A simple board and counter challenge that demonstrates the addition of two binary numbers is followed by more formal consideration of binary addition, culminating in the development of a circuit to add together two 4-bit numbers.

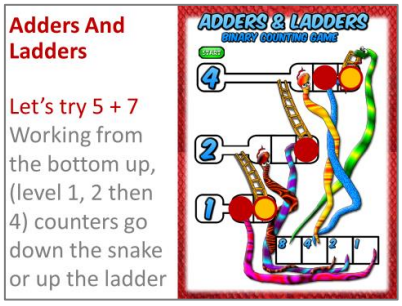
Preparation required:
Adders and ladders board + 6 counters and handout for each group.
Logic Gate Simulator available for all via shared repository.

Adders and Ladders

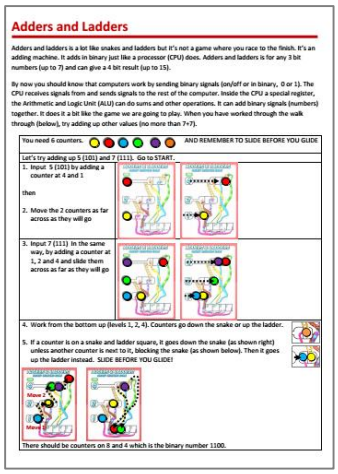
If delving into Boolean algebra is a little daunting, the next exercise should provide some light relief. If we can demonstrate how a computer can use a logic circuit to do some simple maths, we have gone a long way to proving that all computation in a computer can be performed in such a manner i.e. by circuits made up of combinations of logic gates. A standard example is to look at how a computer adds two binary numbers.

Before we look at a circuit to do this, the following exercise ensures children have an introduction to binary addition. It is based on an idea from Neil Ardley's wonderful Young Puffin Fact Book, Bits And Chips. Published in 1991, you still find them available on Amazon secondhand – do try to get hold of a copy. It uses a 'Snakes and Ladders' type board for children to explore binary addition. The resources were developed by Emma Partridge, and shared on the CAS Community. Students will need six counters and a board. The slides provide a walkthrough of how to add two binary numbers – 5 + 7. Make sure you practice in advance before demonstrating it.

The binary value for the first number 101 (5) is placed in the relevant containers, below the start button. Successive clicks reveal the sequence of events. Move the 2 yellow counters as far across as they will go. A second number 111 (7) is input by adding a red counter on each start square on levels 1, 2 and 4. Again, they slide as far as possible to the right. The machine is loaded.

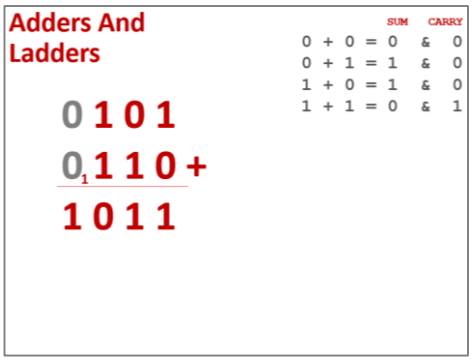


Working from the bottom up, (level 1, 2 then 4) counters go down the snake or up the ladder. The yellow counter on level 1 can't go down the snake as the red counter blocks it … so it goes up the ladder and slides right as far as possible. Now the red counter on level 2 cannot go down the snake as the yellow counter now blocks it … so it also goes up its ladder and to the right as far as possible. The yellow counter on the top level is also blocked so goes up the ladder … and down the snake to the total box! All the other counters are also then checked. On the top level the adjacent red is not on the head of a snake, or ladder, so stays where it is. The leftmost top level counter is on a snake, so down it goes! Neither of the other two counters are on either a snake or a ladder, so they don't move… leaving a counter in the 8 and 4 positions in the 'tray' $1100_2 = 12_{10}$. If the sequence is followed correctly, the counters in the tray will give the answer to the addition in binary. Note the use of the subscript in the presentation to indicate the base of a number.



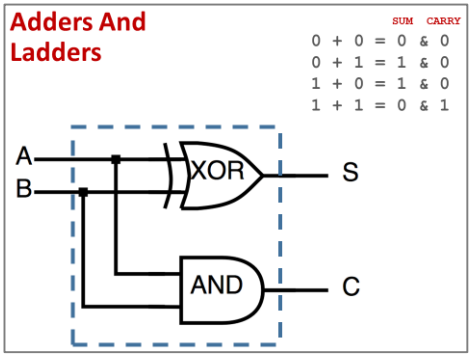
There is a prompt sheet for children to follow. Let them try several additions of two numbers between 0 and 7. It is important they are comfortable about basic binary representation before embarking on the next task. If they struggle to get the right answer, the major pitfall is usually forgetting to go down a snake where there is a choice between both a snake and a ladder (as there is on each rightmost square on each level). They always 'slide' down unless a counter is in the next square to block it, in which case they 'glide' up the ladder. 'Slide before you glide'.

Binary Addition



Let's now try to formalise the rules of adding two binary numbers. Adding in binary is just like any other addition. Lay the two numbers out, one below the other and add each column in turn, working from the right. 1+ 0 = 1, 0 + 1 = 1. But what happens with 1 + 1? Exactly the same rules apply when adding in decimal. If you run out of digits, you carry 1 to the next column. 2 in binary is one-zero (10). So 1 + 1 = 0 and carry 1. So the final column is 0 + 0 + 1 = 1

It's actually very easy to add in binary, because there are only 4 possible rules for adding two bits together. We can summarise them in a table. Ask the pupils to predict the answer to each rule before revealing it with a click.



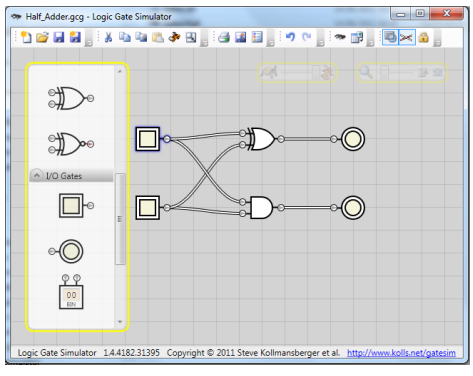
If we consider an adding machine as a black box, the interface is as shown (outside the dotted lines). It has two inputs, A and B, each of which can be 1 or 0. It has two outputs, S for Sum, and C for Carry. Ask students if they know any logic gates that behave in the way described in the Sum column? (It is a XOR). Similarly, we can think of the Carry as another logic circuit. Ask if anyone recognises the truth table this time? (AND). The implementation (inside the dotted lines) is revealed on a click. So a simple adding circuit can be implemented by using two logic gates, connecting the inputs to both an AND and a XOR.

Half Adder Circuit

Armed with this knowledge, students can now build the logic circuit using a simulator. There are several simulators available, but the one suggested is free, requires no installation and has several benefits over some of the alternatives, which will become apparent shortly. It is available in the resources for this unit, or can be downloaded from this link: goo.gl/vISt1i

Included in the resources are the sample circuits we are going to build using this simulator, so do remove those before making it available to students. A folder on a shared network drive is all that is required for students to use the simulator; it loads by clicking GatesWpf.exe.

Also included is a pdf of Steve Kollmansberger's book on Computer Maths and Logic, released under Creative Commons. For those who want to study this in depth (to first year undergraduate level) it is an excellent resource.

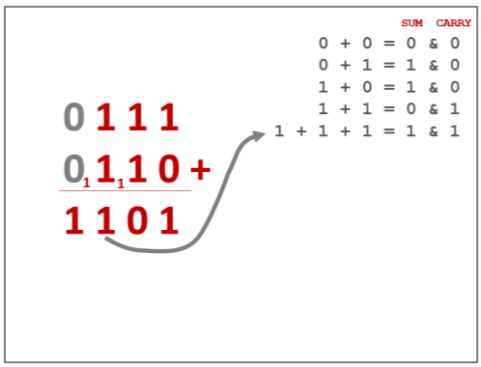


The simulator is very intuitive, the only help students are likely to require is finding the correct components. Switches are found in the I/O Gates palette. AND, OR and NOT gates are in the Basic Gates set, whilst NAND, NOR and XOR will be found in the Compound Gates set.

Components are connected by dragging (a wire) between connections. Inputs and Outputs can be labelled, components can be moved and rotated. Once complete, students can test the logic by clicking on the inputs to toggle their value. It should conform to the addition rules identified earlier.

Once they are familiar with the basic operations, ensure they save their simple adder circuit. Suggest they use the name 'Half Adder', for reasons that will become clear.

More Binary Addition



There is one further addition rule we haven't yet considered. The slides allow a class to consider another sum. A series of clicks will allow you to work through it, but encourage students to come forward, write the answer in a column and articulate which rule they used. It should become apparent that the third column has an extra 'one'. How do we handle that? 1 + 1 + 1 = 3, which in binary is 1, carry 1 (11). We can add this to our table of rules.

Our half adding machine is currently only designed to take two inputs. What modification do we need to make to our half adder diagram, to accommodate this new rule? We need to add a third input, to cope with a carry in from the previous column.

Do we need to redesign our circuit to accommodate this change? Yes and no. Clearly we need to add the extra capability, but we can do it by thinking of a 3 bit addition as two, two bit additions … and we have a machine already designed to handle that.

The slides consider these possible additions:

- 1 + 1 will give a sum of 0, and a carry out of 1. We can then add any carry in, to the sum of the first addition. This gives us a sum of 1 and carry of 1.
- 1 + 0 will give a sum of 1, and a carry out of 0. If we add a 0 carry in, to the sum of the first addition, the sum of 1 remains. Again, the two stage adding gives the correct result.
- 0 + 0 will give a sum of 0, and a carry out of 0. If we add a carry in to the sum of the first addition it gives us a sum of 1. Again, the two stage adding gives the correct result.
- But consider 1 + 0 (which will give a sum of 1, and a carry out of 0) again. If we add a carry in to the sum of the first addition it gives us a sum of 0 and we need to change the carry out to 1.

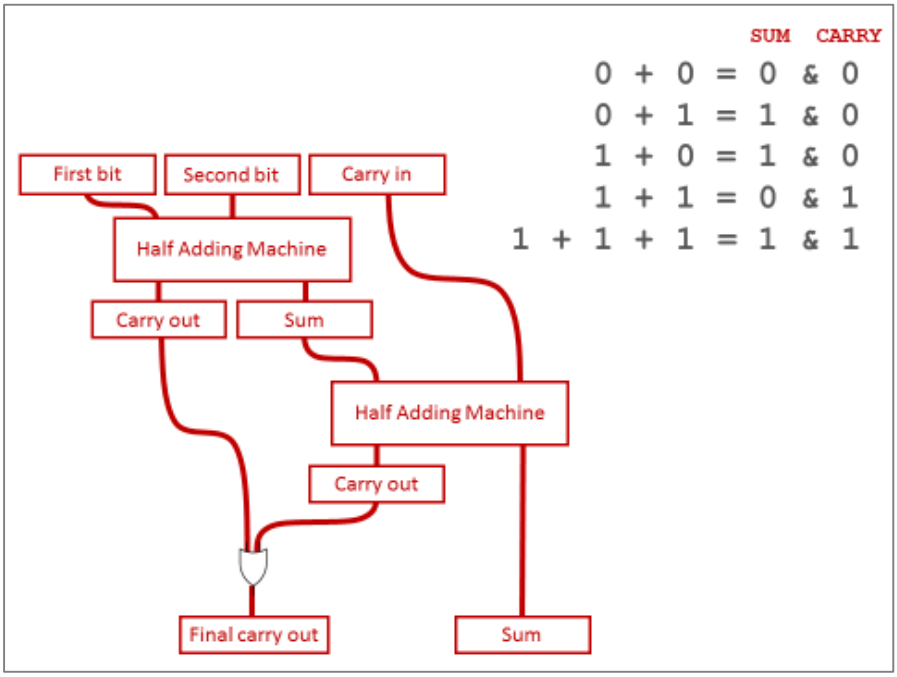
A Full Adder Circuit

The key point from this is that we can use two half adders to do the two stage addition, but the carry out from EITHER circuit must be allowed to set the value of the final carry out.

Armed with the knowledge of the two stage addition, let's build the circuit.

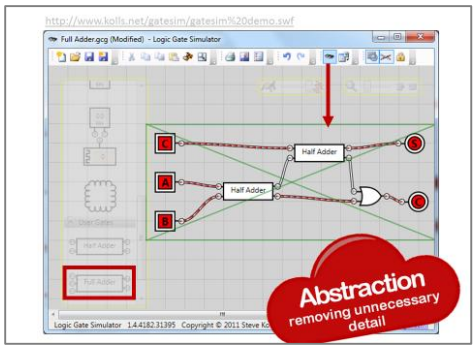
We can start with our Half Adder as previously shown. Then we can add a second Half Adder, using the Sum from the first machine, and the Carry In as inputs. The Sum from this second machine will be our final output, but what about the two Carry Out's from each machine? As EITHER carry out is valid, we can connect the two through an OR gate, to give a final carry.

Study the diagram carefully. It is key to explaining how it works in the presentation slides that follow.



Creating this more complex circuit in a simulator would be challenging, were it not for the fact that we can encapsulate the behaviour of the Half Adder in its own chip. This is the real strength of this simulator.

With the Half Adder circuit open, select the 'Create IC' tool. This allows you to create your own Half Adder chip. In this way we can abstract away the detail of the circuit's implementation. We can now use the new 'chip' to make a Full Adder circuit both easier to build and easier to comprehend.

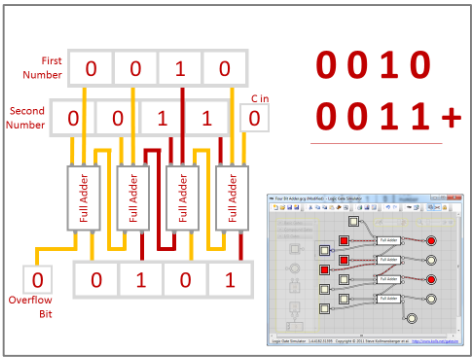


By the same approach, we can now encapsulate a Full Adder circuit in its own chip, and so on. You'll find a very good, 2 minute walkthrough for building a full adder on Steve Kollmansberg's website: goo.gl/DWVoxv.

A Full Adder chip allows two bits to be added together with any incoming carry. Having constructed one, let's consider how they would be linked together to add, in this case, 2 binary numbers, each 4 bits long. The numbers could be set in banks of 4 switches each. In reality, they would be held in registers, which are conceptually similar in makeup.

The initial Carry In would also be set to zero. It has no use here, but does allow further Full Adders to be connected to allow for larger numbers to be handled with larger numbers of bits. The switches from the First Number would be connected to an input of each corresponding Full Adder. Red lines indicate a 1, whilst yellow indicate zero. The same would happen for the Second Number and the initial Carry In.

With the model on the slide, we can now complete the addition in clickable stages. Looking at the sum on the right: in the rightmost column, 0 + 1 will give an output of 1 carry 0. In the circuit the output is therefore enabled but not the carry. Point out how the Carry Out is linked to the Carry In input for the next Full Adder. This completes the inputs for the next chip.



You can ask students what output they would expect from this? 1 + 1 + 0 gives a Sum of 0 and a Carry of 1. The lines in the circuit are enabled on a click: yellow to output 0, and red to pass a carry out of 1, into the next Full Adder. The same thing happens with the next addition. 0 + 0 + 1, outputs 1 and sets the carry to zero. The final calculation outputs 0. Notice that the final Carry Out is connected to an Overflow Bit.

If the addition had generated a number larger than 15, this would have 'overflowed' the number of bits allocated for the addition. The computer would check this value and generate an error message if it held a value of 1.

Once students understand the principle of an adding circuit, all that remains is to build one! Notice how, by encapsulating circuits in their own chips it is a relatively simple thing to do.

Building a full adding circuit may well be beyond many Key Stage 3 students, but there are many simple logic circuit challenges you could give to them. The idea on this slide was shared by Phil Gardner, a CAS member from Devon and the activity cards developed by Mark Clarkson, from Teesside. These make excellent challenges that build up in complexity, ending with half and full adder challenges. They are included in the resources for the unit. Steve Kollmansberg's book, included in the resources is excellent for developing ideas for more challenging material for older students at, say A Level.