CHAPTER 4: ACTIVITIES

ACTIVITY 4a

1 a) Describe the functions of the following registers.

i) Current instruction register

ii) Memory address register

iii) Program counter

b) Status registers contain flags. Three such flags are named N, C and V.

i) What does each of the three flags represent?

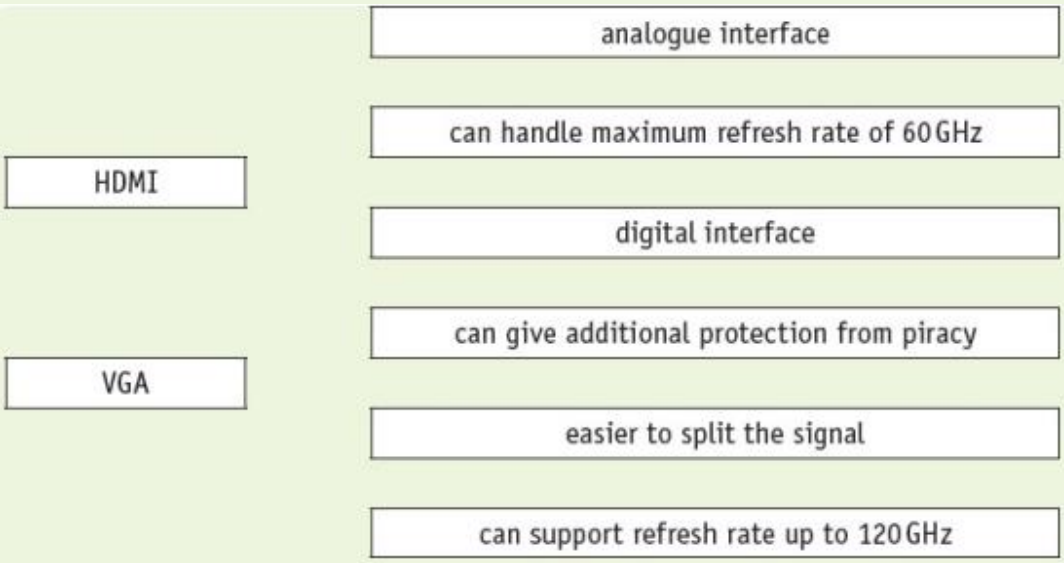
ii) Give an example of the use of each of the three flags.

2 a) Name **three** buses used in the Von Neumann architecture.

b) Describe the function of each named bus.

c) Describe how bus width and clock speed can affect computer performance.

3 Copy the diagram below and connect each feature to the correct port, HDMI or VGA.



4 a) What is meant by the *fetch-execute cycle*?

b) Using register transfer notation, show the main stages in a typical fetch-execute cycle.

5 Copy and complete this paragraph by using terms from this chapter.

The processor_____data and instructions required for an application and temporarily stores them in the_____until they can be processed.

The_____is used to hold the address of the next instruction to be executed. This address is copied to the_____using the_____.

The contents at this address are stored in the_____.

Each instruction is then_____and finally_____sending out_____using the_____. Any calculations carried out are done using the_____. During any calculations, data is temporarily held in a special register known as the_____.

Activity 4A

1 a) i) CIR – stores the current instructions being executed.

ii) MAR – stores the address of the memory location which is about to be accessed.

iii) PC – stores the address of the next instruction to be executed.

b) i) N = negative flag; set to 1 if result of calculation is negative

C = carry flag; set to 1 if there is a carry bit following a calculation

V = overflow flag; set to 1 if there is an overflow bit following a calculation

ii) two positive numbers added together give a negative result:

```
  01110001
+ 01011110        N = 1
  11001111
```

when two bytes are added together an arithmetic carry bit is generated from the most significant bit position:

(Note: the carry bit is used for **unsigned** integers)

```
  10000110
+ 11001011        C = 1
 101010001
```

overflow occurs when 7 bits are used for the binary number and the 8th bit is a sign bit; if +127 is the largest integer which can be stored then a sum > 127 will cause overflow:

```
  01110001
+ 01011110        V = 1
  11001111
```

2 a) address bus, data bus, control bus

b)

Address bus

- carries addresses to memory controller, thus identifying memory location which is to be read or written to
- bus is unidirectional between CPU and memory thus preventing an address being carried back to the CPU.

Data bus

- bidirectional bus which carries data (address, instruction or numerical value) throughout the processor
- data can be carried from CPU to memory (and vice versa) and to/from input/output devices.

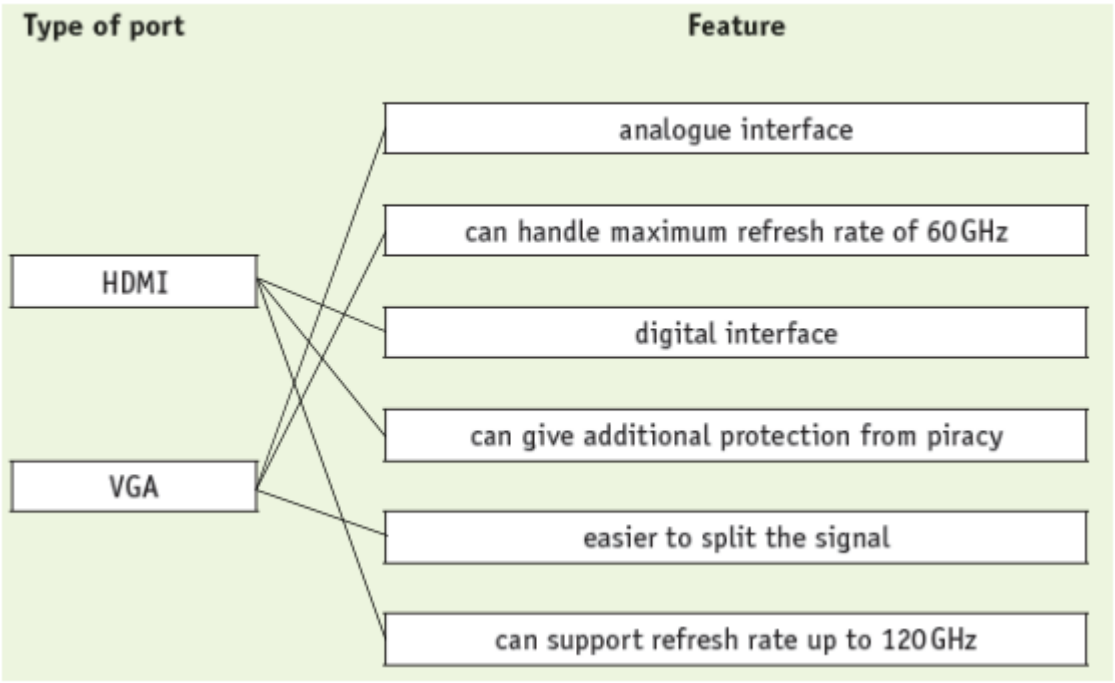
Control bus

- bidirectional bus which transmits signals from the control unit to other components of the computer
- this includes clock signals used to synchronise all operations.

c)

- In general, a higher clock speed increases the performance of a computer.
- Using a wider address bus and data bus allows a larger range of memory locations to be directly accessed or allows a larger word size to be handled by the computer.
- Both of the above speed up computer operations.
- However, it is not a simple matter of just increasing clock speed since …
- … If the processor doesn't use wider buses (etc.), increasing the clock speed alone could lead to problems such as overheating and non-synchronisation of operations.
- Risk of overclocking can occur.

3



4 a) **Fetch-execute cycle**

- is the basic operational process of a computer system
- is the process whereby computer fetches (retrieves) a program instruction from memory and determines what the instruction 'means'
- once decoded, the instruction is executed
- makes use of buses and addresses to carry out the various functions/operations.

b)

MAR	⟵	[PC]	(contents of PC copied into MAR)
PC	⟵	[PC] + 1	(PC is incremented by 1)
MDR	⟵	[[MAR]]	(data stored at address shown in MAR is copied into MDR)
CIR	⟵	[MDR]	(contents of MDR copied into CIR)

5 fetches, immediate access store (IAS), program counter (PC), MAR, address bus, MDR, decoded, executed, control signals, control bus, ALU, accumulator

ACTIVITY 4B

1 a) State the contents of the accumulator after the following instructions have been executed. The memory location with address 200 contains 300, the memory location with address 300 contains 50.

i) LDM #200

ii) LDD 200

iii) LDI 200

b) Write an assembly language instruction to:

i) compare the accumulator with 5

ii) jump to address 100 if the comparison is true.

2 a) Copy and complete the symbol table for this assembly language program. Assume that the translated program will start at memory address 100.

b) Complete a trace table to show the execution of this assembly language program.

c) State the task that this assembly language program performs.

```
Label   Opcode  Operand
        LDD     number1
        SUB     number2
        ADD     number3
        CMP     #10
        JPE     nomore
        ADD     number4
nomore: STO     total
        END

number1: #30
number2: #40
number3: #20
number4: #50
  total: #0
```

3 a) Using the assembly language instructions given in this section, write an assembly language program to output the ASCII value of each element of an array of four elements.

b) Complete the symbol table for your assembly language program. Assume that the translated program will start at memory address 100.

c) Complete a trace table to show the execution of your assembly language program.

Activity 4B

1 a) i) 200
ii) 300
(iii) 50

b) i) `CMP #5`
ii) `JPE 100`

2 a)

Label	Address
nomore	106
Number1	109
Number2	110
Number3	111
Number4	112
total	113

b)

CIR	Opcode	Operand	ACC	**total** 113
100	LDD	number1	30	0
101	SUB	number2	-10	0
102	ADD	number3	10	0
103	CMP	#10	10	0
104	JPE	nomore	10	0
106	STO	total	10	10
107	END		10	10

c) Subtracts `number2` from `number1`, then add `number3`. Only add `number4` if the result of the first calculation is not equal to 10. Store the answer in total

3 a)

Label	Opcode	Operand	Comment
	LDM	#0	Load 0 into ACC
	STO	counter	Store 0 in counter
	LDR	#0	Set IX to 0
loop:	LDX	array	Load the element of the array indexed by IX into ACC
	OUT		Output the ASCII character
	INC	IX	Add 1 to the contents of IX
	LDD	counter	Load counter into ACC
	INC	ACC	Add 1 to ACC
	STO	counter	Store result in counter
	CMP	#4	Compare with 4
	JPN	loop	If ACC not equal to 4 then return to start of loop
	END		
array:	#67		Array of 4 ASCII characters
	#79		
	#68		
	#69		
counter:			counter for loop

b)

Label	Address
loop	103
array	113
counter	117

c)

CIR	Opcode	Operand	ACC	IX	counter	Output
100	LDM	#0	0			
101	STO	counter	0			
102	LDR	#0	0	0	0	
103	LDX	array	67	0	0	
104	OUT		67	0	0	C
105	INC	IX	67	1	0	
106	LDD	counter	0	1	0	
107	INC	ACC	1	1	0	
108	STO	counter	1	1	1	
109	CMP	#4	1	1	1	
110	JPN	loop	1	1	1	
103	LDX	array	79	1	1	
104	OUT		79	1	1	O
105	INC	IX	79	2	1	
106	LDD	counter	1	2	1	
107	INC	ACC	3	3	2	
108	STO	counter	3	3	3	
109	CMP	#4	3	3	3	
110	JPN	loop	3	3	3	
103	LDX	array	69	3	3	
104	OUT		69	3	3	E
105	INC	IX	69	4	3	
106	LDD	counter	3	4	3	
107	INC	ACC	4	4	3	
108	STO	counter	4	4	4	
109	CMP	#4	4	4	4	
110	JPN	loop	4	4	4	
111	END	counter	2	2	2	

ACTIVITY 4C

1 a) State the contents of the accumulator after the following instructions have been executed. The accumulator contains B00011001.

i) LSL #4

ii) LSR #5

b) Write an assembly language instruction to:

i) set bit 4 in the accumulator

ii) clear bit 1 in the accumulator.

2 a) Describe the difference between *arithmetic shifts* and *logical shifts*.

b) Explain, with the aid of examples, how a *cyclic shift* works.

c) This register is shown before and after it has been shifted. Identify the type of shift that has taken place.

0	0	1	1	0	1	0	1

1	0	1	0	1	0	0	0

Activity 4C

1 a) i) `B10010000`

ii) `B00000000`

b) i) `OR #B1000`

ii) `XOR #B1`

2 a) Arithmetic shifts preserve the sign bit of the number and logical shifts always fill with zeros.

b) No bits are lost during a shift, bits shifted out of one end of the register are introduced at the other end of the register, for example an 8-bit register containing the binary value 10101111 shifted left cyclically 3 places would become 01111101.

c) Shift left logical 3.

CHAPTER 4: QUESTIONS

1 a) Write these six stages of the Von Neumann fetch-execute cycle in the correct order.

[6]

- instruction is copied from the MDR and is placed in the CIR
- the instruction is executed
- the instruction is decoded
- the address contained in PC is copied to the MAR
- the value in PC is incremented by 1
- instruction is copied from memory location in MAR and placed in MDR

b) Explain how the following affect the performance of a computer system.

i) Width of the data bus and address bus.

[2]

ii) The clock speed.

[2]

iii) Use of dual core or quad core processors.

[2]

c) A student accessed the BIOS on their computer. They increased the clock speed from 2.5 GHz to 3.2 GHz.
Explain the potential dangers in doing this.

2 a) Explain the main differences between HDMI, VGA and USB ports when sending data to peripherals.

[5]

b) Describe how interrupts can be used to service a printer printing out a large 1000 page document.

[5]

3 a) i) Name **three** special registers used in a typical processor.

[3]

ii) Explain the purpose of the three registers named in part i).

[3]

b) Explain how interrupts are used when a processor sends a document to a printer.

[4]

4 A programmer is writing a program in assembly language. They need to use shift instructions.
Describe, using examples, three types of shift instructions the programmer could use.

[6]

5 An intruder detection system for a large house has four sensors. An 8-bit memory location stores the output from each sensor in its own bit position.
The bit value for each sensor shows:

- 1 – the sensor has been triggered
- 0 – the sensor has not been triggered

The bit positions are used as follows:

Not used				Sensor 4	Sensor 3	Sensor 2	Sensor 1

The output from the intruder detection system is a loud alarm.

a) i) State the name of the type of system to which intruder detection systems belong.

[1]

ii) Justify your answer to part i).

[1]

b) Name **two** sensors that could be used in this intruder detection system.
Give a reason for your choice.

[4]

c) The intruder system is set up so that the alarm will only sound if two or more sensors have been triggered. An assembly language program has been written to process the contents of the memory location.
This table shows part of the instruction set for the processor used.

Instruction		**Explanation**
Opcode	**Operand**	
LDD	<address>	Direct addressing. Load the contents of 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)
ADD	<address>	Add the contents of the given address to the contents of ACC
AND	<address>	Bitwise AND operation of the contents of ACC with the contents of <address>
CMP	#n	Compare the contents of ACC with the number n
JMP	<address>	Jump to the given address
JPE	<address>	Following a compare instruction, jump to <address> if the compare was True
JGT	<address>	Following a compare instruction, jump to <address> if the content of ACC is greater than the number used in the compare instruction
END		End the program and return to the operating system

Part of the assembly code is:

	Opcode	Operand
SENSORS:		B00001010
COUNT:		0
VALUE:		1
LOOP:	LDD	SENSORS
	AND	VALUE
	CMP	#0
	JPE	ZERO
	LDD	COUNT
	INC	ACC
	STO	COUNT
ZERO:	LDD	VALUE
	CMP	#8
	JPE	EXIT
	ADD	VALUE
	STO	VALUE
	JMP	LOOP
EXIT:	LDD	COUNT
TEST:	CMP	…
	JGT	ALARM

i) Copy the table below and dry run the assembly language code.
Start at LOOP and finish when EXIT is reached.

[4]

BITREG	COUNT	VALUE	ACC
B00001010	0	1	

ii) The operand for the instruction labelled TEST is missing.
State the missing operand.

[1]

iii) The intruder detection system is improved and now has eight sensors. One instruction in the assembly language code will need to be amended.
Identify this instruction. Write the amended instruction.

[2]

CHAPTER 4: ANSWERS

1 a)

Stage	Order
Instruction is copied from the MDR and is placed in the CIR	3
Instruction is executed	6
Instruction is decoded	5
Address contained in PC is copied to the MAR	1
Value in PC is incremented by 1	4
Instruction is copied from memory location in MAR and placed in MDR	2

b) i) **Width of the data bus and address bus**

- determines number of bits that can be simultaneously transferred
- hence wider bus improves processing speed as fewer transfers are needed
- double bus width = 2 × data transferred per clock pulse.

ii) **Clock speed**

- determines number of cycles computer can execute per second
- increasing clock speed increases number of operations per unit time
- limited by heat generated by higher clock speeds.

iii) **Dual or quad core**

- each CPU contains 2 cores or 4 cores
- CPU with 2 or 4 processors in the same integrated circuit; each processor has its own cache and controller
- single computing component with 2/4 independent processing units (cores) which can read and execute program instructions
- this gives processor double or four times the processing power of a single core processor
- however, this isn't the case in reality since the CPU needs to communicate with each core which reduces overall performance
- software frequently can't take advantage of 2 or 4 core processors again reducing potential overall performance.

c) Dangers are overheating and risk of going out of synchronisation causing errors and potential computer 'crash' due to instructions no longer correctly synchronized.

2 a **HDMI**

- allows output (audio and visual) from a computer to be connected to HDMI-enabled monitor/tv
- supports high definition and enhanced signals
- it is a digital system which can support high definition televisions which require more data and at a faster data transfer rate
- can also protect against piracy by using authentication protocols
- supports a refresh rate of 120 Hz.

VGA

- supports 640×480 resolution with refresh rate of 60 Hz
- analogue system which makes it easier to split the signals between more than one device.

USB

- allows computers to communicate with peripherals
- uses 4-wire shielded cables
- devices are automatically detected when plugged into computer USB port
- serial data transmission
- industry standard to connect devices to a computer.

b **Interrupts**

- data sent to printer buffer from computer
- contents of buffer sent to printer and document starts to be printed
- processor carries on with other tasks while printing continues in background
- if printer runs out of paper, runs out of ink, paper jam (etc.) then it sends out an interrupt signal
- if interrupt sent out, message is displayed on computer screen requesting user to resolve issue
- once all the data from the buffer is printed, printer sends an interrupt signal to the processor requesting more data
- depending on its priority, interrupt is serviced
- once interrupt is serviced, more data is sent to the printer buffer and the above stages are repeated until all 1000 pages are printed out.

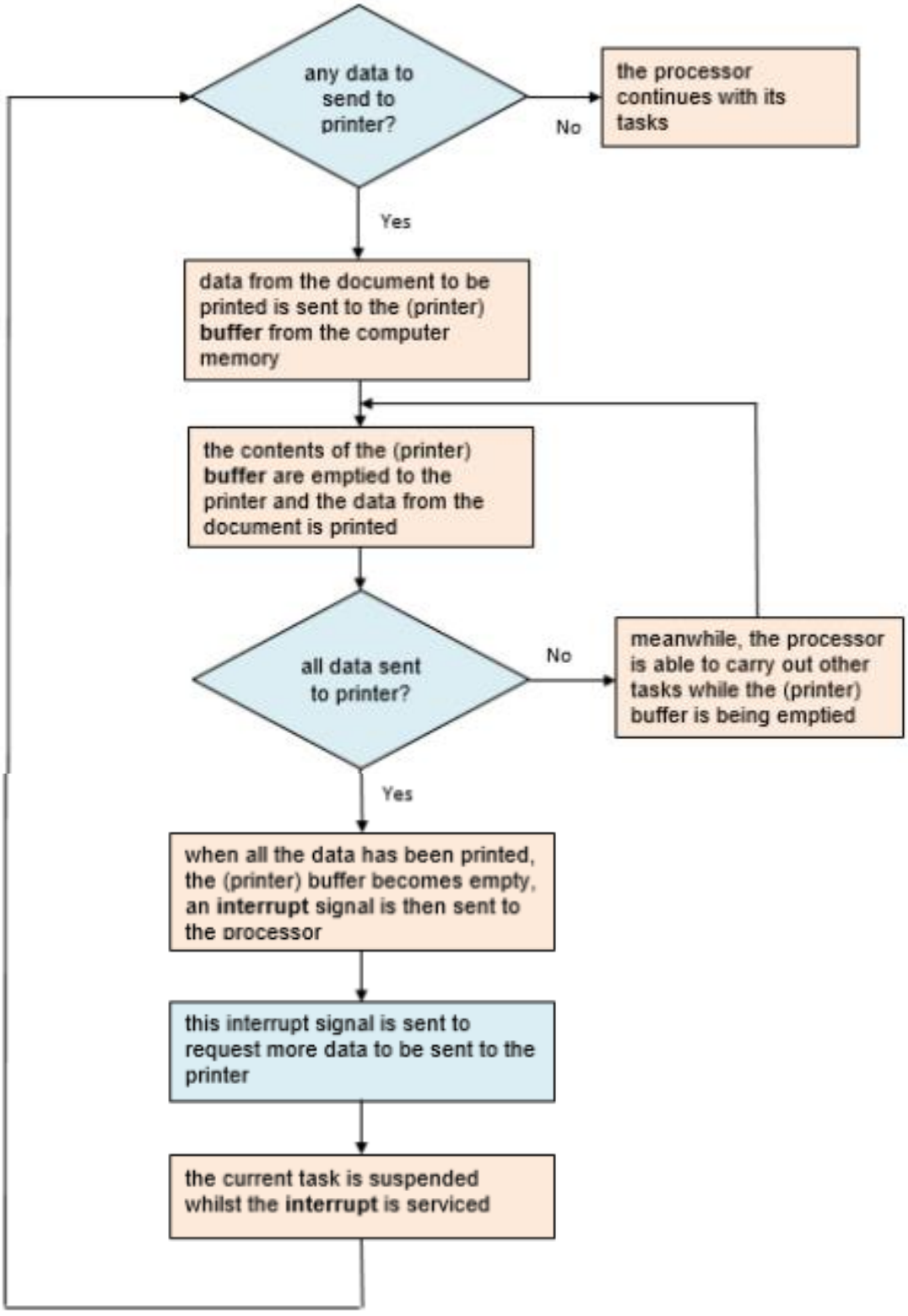
3 a i) ii) PC – stores address of next instruction to be executed

MDR – stores data in transit between memory and other registers

CIR – stores current instruction being executed

MAR – stores address of memory location which is about to be accessed

b


any data to send to printer?
No
the processor continues with its tasks
Yes
data from the document to be printed is sent to the (printer) buffer from the computer memory
the contents of the (printer) buffer are emptied to the printer and the data from the document is printed
all data sent to printer?
No
meanwhile, the processor is able to carry out other tasks while the (printer) buffer is being emptied
Yes
when all the data has been printed, the (printer) buffer becomes empty, an interrupt signal is then sent to the processor
this interrupt signal is sent to request more data to be sent to the printer
the current task is suspended whilst the interrupt is serviced

4 **Logical shift** – bits shifted out of the register are replaced with zeros, for example an 8-bit register containing the binary value 10101111 shifted left logically 3 places would become 01111000

Arithmetic shift – the sign of the number is preserved, for example an 8-bit register containing the binary value 10101111 shifted right arithmetically 3 places would become 11110101. Arithmetic shifts can be used for multiplication or division by powers of two.

Cyclic shift – no bits are lost during a shift, bits shifted out of one end of the register are introduced at the other end of the register, for example an 8-bit register containing the binary value 10101111 shifted left cyclically 3 places would become 01111101

5

<table>
<tr><td>a) i)</td><td colspan="5">monitoring system</td></tr>
<tr><td>ii)</td><td colspan="5">there is no ‘control’ taking place</td></tr>
<tr><td>b)</td><td colspan="5">Pressure …If intruder steps on sensor, Infra-red …If beam cut by intruder</td></tr>
<tr><td>c) i)</td><td></td><td>SENSORS</td><td>COUNT</td><td>VALUE</td><td>ACC</td></tr>
<tr><td></td><td></td><td>B00001010</td><td>0</td><td>1</td><td>B00001010</td></tr>
<tr><td></td><td></td><td></td><td></td><td></td><td>B00000000</td></tr>
<tr><td></td><td></td><td></td><td></td><td></td><td>1</td></tr>
<tr><td></td><td></td><td></td><td></td><td>2</td><td>2</td></tr>
<tr><td></td><td></td><td></td><td></td><td></td><td>B00001010</td></tr>
<tr><td></td><td></td><td></td><td></td><td></td><td>B00000010</td></tr>
<tr><td></td><td></td><td></td><td></td><td></td><td>0</td></tr>
<tr><td></td><td></td><td></td><td>1</td><td></td><td>1</td></tr>
<tr><td></td><td></td><td></td><td></td><td></td><td>2</td></tr>
<tr><td></td><td></td><td></td><td></td><td>4</td><td>4</td></tr>
<tr><td></td><td></td><td></td><td></td><td></td><td>B00001010</td></tr>
<tr><td></td><td></td><td></td><td></td><td></td><td>B00000000</td></tr>
<tr><td></td><td></td><td></td><td></td><td></td><td>4</td></tr>
<tr><td></td><td></td><td></td><td></td><td>8</td><td>8</td></tr>
<tr><td></td><td></td><td></td><td></td><td></td><td>B00001010</td></tr>
<tr><td></td><td></td><td></td><td></td><td></td><td>B00001000</td></tr>
<tr><td></td><td></td><td></td><td></td><td></td><td>1</td></tr>
<tr><td></td><td></td><td></td><td>2</td><td></td><td>2</td></tr>
<tr><td></td><td></td><td></td><td></td><td></td><td>8</td></tr>
<tr><td>c) ii)</td><td colspan="5">#1</td></tr>
<tr><td>c) iii)</td><td colspan="5">CMP #8 instruction
CMP #128</td></tr>
</table>