# Workshop 02 - Adders

### Workshop - More Combinatorial Hardware

**Make sure you are marked off during this session. Immediately before marking flag yourself in the pracmarker:**

[**https://cs.adelaide.edu.au/services/pracmarker/**](https://cs.adelaide.edu.au/services/pracmarker/)

This workshop is worth 0.83% of your course mark.  These marks will be given for your participation in your session. As with the first session the purpose of this session is to give you the chance to work with the design of hardware. In this case building simple components from basic gates that you saw in chapter 1.

In this workshop you will be producing HDL code.  For the HDL you may find t[he HDL Survival Guide (Links to an external site.)Links to an external site.](http://www.nand2tetris.org/software/HDL%20Survival%20Guide.html) useful.

The questions below are derived from the second assignment of the Nand2teris course. See [this link (Links to an external site.)Links to an external site.](http://nand2tetris.org/02.php) for more information and test scripts.

Answer the following questions.

##### Question 1  (elementary)

Write HDL code for a **half-adder** gate. (make sure you include comments describing your design).

**In your log for this workshop**write some brief notes your development process for this half-adder gate.

##### Question 2 (intermediate)

Write HDL code for a **full-adder** gate (again, include comments about design)

**Write brief notes**  logging your development process for this full-adder gate.

##### Question 3 (more challenging)

Look at [table 2.6 in the textbook (Links to an external site.)Links to an external site.](http://www.nand2tetris.org/chapters/chapter%2002.pdf). This table describes the behaviour of an ALU in terms of input switches. Answer the following:

1. Consider only the rows for the operations x&y and x+y. **Draw** the gates for an ALU that implements these two operations. Note that this means the only control input needed is **f**. You must assume that you have two 16-bit inputs **x** and **y.**In your answer you may assume that you can use gates for a 16 bit adder and 16-bit versions of any gate specified in your[first assignment (Links to an external site.)Links to an external site.](http://nand2tetris.org/01.php).  This means you should be able to write your answer using just a small number of gates.
2. Copy your diagram from part 1 above and add an input for the last control value **no**. Connect this wire to logic that implements the behaviour specified at the top of the **no** column. What operations would be implemented if **no=1**. (hint: they are not any of the operations currently in the table).

**Include notes** logging your development process for this full-adder gate.

At the end of your session go to the [pracmarker system](https://cs.adelaide.edu.au/services/pracmarker) (navigate to workshop 2 of this course)  and, when the supervisor comes to check your work  flag your work for marking.

#### Additional Questions

##### Summary of 2’s Complement Representation of Numbers

Adders and the ALU in these machines use 2s Complement arithmetic to represent numbers. 2’s complement works by imagining that your binary numbers are put on a number line where zero is shifted rights so the numbers starting with a 1 bit represent negative numbers. See below for the 2’s complement number line for 3-bit numbers.

Notice how the negative binary numbers all start with 1 and the positive all start with zero. To convert a 2’s complement binary x to (-x) you simply invert the bits and add one. This works both going from +ve to –ve and the other way around. 2’s complement is the representation for binary integers on most computers. In this course we use 2’s complement for our arithmetic. Do the following sums in 3-bit 2’s complement arithmetic. -3+4, -3+-2, 3+-7, 2+3 (see how your results compare with adding on the number line).

##### Question 4

Implement and your half-adder on the simulator. Run the assignment 02 tests on it. Would a multi-bit full-adder start with a half adder? Write down one advantage and one disadvantage of doing this.

##### Question 5

Think about why a full adder take in 3 numbers and not two numbers? Draw a diagram of how you would link 4 full adders to make a 4 bit adder? What happens to the last carry bit if you add 010 to 111? Is the result of the addition correct?

##### End of Questions