# Worksheet 2:

**Task 2: Binary Addition**

Using rules (a)-(e) of binary addition below as a guide, work out the answers to questions 1-10.

1. **0 + 0 = 0**
2. **0 + 1 = 1**
3. **1 + 0 = 1**
4. **1 + 1 = 0** *Carry* ***1* = 10**
5. **1 + 1 + 1 = 1** *Carry* **1 = 11**
6. Start with this simple sum. (You can use rules (a) and (b) to help you if necessary.) Calculate the denary equivalent to check that it is correct.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | **8** | **4** | **2** | **1** |  |  | **Denary equivalent** |
|  | 0 | 0 | 1 | 1 | = |  | 3 |
| + | 0 | 1 | 0 | 0 | = | + | 4 |
| = | 0 | 1 | 1 | 1 | = | = | 7 |

1. Use the same techniques as you did in the last question to find the binary result.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | 0 | 0 | 0 | 1 |
| + | 1 | 1 | 0 | 0 |
| = | 1 | 1 | 0 | 1 |

1. Now use rule (d) to help with this problem. Use the carry row at the top for the carried 1.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  |  |  |  |
|  | 0 | 0 | 1 | 1 |
| + | 1 | 0 | 1 | 0 |
| = | 1 | 1 | 0 | 1 |

1. This one will carry into a new column. Remember that like in denary addition, the last carry just makes the number bigger and is added on to the left of the number.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |
|  |  | 1 | 0 | 1 | 0 |
| + |  | 1 | 0 | 1 | 1 |
| = | 1 | 0 | 1 | 0 | 1 |

1. Use rule (e) in this question. Use the carry row again and remember: 1+1+1 = 1 carry 1.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |
|  |  | 0 | 1 | 1 | 0 |
| + |  | 1 | 1 | 1 | 0 |
| = |  |  |  |  |  |

1. Now try a full 8-bit binary pattern. Apply the same rules as before.

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |  |  |
|  | 1 | 0 | 0 | 0 | 1 | 0 | 1 | 1 |
| + | 0 | 1 | 0 | 0 | 1 | 0 | 1 | 0 |
| = | 1 | 1 | 0 | 1 | 0 | 1 | 0 | 1 |

1. Here is another 8-bit pattern that requires you use all of the rules.

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |  |  |
|  | 1 | 0 | 0 | 1 | 1 | 1 | 1 | 0 |
| + | 0 | 1 | 0 | 1 | 1 | 0 | 1 | 0 |
| = | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 |

1. Now try without the help of the grid or rules (a) to (e) to refer to.

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |  |  |
|  | 1 | 0 | 1 | 0 | 0 | 1 | 0 | 1 |
| + | 0 | 0 | 1 | 1 | 1 | 1 | 0 | 0 |
| = | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 1 |

An 8-bit binary pattern can hold 256 different numbers – 0-255. When the result of the addition is greater than 255, an overflow error occurs!

1. Try and work out the answer here using all the normal rules and note what happens.

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |  |  |  |
|  |  | 1 | 1 | 1 | 1 | 0 | 0 | 1 | 1 |
| + |  | 1 | 0 | 1 | 0 | 0 | 1 | 0 | 1 |
| = | 1 | 1 | 0 | 0 | 1 | 1 | 0 | 0 | 0 |

0

1. How do computers hold numbers greater than 255?