be used in alarm systems.
- Sound: Detects sound, can be used in hotels for turning on lights.
- pH: Measures the pH value, can be used in farming.
- Moisture: Measures the level of moisture, used in soil.
- Humidity: Measures the level of moisture in air.
- Precipitation: Measures the amount of rainfall.

Hardware Devices

- ADC: To convert analogue signals from a sensor to a digital value that can be recorded.
- Actuator: E.g., turns the sprinkler on or off.
- Sensors:
- Processor: Processes the value recorded by the sensor.
- Cable: To transfer signal.
- Visible warning device: To give warnings to humans.

Question : How monitoring and control system will help achieve _________?

- Name of sensors used: _________
- What will those sensors do: _________
- These sensors will read the value and send it to the ADC.
- ADC will convert the analogue signal into digital signals.
- ADC sends data to the microprocessor.
- If control system is used, the microprocessor will compare the values with the provided range.
- The microprocessor will send a signal to the actuator.
- Purpose of an actuator: _________
- If more tasks need to be done, send the signal to another actuator.
- Purpose of actuator: _________

Concept Of Feedback

Feedback is the data which is collected from sensors and is analyzed or used as input.

Question : What is the importance of feedback?

- To ensure that the system operates within the given criteria.
- By enabling system output to affect subsequent system inputs.
- Thus enabling the system to automatically adjust conditions.

Question : What is the use of feedback?

- Sensor will continually measure ________.
- Readings are sent to the processor, and it compares the reading with the preset value.
- If readings are out of range, then actuators will do ________.
- Feedback will ensure that the readings remain in range.

(c) The equipment records the temperature in all seven rooms in the museum.

Each recording is stored as two successive bytes in memory. The format is as shown.

Temperature — **Byte 1**

Room — **Byte 2**

7	6	5	4	3	2	1	0

The room is indicated by the setting of one of the bits in **Byte 2** to 1. For example, room 7 is indicated by setting bit 7 to 1.

Bit 0 of **Byte 2** is a flag:

- The flag's initial value is zero.
- When the reading has been processed, the flag's value is set to 1.

Byte 1 contains the temperature reading as an unsigned integer.

One reading returns the following binary data.

Temperature — **Byte 1**

1	0	1	1	0	0	1	1

Room — **Byte 2**

7	6	5	4	3	2	1	0
0	0	1	0	0	0	0	1

(i) Analyse the data contained in the two bytes.

The temperature reading is 179

Reading is of Room Number 5

and the reading has been processed

[3]

(ii) The system receives a temperature reading of 238 from room number 4.

Complete the bytes to show the two bytes for this recording. The reading has not yet been processed.

Byte 1

1	1	1	0	1	1	1	0

Byte 2

7	6	5	4	3	2	1	0
0	0	0	1	0	0	0	0

[2]

Binary Shifting

Logical **Arithmetic** **Cyclic**

LOGICAL: **Bits shifted out of the registers are replaced by zeros.**

Logical Shift Left Once (LSL #1)

1	0	1	1	0	1	1	0

0	1	1	0	1	1	0	0

Logical Shift Right Twice (LSR #2)

1	0	1	1	0	1	1	0

0	0	1	0	1	1	0	1

Logical Shift In Assembly Language

OPCODE	OPERAND	EXPLANATION
LSL	#n	Bits in ACC are shifted logically n places to the left.
LSR	#n	Bits in ACC are shifted logically n places to the right.

(c) (i) The Accumulator currently contains the binary number:

0	0	1	1	0	1	0	1

Write the contents of the Accumulator after the processor has executed the following instruction:

```
LSL #2
```

1	1	0	1	0	1	0	0

[1]

(ii) The Accumulator currently contains the binary number:

0	0	1	1	0	1	0	1

Identify the mathematical operation that the following instruction will perform on the contents of the accumulator.

```
LSR #3
```

Division By 8

[1]

Cyclic Shifting: No bits are lost during a shift. Bits shifted out of one end of the register are inserted at the other end of the register.

Cyclic Shift Left Twice CSL #2

1	0	1	1	0	1	1	0

1	1	0	1	1	0	1	0

Cyclic Shift Right Thrice CSR #3

1	0	1	1	0	1	1	0

1	1	0	1	0	1	1	0

Arithmetic Shifting: Sign bit remains the same, and the rest same as logical shifting.

Arithmetic Shift Left Once ASL # 1

1	0	1	1	0	1	1	0

1	1	1	0	1	1	0	0

Arithmetic Shift Right Once ASR # 1

1	0	1	1	0	1	1	0

1	0	0	1	1	0	1	1

Bit Manipulation In Monitoring And Control System

In monitoring and control system, each bit in a register or memory location can be used as a flag and would need to be tested, set, cleared separately.

- AND is used to check if the bit has been set.
- OR is used to set the bit.
- XOR is used to clear a bit that has been set.

- In assembly language we represent binary B10000001
- In assembly language we represent denary #32
- In assembly language we represent hexadecimal &F8

OR Bitwise Operation

Opcode	Operand	Explanation
OR	n	Bitwise OR operation of the content of ACC with operand

ACC	1	0	1	1	0	1	1	0
OR # 64	0	1	0	0	0	0	0	0
	1	1	1	1	0	1	1	0

Use of OR instruction in monitoring and control system.

Note: Whenever you want to set a bit, we use OR bitwise operation.

200 (Memory location)	1	0	1	1	0	1	1	0
					A3	A2	A1	flag

Question: Set Alarm 3 to ON without changing other bits.

Step 1: Write 1 below alarm 3 and the rest should be 0.
Step 2: OR that new binary value.

200 (Memory location)	1	0	1	1	0	1	1	0
	0	0	0	0	1	0	0	0
	0	0	0	0	1	0	0	0

Question: Write this is assembly language code ?

LDD 200
OR # 8
STO 200

(ii) The system receives a temperature reading of −2 from fish tank number 4.

Complete the bytes to show the values for this reading after it has been processed.

7	6	5	4	3	2	1	0

Byte 1

Byte 2

[2]

(d) A hardware device to affect the temperature of each tank is on or off depending on the value of a bit in memory location `6753`.

If bit 4 is 1, then the hardware device in fish tank 4 is on.

Write **assembly language** instructions to set bit 4 of memory location `6753` to 1 without changing any other bits. Use the instruction set provided.

LDD 6753

OR # 16

STO 6753

[3]

AND Bitwise Operation

0	0	0
0	1	0
1	0	0
1	1	1

Opcode	Operand	Explanation
AND	n	Bitwise AND operation of the content of ACC with operand

ACC	1	0	1	1	0	1	1	0
AND #32	0	0	1	0	0	0	0	0
	0	0	1	0	0	0	0	0

Note: If there was 1 in the place of 32, then after AND it would be the same as the operand.

Use of AND instruction in monitoring and control system:

- **AND is used to check if the bit was set (1) or 0.**
- **If you want to check, the AND is followed by a CMP instruction.**

Question: Check if bit 4 contains 1 or not.

Step 1: Put 1 below the bit you want to check.
Step 2: AND the new binary value with the value in ACC.
Step 3: Compare with the new value which you created.

If 1

AND #16	0	0	0	1	0	0	0	0
CMP #16	0	0	0	1	0	0	0	0

If true bit 4 is set

If 0

7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0

AND #16

0	0	0	1	0	0	0	0

0	0	0	0	0	0	0	0

Question: Write this in assembly language code ?

LDD 423
AND #16
CMP #16

(d) (i) The accumulator is loaded with the value of byte 1 from location 106.

Write the assembly language instruction to check whether the reading in byte 2 came from location 4.

```
LDD 106          // data loaded from address 106
```

AND #16

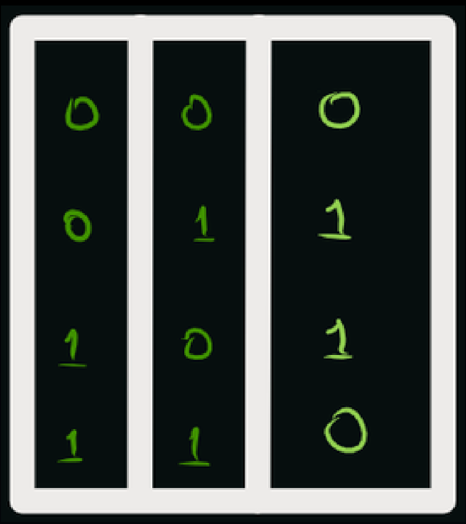
....[4]

(ii) Write the assembly language instruction to set the flag (bit 0) of the byte contained in the accumulator to 1.

OR # 1

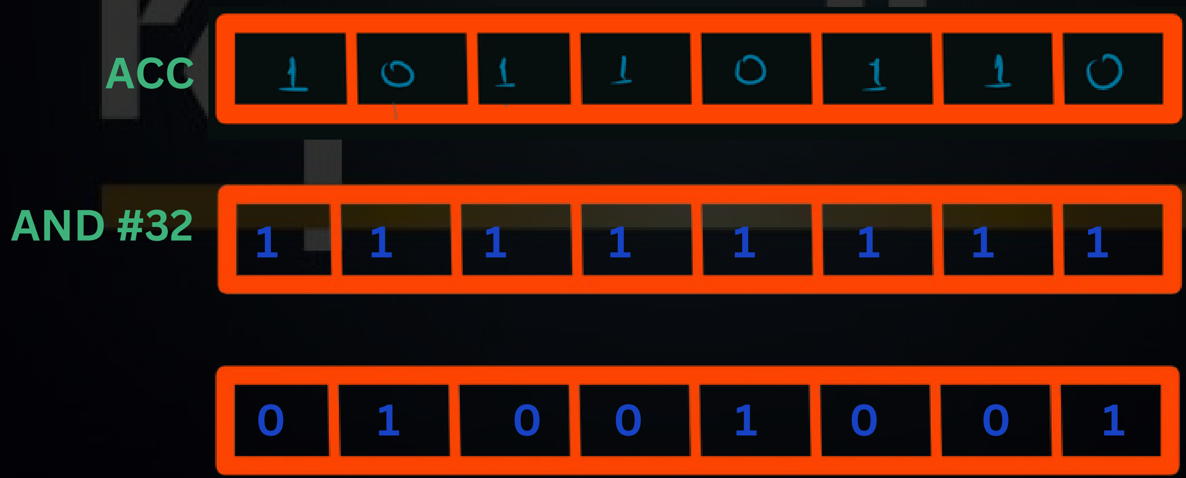
....[2]

XOR Bitwise Operation



Opcode	Operand	Explanation
XOR	n	Bitwise XOR operation of the content of ACC with operand

Note: Used to clear or inverse the values.



Clearing The BIt

7	6	5	4	3	2	1	0
1	0	1	1	0	1	1	0

Question: Clear bit 5

Step 1: Put 1 below the bit you want to clear.
Step 2: Apply XOR with the new value.

	7	6	5	4	3	2	1	0
ACC	1	0	1	1	0	1	1	0
XOR #32	0	0	1	0	0	0	0	0
	1	0	0	1	0	1	1	0

(b) Some bit manipulation instructions are shown in the table:

Instruction		Explanation
Opcode	Operand	
AND	#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	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	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 address or a symbolic address # denotes a denary number, e.g. #123		

The contents of the memory address 300 are shown:

Bit Number	7	6	5	4	3	2	1	0
300	0	1	1	0	0	1	1	0

(i) The contents of memory address 300 represent an unsigned binary integer.

Write the denary value of the unsigned binary integer in memory address 300.

.... [1]

(ii) An assembly language program needs to test if bit number 2 in memory address 300 is a 1.

Complete the assembly language instruction to perform this test.

AND
.... #4

[1]

(iii) An assembly language program needs to set bit numbers 4, 5, 6 and 7 to 0, but keep bits 0 to 3 with their existing values.

Write the assembly language instruction to perform this action.

AND #15
....

.... [2]

(b) The following table shows part of the instruction set for a processor. The processor has one general purpose register, 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
`STO`	`<address>`	Store the contents of ACC at 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>
`JPN`	`<address>`	Following a compare instruction, jump to <address> if the compare was False
`JMP`	`<address>`	Jump to the given address
`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
`XOR`	`#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>
`AND`	`#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>
`OR`	`#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 symbolic address # denotes a denary number, e.g. #123 B denotes a binary number, e.g. B01001101		

The current contents of main memory are shown:

Address	**Data**
100	00001111
101	11110000
102	01010101
103	11111111
104	00000000

Each row of the following table shows the current contents of ACC in binary and the instruction that will be performed on those contents.

Complete the table by writing the new contents of the ACC after the execution of each instruction.

Current contents of the ACC	Instruction	New contents of the ACC
11111111	OR 101	
00000000	XOR #15	
10101010	LSR #2	
01010101	AND 104	

[4]

(c) The table shows part of the instruction set for a processor. The processor has one general purpose register, the Accumulator (ACC).

Instruction		Explanation
Opcode	**Operand**	
AND	#n	Bitwise AND operation of the contents of ACC with the operand
XOR	#n	Bitwise XOR operation of the contents of ACC with the operand
OR	#n	Bitwise OR operation of the contents of ACC with the operand
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
# denotes a denary number, e.g. #123		

(i) Complete the register to show the result **after** the instruction `AND #2` is executed.

Register before:

0	1	1	0	1	1	0	1

Register after:

[1]

(ii) Complete the register to show the result **after** the instruction `OR #8` is executed.

Register before:

0	1	1	0	1	1	0	1

Register after:

[1]

(iii) Complete the register to show the result **after** the operation `LSL #4` is executed.

Register before:

0	1	1	0	1	1	0	1

Register after:

[1]

(a) The ACC currently contains the following positive binary integer:

0	1	1	0	0	1	0	1

Write the bit manipulation instruction that would change the binary integer in ACC to:

1	1	1	1	1	1	1	1

Opcode .... Operand ....

[2]

(b) The ACC currently contains the following positive binary integer:

0	1	1	0	0	1	0	1

Write the bit manipulation instruction that would change the binary integer in ACC to:

1	0	0	1	1	0	1	0

Opcode .... Operand ....

[2]

3(a)	**1 mark** for correct opcode and **1 mark** for corresponding operand OR #255 // OR #154 // XOR #154 e.g. • OR... • ... #255	**2**
3(b)	**1 mark** for correct opcode and **1 mark** for corresponding operand XOR #255 e.g. • XOR... • ... #255	**2**

Monitoring And Control System And Bit Manipulation

Question 1

(c) The temperature in her refrigerator must be kept between 4 and 6 degrees Celsius.

The microprocessor in the refrigerator turns on the cooling if the temperature is too high, and turns off the cooling if the temperature is too low.

Explain why the system in the refrigerator is a control and not a monitoring system.

....

....

....

....

....

.... [2]

Question 2

(c) **(i)** The Accumulator currently contains the binary number:

0	0	1	1	0	1	0	1

Write the contents of the Accumulator after the processor has executed the following instruction:

`LSL #2`

[1]

(ii) The Accumulator currently contains the binary number:

0	0	1	1	0	1	0	1

Identify the mathematical operation that the following instruction will perform on the contents of the accumulator.

```
LSR #3
```

....

.... [1]

Question 3

(b) Some bit manipulation instructions are shown in the table:

Instruction		Explanation
Opcode	**Operand**	
`AND`	`#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`	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`	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 address or a symbolic address # denotes a denary number, e.g. #123		

The contents of the memory address 300 are shown:

Bit Number	7	6	5	4	3	2	1	0
300	0	1	1	0	0	1	1	0

(i) The contents of memory address `300` represent an unsigned binary integer.

Write the denary value of the unsigned binary integer in memory address `300`.

.... [1]

(ii) An assembly language program needs to test if bit number `2` in memory address `300` is a `1`.

Complete the assembly language instruction to perform this test.

.... `#4`

[1]

(iii) An assembly language program needs to set bit numbers `4`, `5`, `6` and `7` to `0`, but keep bits `0` to `3` with their existing values.

Write the assembly language instruction to perform this action.

....

.... [2]

Question 4

(b) The following table shows part of the instruction set for a processor. The processor has one general purpose register, 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
STO	<address>	Store the contents of ACC at 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>
JPN	<address>	Following a compare instruction, jump to <address> if the compare was False
JMP	<address>	Jump to the given address
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
XOR	#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>
AND	#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>
OR	#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 symbolic address # denotes a denary number, e.g. #123 B denotes a binary number, e.g. B01001101		

The current contents of main memory are shown:

Address	Data
100	00001111
101	11110000
102	01010101
103	11111111
104	00000000

Each row of the following table shows the current contents of ACC in binary and the instruction that will be performed on those contents.

Complete the table by writing the new contents of the ACC after the execution of each instruction.

Current contents of the ACC	Instruction	New contents of the ACC
11111111	OR 101	
00000000	XOR #15	
10101010	LSR #2	
01010101	AND 104	

[4]

Question 5

(b) The following table shows part of the instruction set for a processor. The processor has one general purpose register, 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
STO	<address>	Store the contents of ACC at 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>
JPN	<address>	Following a compare instruction, jump to <address> if the compare was False
JMP	<address>	Jump to the given address
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
XOR	#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	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>
AND	#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>
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 symbolic address
# denotes a denary number, e.g. #123

The current contents of main memory are shown:

Address	Data
100	01010101
101	11110000
102	00001111
103	00000000
104	11111111

(i) In the following table, each row shows the current contents of the ACC in binary and the instruction that will be performed on those contents.

Complete the table by writing the new contents of the ACC after the execution of each instruction.

Current contents of the ACC	Instruction	New contents of the ACC
01010101	XOR 101	
11110000	AND 104	
00001111	LSL #4	
11111111	OR 102	

[4]

Question 6

(c) The table shows part of the instruction set for a processor. The processor has one general purpose register, the Accumulator (ACC).

Instruction		Explanation
Opcode	**Operand**	
AND	#n	Bitwise AND operation of the contents of ACC with the operand
XOR	#n	Bitwise XOR operation of the contents of ACC with the operand
OR	#n	Bitwise OR operation of the contents of ACC with the operand
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
# denotes a denary number, e.g. #123		

(i) Complete the register to show the result **after** the instruction `AND #2` is executed.

Register before:

0	1	1	0	1	1	0	1

Register after:

[1]

(ii) Complete the register to show the result **after** the instruction `OR #8` is executed.

Register before:

0	1	1	0	1	1	0	1

Register after:

[1]

(iii) Complete the register to show the result **after** the operation `LSL #4` is executed.

Register before:

0	1	1	0	1	1	0	1

Register after:

[1]

Question 7

3 The table shows part of the instruction set for a processor. The processor has one general purpose register, the Accumulator (ACC).

Instruction		Explanation
Opcode	**Operand**	
`AND`	`#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`	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`	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		

(a) The ACC currently contains the following positive binary integer:

0	1	1	0	0	1	0	1

Write the bit manipulation instruction that would change the binary integer in ACC to:

1	1	1	1	1	1	1	1

Opcode .... Operand ....

[2]

(b) The ACC currently contains the following positive binary integer:

0	1	1	0	0	1	0	1

Write the bit manipulation instruction that would change the binary integer in ACC to:

1	0	0	1	1	0	1	0

Opcode .... Operand ....

[2]

Question 8

(c) The table shows part of the instruction set for a processor. The processor has one general purpose register, the Accumulator (ACC).

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
OR	Bn	Bitwise OR operation of the contents of ACC with the operand
B denotes a binary number, e.g. B01001010		

The binary value 00111101 is stored in the memory address 200.

Each instruction in the diagram is performed on the data in memory address 200.

Draw **one** line from each instruction to its correct result.

Instruction	**Result**
	01111101
XOR B11110000	00111101
OR B01010101	11111111
AND B11111111	11000010
	11001101

[3]

Question 9

(b) The following table shows another part of the instruction set for the processor.

Instruction		Explanation
Opcode	**Operand**	
`AND`	`#n`	Bitwise AND operation of the contents of ACC with the operand
`AND`	`Bn`	Bitwise AND operation of the contents of ACC with the binary number n
`XOR`	`#n`	Bitwise XOR operation of the contents of ACC with the operand
`XOR`	`Bn`	Bitwise XOR operation of the contents of ACC with the binary number n
`OR`	`#n`	Bitwise OR operation of the contents of ACC with the operand
`OR`	`Bn`	Bitwise OR operation of the contents of ACC with the binary number n
`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. B01001101		

(i) The current contents of the ACC are:

1	0	0	1	0	0	1	1

Show the result after the execution of the following instruction.

`XOR B00011111`

....

....

[1]

(ii) The current contents of the ACC are:

1	0	0	1	0	0	1	1

Show the result after the execution of the following instruction.

`AND B11110000`

....

....

[1]

(iii) The current contents of the ACC are:

1	0	0	1	0	0	1	1

Show the result after the execution of the following instruction.

`OR B11001100`

....

....

[1]

(iv) The current contents of the ACC are:

1	0	0	1	0	0	1	1

Show the result after the execution of the following instruction.

`LSR #2`

....

....

[1]

Question 10

(b) The following table shows another part of the instruction set for the processor.

Instruction		Explanation
Opcode	**Operand**	
AND	#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	Bitwise XOR operation of the contents of ACC with the operand
XOR	Bn	Bitwise XOR operation of the contents of ACC with the binary number n
XOR	<address>	Bitwise XOR operation of the contents of ACC with the contents of <address>
OR	#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 symbolic address # denotes a denary number, e.g. #123 B denotes a binary number, e.g. B01001101		

The contents of memory addresses 50 and 51 are shown:

Memory address	Data value
50	01001101
51	10001111

(i) The current contents of the ACC are:

0	1	0	1	0	0	1	1

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

`XOR B00011111`

[1]

(ii) The current contents of the ACC are:

0	1	0	1	0	0	1	1

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

```
AND 50
```

....

....

[1]

(iii) The current contents of the ACC are:

0	1	0	1	0	0	1	1

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

```
LSL #3
```

....

....

[1]

(iv) The current contents of the ACC are:

0	1	0	1	0	0	1	1

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

```
OR 51
```

....

....

[1]

Question 11

(b) The following table shows another part of the instruction set for the processor.

Instruction		Explanation
Opcode	**Operand**	
AND	#n	Bitwise AND operation of the contents of ACC with the operand
AND	Bn	Bitwise AND operation of the contents of ACC with the binary number n
AND	<address>	Bitwise AND operation of the contents of ACC with the contents of <address>
XOR	#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	Bitwise OR operation of the contents of ACC with the operand
OR	Bn	Bitwise OR operation of the contents of ACC with the binary number n
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. B01001101		

(i) The current contents of the ACC are:

0	0	1	1	0	1	1	0

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

```
AND B01001100
```

....

....

[1]

(ii) The current contents of the ACC are:

1	0	0	1	0	1	0	1

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

```
OR B01001111
```

....

....

[1]

(iii) The current contents of the ACC are:

1	0	0	1	1	1	0	1

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

```
LSR #2
```

....

....

[1]

Question 12

10 **(a)** Explain the importance of feedback in a control system.

....

....

....

....

....

.... [3]

(b) **(i)** Identify **one** sensor that could be used in a car alarm system.

Justify your choice.

Sensor ....

Justification ....

....

....

....

[2]

Question 13

(vi) Show the result of a 3-place right logical shift on the binary number:

11001100

.... [1]

(d) Show the result of a 2-place left logical shift on the binary number:

01001111

.... [1]

Question 14

(c) 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]

Question 15

9 **(a)** Explain the importance of feedback in a control system.

....

....

....

.... [2]

Question 16

1 A factory makes chocolate bars.

The factory uses a conveyor belt that moves the products from one stage of production to the next stage.

(a) An automated system counts the number of chocolate bars made at the end of production.

The system includes a sensor positioned above the conveyor belt.

Identify **one** appropriate type of sensor that can be used.

.... [1]

(b) A second automated system removes chocolate bars with an incorrect weight from the production line.

Describe the role of an **actuator** in this second system.

....

....

....

.... [2]

Question 17

(b) 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	

[3]

Question 18

(b) 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`	

[3]

Question 19

(b) 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.

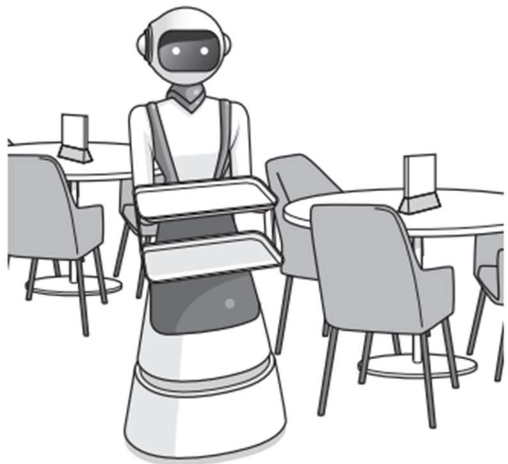
The binary number 11111111 is reloaded into the ACC before each instruction is run.

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

[3]

Question 20

7 Robots are used to serve food and drink to customers at a restaurant.



(a) A robot navigates through the restaurant to the table it is serving.

Complete the table by identifying **two** sensors that can be included in the robot **and** the purpose of each sensor in the navigation system.

Sensor	Purpose of sensor in navigation system
....	....
....	....

[2]

(c) The navigation system can be considered an example of a control system.

Describe how feedback is used in a control system.

....

....

....

.... [2]

Answer

Answer 1

5(c)	**1 mark** per bullet point • The system uses feedback • The system causes the temperature to change // produces an action	2

Answer 2

3(c)(i)	1 | 1 | 0 | 1 | 0 | 1 | 0 | 0	1
3(c)(ii)	**1 mark** for correct answer The number is divided by 8 (and only whole number retained)	1

Answer 3

4(b)(i)	102	1
4(b)(ii)	AND	1

4(b)(iii)	**1 mark** for AND, **1 mark** for #15 AND #15	2

Answer 4

6(b) — **1 mark** for each correct row — 4

Current contents of the ACC	Instruction	New contents of the ACC
11111111	OR 101	**11111111**
00000000	XOR #15	**00001111**
10101010	LSR #2	**00101010**
01010101	AND 104	**00000000**

Answer 5

8(b)(i)	**1 mark** for each correct answer	**4**

Current contents of the ACC	Instruction	New contents of the ACC
01010101	XOR 101	**1010 0101**
11110000	AND 104	**1111 0000**
00001111	LSL #4	**1111 0000**
11111111	OR 102	**1111 1111**

Answer 6

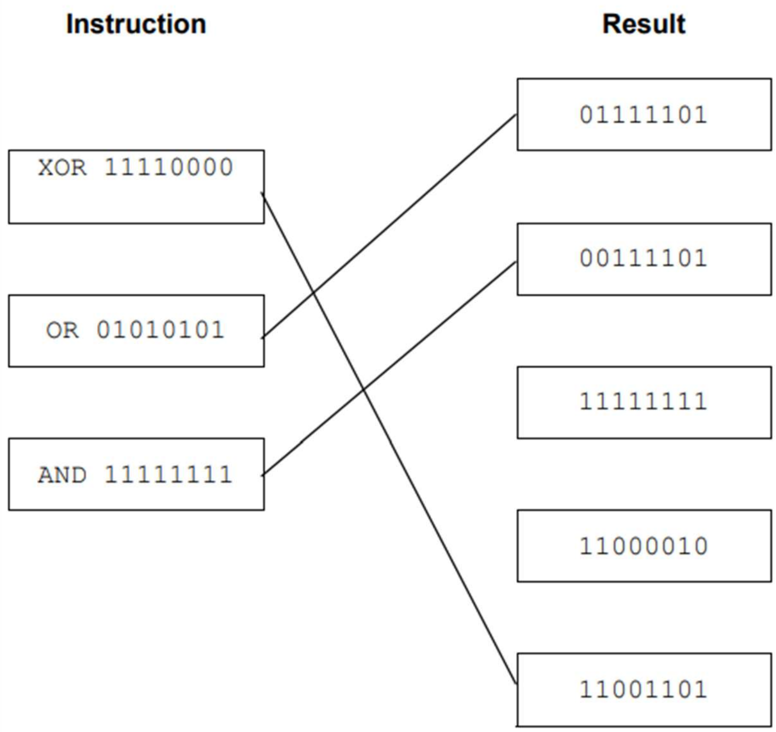
6(c)(i)	0000 0000	**1**
6(c)(ii)	0110 1101	**1**
6(c)(iii)	1101 0000	**1**

Answer 7

3(a)	**1 mark** for correct opcode and **1 mark** for corresponding operand OR #255 // OR #154 // XOR #154 e.g. • OR... • ... #255	**2**
3(b)	**1 mark** for correct opcode and **1 mark** for corresponding operand XOR #255 e.g. • XOR... • ... #255	**2**

Answer 8

3(c)	**1 mark** for each correct line	**3**



Answer 9

6(b)(i)	1000 1100	**1**
6(b)(ii)	1001 0000	**1**
6(b)(iii)	1101 1111	**1**
6(b)(iv)	0010 0100	**1**

Answer 10

7(b)(i)	0100 1100	**1**
7(b)(ii)	0100 0001	**1**
7(b)(iii)	1001 1000	**1**
7(b)(iv)	1101 1111	**1**

Answer 11

6(b)(i)	0000 0100	**1**
6(b)(ii)	1101 1111	**1**
6(b)(iii)	0010 0111	**1**

Answer 12

10(a)	**1 mark** for each bullet point: • to ensure the system operates with the given criteria • ... by enabling system output to affect subsequent system input • ... thus allowing conditions to be automatically adjusted	**3**
10(b)(i)	**1 mark** for identification of a suitable sensor **1 mark** for corresponding justification Example: • sound sensor • if a sound occurs inside the car the alarm is activated • infra-red sensor • senses the heat of person in the car / infra-red beams are broken • pressure sensor • an intruder sits in the driver's seat	**2**

Answer 13

3(d)(vi)	00011001	**1**
4(d)	0011 1100	**1**

Answer 14

8(c)(i)	0000 0101	**1**
8(c)(ii)	0000 0010	**1**
8(c)(iii)	1101 0010	**1**

Answer 15

9(a)	**One mark** for each bullet point (**max 2**) • Feedback ensures that a system operates within set criteria / constraints • ...by enabling system output to affect (subsequent) system input • ...thus allowing conditions to be automatically adjusted	2

Answer 16

1(a)	**1 mark** for: • infra-red / proximity (sensor)	1
1(b)	**1 mark** for each bullet point (**max 2**) • Actuator generates a signal / causes an action / converts electrical energy into a mechanical force • ... to push an arm // to open a trap door // to pick up the chocolate bar with the incorrect weight	2

Answer 17

4(b)	**1 mark** for each correct answer:	3

Program Number	Code	ACC Content
1	`AND 31`	**1001 1010**
2	`XOR B01001111`	**1101 0101**
3	`OR #30`	**1001 1110**

Answer 18

5(b)	**1 mark for** each correct answer:	3

Program Number	Code	ACC Content
1	XOR 29	**0100 1001**
2	AND #29	**0000 0100**
3	OR B11111111	**1111 1111**

Answer 19

3(b)	**1 mark** for each correct answer:	3

Instruction Number	Instruction	ACC Content
1	`LSL #2`	**1111 1100**
2	`XOR 100`	**1111 0010**
3	`AND 103`	**0011 0111**

Answer 20

7(a)	**1 mark** for sensor **and** matching purpose to **max 2**:	2

Sensor	Purpose of sensor in navigation system
Pressure	To detect if a table or other obstacle has been hit // to detect when food is put on/taken off the tray so it can move on
Infra-red	To detect if there is an obstacle in the way // to indicate that it has reached the desired table
Sound	To detect if someone is speaking so that it can use AI to decipher the speech and whether the robot is required to stop

7(c)	**1 mark** each to **max 2**:	2

- Feedback ensures that a system operates within set criteria / constraints
- ... by enabling system output to affect subsequent system input
- ... thus allowing conditions to be **automatically** adjusted