Binary addition and overflow

2.1.4, 2.1.5 · Data · revision notes

How to add two positive 8-bit binary numbers using the binary addition rules, and what an overflow is and why it happens. Expect a 2-mark addition and 1 to 4 marks explaining overflow, often in a context such as a counter.

Key facts

- 0 + 0 = 0
- 0 + 1 = 1
- 1 + 1 = 0 carry 1
- 1 + 1 + 1 = 1 carry 1
- Overflow: a result too large for the number of bits available
- 8-bit unsigned overflow: a total above 255, shown by a carry out of the MSB
- After an 8-bit overflow, the stored result = true total − 256
- Grade 9+ only Two's complement overflow: two positive numbers give a result with an MSB of 1 (it looks negative)

Notes

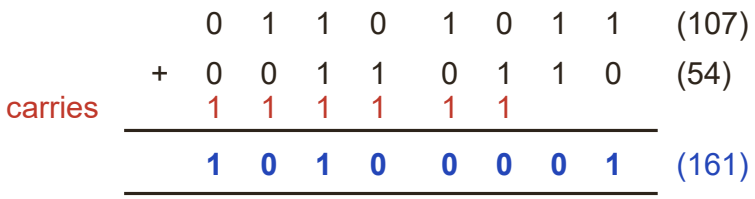
The rules for adding binary

- 0 + 0 = 0, and 0 + 1 = 1.
- 1 + 1 = 2, which is 10 in binary: write 0 and carry 1 into the next column.
- 1 + 1 + 1 (including a carried 1) = 3, which is 11 in binary: write 1 and carry 1.

Adding two 8-bit numbers

- Write one number above the other, with the columns lined up. Start at the right (the least significant bit) and work left.
- Write each carry clearly under the next column so that you remember to add it.
- Example (the denary values are in brackets):

```
  0110 1011  (107)
+ 0011 0110   (54)
= 1010 0001  (161)
```

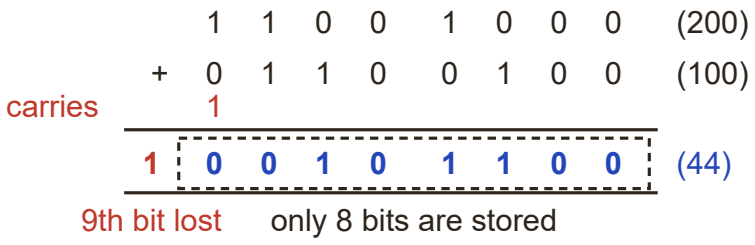


Work from the right. Each carry (red) is added into the next column to the left

- Check by converting to denary: 107 + 54 = 161, and 1010 0001 = 128 + 32 + 1 = 161.

Overflow

- An **overflow** happens when the result of a calculation is too large to be stored in the number of bits available.
- 8 bits can store up to 255. If two 8-bit numbers add up to more than 255, a 1 is carried out of the most significant bit: the result needs a 9th bit that does not fit.
- The extra bit is lost, so the stored result is wrong. Example: 1100 1000 (200) + 0110 0100 (100) = 1 0010 1100 (300), but only 0010 1100 (44) is stored.



Overflow: 200 + 100 = 300 needs 9 bits. The carry out of the MSB is lost, so 44 is stored

- Consequences: a wrong value is stored and used in later calculations or decisions, or the program reports an error or crashes.
- To avoid it, use more bits for the value, or make the program check for overflow.
- Two's complement: two positive numbers can add up to more than +127 without a carry out. The carry into the MSB makes the result look negative, e.g. 0110 0000 (96) + 0101 0000 (80) = 1011 0000, which reads as −80. This is still an overflow. Grade 9+ only

How to answer each type of question

Add two 8-bit binary numbers

2 marks **Grade 5**

1. Line up the columns and start at the right.
2. Use the rules: 1 + 1 = 0 carry 1, and 1 + 1 + 1 = 1 carry 1.
3. Write every carry under the next column.
4. Check by converting both numbers and your answer to denary.

Example. Add these two 8-bit binary numbers. Give your answer as an 8-bit binary number.

```
  0101 1101
+ 0010 0111
```

Model answer.

	0	1	0	1	1	1	0	1
+	0	0	1	0	0	1	1	1
carries	1	1	1	1	1	1	1	
	1	**0**	**0**	**0**	**0**	**1**	**0**	**0**

1000 0100

Left-hand four bits 1000 (1); right-hand four bits 0100 (1)

(Check: 93 + 39 = 132 = 128 + 4.)

Explain why an overflow has occurred

2 marks **Grade 6**

1. Work out the true total (in denary if that helps).
2. Compare it with 255, the largest value 8 bits can store.
3. Say that the carry out of the MSB is lost, and give the wrong value that is stored if you can.

Example. A car park counter stores the number of cars as an 8-bit unsigned binary integer. It currently holds 1111 0000. Another 0001 0100 cars arrive.
Explain why the counter will not show the correct total.

Model answer. 240 + 20 = 260, which is larger than 255, the largest value 8 bits can store (1). The result needs 9 bits, so an overflow occurs: the carry out of the most significant bit is lost and only 0000 0100 (4) is stored (1).

Describe how binary addition works and how overflow is detected 3 to 4 marks **Grade 7**

1. Say that the bits are added column by column from the right.
2. Give the carry rules.
3. Say that a carry out of the leftmost column means the result does not fit: an overflow.

Example. Describe how a processor adds two 8-bit unsigned binary numbers and how it can tell that an overflow has occurred.

Model answer. The bits are added column by column, starting with the least significant (rightmost) bit (1). 1 + 1 gives 0 with a carry of 1 into the next column (1). A carried 1 is added into the next column, so 1 + 1 + 1 gives 1 carry 1 (1). If a 1 is carried out of the most significant (leftmost) bit, the result needs 9 bits, so an overflow has occurred (1).

Explain an overflow in two's complement addition 2 marks **Grade 9**

1. Work out the correct total and compare it with +127.
2. Say that the carry into the sign bit makes the MSB 1.
3. Say that two positive numbers have given a negative-looking result, which shows an overflow.

Example. Two 8-bit two's complement integers, 0100 1100 (+76) and 0011 1000 (+56), are added. The result is 1000 0100.
Explain why this result is wrong.

Model answer. The correct total is +132, which is larger than +127, the largest value 8-bit two's complement can store (1). The carry into the sign bit makes the MSB 1, so the result reads as −124: adding two positive numbers has given a negative result, so an overflow has occurred (1).

Don’t lose marks

- Writing 2 in a column. Binary only has 0 and 1: 1 + 1 is 0 carry 1.
- Forgetting to add the carried 1 into the next column.
- Starting the addition from the left instead of the right.
- Defining overflow as 'the number is too big' without adding 'for the number of bits available'.
- When asked for the 8-bit result after an overflow, writing all 9 bits. Leave off the carried-out bit, and say that an overflow has occurred.

More tips

Memory tricks

- Say it as you add: 'one plus one is nought, carry one'; 'one plus one plus one is one, carry one'.
- Estimate in denary first. If the two values add up to more than 255, expect an overflow.
- Quick overflow test for 8-bit unsigned numbers: if both numbers have a 1 in the MSB, there will always be an overflow (128 + 128 = 256).
- Look at the last column: a carry left over after the leftmost bit means an overflow.

Exam technique

- Show your carries. Clear carries make it easier to find a slip when you check your answer.
- In overflow explanations, give the numbers: the true total, the limit of 255 and the fact that the 9th bit is lost.
- Use the context: say what goes wrong in the scenario (e.g. the counter shows 4 cars instead of 260).
- Addition answers are often marked in groups of four bits, so write your answer as two nibbles.

What each grade needs

What you need to be able to do, from the first marks up to the top grade.

Grade 4 — **Use the four binary addition rules**
0 + 0 = 0, 0 + 1 = 1, 1 + 1 = 0 carry 1, and 1 + 1 + 1 = 1 carry 1.

Grade 5 — **Add two 8-bit binary numbers**
Add column by column from the right, carrying 1 into the next column when needed.

Grade 5 — **Define overflow**
An overflow happens when a result is too large to be stored in the number of bits available.

Grade 6 — **Spot an overflow in an 8-bit addition**
A 1 carried out of the most significant bit means the result needs 9 bits.

Grade 7 — **Explain the effect of an overflow**
The carried-out bit is lost, so the stored result is wrong (256 less than the true total for 8 bits).

Grade 9 — **Explain overflow when adding two's complement numbers**
Two positive numbers with a total above +127 give a pattern with an MSB of 1, which reads as negative.

Free notes, Full packs, practice papers and mock exams at chhetriacademy.com. Written for this site; not taken from past papers. Stuck? 1-to-1 tutoring with a free 30-minute trial lesson.