Mark schemes

**Q1.**

**3 marks are for AO2 (apply)**

**1 mark** for getting Part 1 **or** Part 2 correct on any of the three diagrams.

**1 mark** for getting **corresponding** Part 1 **or** Part 2 correct on the same diagram.

**1 mark** for getting **corresponding** Part 3 correct on the same diagram.

**MAX 2** if not fully correct

Mark response against diagram that will give the highest mark.

Mark point 3 can only be awarded if at least one other mark point has been awarded.

Alternative Diagram 1



Alternative Diagram 2



Alternative Diagram 3



**[3]**

**Q2.**

**Mark is for AO1 (knowledge)**

NOR;

**[1]**

**Q3.**

**Mark is for AO1 (knowledge)**

NAND;

**A**. NOT AND

**[1]**

**Q4.**

**Marks are for AO2 (apply)**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|   |   | **1** | **2** | **3** | **4** | **5** |
| **A** | **B** |  | **A +**  |  |  • **B** |  |
| 0 | 0 | 1 | 1 | 1 | 0 | 1 |
| 0 | 1 | 0 | 0 | 1 | 1 | 0 |
| 1 | 0 | 1 | 1 | 0 | 0 | 1 |
| 1 | 1 | 0 | 1 | 0 | 0 | 1 |

**Mark as follows:**

**1 mark** for column 1 and 3 correct

**1 mark** for column 4 correct

**1 mark** for columns 2 and 5 correct and identical

**I**. order of columns

**[3]**

**Q5.**

**4 marks are for AO2 (apply)**

**Marking guidance for examiners**

•   Award marks for working out until an incorrect step has been made. If a student misses out some steps but does not make an error then continue marking.

•   If, in any one step, a candidate is simplifying different parts of an expression simultaneously award all relevant marks for this multiple stage but don’t award any further marks for working in any parts simplified incorrectly. Example, if the expression P.P.(P+Q) + P.P.1 was changed to P.(P+Q)+P.0, the candidate would get one mark for simplifying the first part to P.(P+Q) and could get further marks for correctly simplifying this part of the expression further but should not be awarded marks for simplifying the incorrectly changed part P.0 (ie to 0)

**Mark as follows**

**1 mark** for final answer A

**Max 3 marks for working:**

•   **1 mark** for each application of an identity or theorem other than cancelling NOTs that produces a simpler expression.

•   **1 mark** for a single successful application of the distributive law that produces a simpler expression.

**Note:** a simpler expression is one that is logically equivalent to the original expression but uses fewer logical operators.

**Max 3** if answer is correct but any incorrect working or significant steps of working is missing.

**Example working (1)**

****

**Example working (2)**

****

**[4]**

**Q6.**

**Marks are for AO2 (apply)**

**Marking guidance for examiners**

•   Award marks for working out until an incorrect step has been made.

•   If, in any one step, a candidate is simplifying different parts of an expression simultaneously award all relevant marks for this multiple stage but don’t award any further marks for working in any parts simplified incorrectly. Example, if the expression P.P.(P+Q) + P.P.1 was changed to P.(P+Q)+P.0, the candidate would get one mark for simplifying the first part to P.(P+Q) and could get further marks for correctly simplifying this part of the expression further but should not be awarded marks for simplifying the incorrectly changed part P.0 (i.e. to 0)

**Mark as follows**

**1 mark** for final answer A

**3 marks for working**

**Max 3** for working. Award up to two marks for applying each one of the three techniques (one mark per application):

•   a successful application of De Morgan’s Law (and any associated cancellation of NOTs) that produces a simpler expression

•   applying an identity other than cancelling NOTs that produces a simpler expression

•   successfully expanding brackets

**Note:** A simpler expression is one that is logically equivalent to the original expression but uses fewer logical operators.

**Note:** Any application of De Morgan’s Law or expanding brackets which result in an expression which should be bracketed must be shown with brackets to be awarded a mark.



**[4]**

**Q7.**

(a)     operand;

**R** operand code

**1**

(b)     (i)      PC        0010;
MAR     0001;
MBR     00100100;

**3**

(ii)     The instruction is held in the CIR // instruction in CIR is decoded;
**A** IR

The control unit / instruction decoder decodes the instruction;
**NE** the processor decodes the instruction

Instruction will be split into opcode and operand;
**R** if it is implied that a register will do this splitting / decoding

Relevant part of processor / CPU executes instruction // using ALU to perform calculations;
**A** instruction executed by the control unit / ALU
**NE** processor executes instruction

Further memory fetches / saves carried out if required;

Result of computation stored in accumulator / register / written to main memory;

Status register updated;
If jump / branch instruction PC is updated;

*By example:*Will ADD contents memory location 0100 to accumulator;

**MAX 3**

(c)     The current value in the accumulator would be stored in (memory) address / location 0011 / 3;

Number 011 / 3 stored in (memory) address / location 0011 / 3;

**MAX 1**

**[8]**

**Q8.**

**4 marks for AO1 (knowledge) and 4 marks for AO1 (understanding)**

|  |  |
| --- | --- |
| **Description** | **Explanation** |
| Contents of the Program Counter / PC transferred to the Memory Address Register / MAR | so that the PC can be updated // to enable the memory address to be transferred along the address bus/to the memory |
| Contents of MAR placed onto address bus | so the correct location in the main memory will be accessed |
| Contents of addressed memory location/value received on data bus loaded into the Memory Buffer Register / MBR | not all fetches will be for instructions so cannot be loaded directly into Current Instruction Register / CIR // the value will only be present transiently on the bus so must be stored in a register // the MBR is used to cope with the speed difference between the processor and the main memory |
| (Contents of) PC is incremented | so that the next instruction in the sequence can be fetched |
| The contents of the MBR is copied to the CIR | so that if data is fetched/written during the execute phase it does not overwrite the instruction // because the control unit uses the instruction from the CIR |

**A.** Memory Data Register/MDR for Memory Buffer Register/MBR

**Max 4** for descriptions

**Max 4** for explanations

**Max 8**

**[8]**