MODULE C DATA REPRESENTATION · C5

Binary addition and overflow: mark scheme

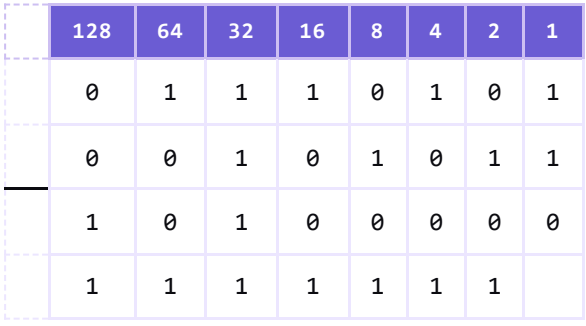
21 marks · Cambridge 0478 syllabus 1.1.4

How to use. One mark per creditworthy point up to the maximum shown. **/** separates alternative words; **//** separates alternative answers. Words in (brackets) are not needed for the mark. Binary addition: one mark for each correct nibble (max 2) plus one mark for a correct method (carries shown). Converting to denary and back does not earn the method mark. These questions say "You must show all your working", so a correct answer with no carries earns the nibble marks only. In the grids, each carry is shown under the column it is carried **into** (the dashed column is the overflow bit); carries written above the numbers are equally fine.

QUESTION	ANSWER	MARKS	GUIDANCE
1(a) 0478 1.1.4(b)	255	**1**	Correct answer only. Do not accept 256.
1(b) 0478 1.1.4(a)	One mark for each of: method: carries shown correctly (into the 2, 4, 8, 16, 32, 64 and 128 columns) 1000 (most significant nibble) 0100 (least significant nibble)	**3**	Answer 10000100. Nibble marks only for the correct nibble in the correct place. Check: 103 + 29 = 132.
2(a) 0478 1.1.4(a) (b)	One mark for each of: method: carries shown correctly (into the 4, 16, 32, 64 and 128 columns) 0001 (most significant nibble) 0101 (least significant nibble) overflow bit shown // final carry of 1 out of the 128 column (shown as a 9th bit)	**4**	8-bit answer 00010101; accept 100010101 (9 bits) for the nibble marks. Check: 186 + 91 = 277; 277 − 256 = 21 = 00010101.

	128	64	32	16	8	4	2	1	
	0	1	1	0	0	1	1	1	
	0	0	0	1	1	1	0	1	+
	1	0	0	0	0	1	0	0	Answer
	1	1	1	1	1	1	1		Carries

	128	64	32	16	8	4	2	1	
	1	0	1	1	1	0	1	0	
	0	1	0	1	1	0	1	1	+
1	0	0	0	1	0	1	0	1	Answer
1	1	1	1	1		1			Carries

QUESTION	ANSWER	MARKS	GUIDANCE
2(b) 0478 1.1.4(b)	Any two from: the result (277) is greater than 255 ... which is the largest value an 8-bit register can store the result needs 9 bits // there is a carry out of the most significant bit / 128 column there is no 9th bit / space in the register for the carry // the extra bit cannot be stored	**2**	max 2. "The number is too big" alone is too vague.
3(a) 0478 1.1.4(b)	Greater than 255 in 8-bit register: True Both MSBs 0 can overflow: False 16-bit register stores 300: True Correct result still stored: False	**4**	One mark per correct row. More than one tick in a row = 0 for that row. Row 2: the largest sum is 127 + 127 = 254. Row 3: the 16-bit maximum is 65 535.
4(a) 0478 1.1.4(b)	65 535	**1**	Accept 65535. Do not accept 65 536.
4(b) 0478 1.1.4(b)	Any two from: an overflow error occurs when a value outside / greater than the limit should be returned / stored e.g. a result greater than 255 in an 8-bit register // greater than 65 535 in a 16-bit register the value needs more bits than are available // there are not enough bits to store it (so) the stored result is incorrect	**2**	max 2.
5(a) 0478 1.1.4(a)	One mark for each of: method: carries shown correctly (into the 2, 4, 8, 16, 32, 64 and 128 columns) 1010 (most significant nibble) 0000 (least significant nibble) 	**3**	Answer 10100000. Check: 117 + 43 = 160.
5(b) 0478 1.1.2(b)(i)	160	**1**	Follow through from the candidate's 8-bit answer to 5(a).

Total: 21 marks