# Week 4 Reinforcement

You should try the exercises and make sure that you have your solutions available during the lectorial in week 5. You may want to work on these in small study groups.

As well as working on these exercises, you should also spend some of your individual study time working on your coursework. Remember that it is an individual coursework, but you may ask the tutors questions.

### Part 1 Improving the automation

In the lab you created a simple type of computer that could automatically carry out a list of instructions. In this task you will improve that automation to make it more flexible in one way.

One of the limitations of the automation created in the lab was that one of the inputs to the ALU had to be entered manually into an input pin. In this exercise we will look at including a value in the instruction in the ROM as the second input to the ALU.

Currently the program instruction is just two bits indicating which operation the ALU should do. This will be extended to 16 bits. The two most significant bits will be the operation for the ALU, and the eight least significant bits will be a value to use as the second input to the ALU. Remember that the bits are numbered from 0 to 15 with the least significant bit numbered as 0.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| ALU operation | | Unused (00) | | Unused (0000) | | | | Immediate value to use as input | | | | | | | |
| Hex digit | | | | Hex digit | | | | Hex digit | | | | Hex digit | | | |

Remember that 16 bits is two bytes or four hex digits. The first two bits for the ALU operation will be followed by two zeros to form the four bits in a hex digit

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ALU operation - decimal | Operation description | ALU operation - binary | New format | Hex |
| 0 | Add | 00 | 0000 | 0 |
| 1 | AND | 01 | 0100 | 4 |
| 2 | Subtract 1 | 10 | 1000 | 8 |
| 3 | Left shift | 11 | 1100 | c |

Take a copy of the automation that you created in the lab. Change the ROM to have Data Bit Width of 16 and disconnect it from the input to the ALU.

The instructions from the ROM can now contain an ALU operation and a value, so we need to split the Data from the ROM into separate bits and then combine the bits we need for the relevant inputs to the ALU. Use a splitter to separate all 16 bits from the ROM data into separate bits. Use two splitters to combine bits 0 to 7 to pass to the second input of the ALU and bits 14 & 15 to pass as the ALU operation.

When everything is connected, we need to develop a program. I suggest the following program starting from the accumulator and program counter being set to zero. You should work out the instruction in hex for each instruction so that you can key it into the program memory.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Program counter | Operation (decimal) | Instruction in binary | Instruction in hex | Expected value in accumulator (binary) | Expected value in accumulator (hex) |
| 0 | Add 27 | 0000 0000 0001 1011 |  | 0001 1011 |  |
| 1 | AND with 60 | 0100 0000 0011 1100 |  |  |  |
| 2 | Subtract 1 | 1000 0000 0000 0000 |  |  |  |
| 3 | Left shift | 1100 0000 0000 0000 |  |  |  |

You should work out what you expect to happen at each step in binary and convert to hex so that you can test whether your computing device is working correctly.

### Part 2 Little Man Computer

Following from the exercises on Little Man Computer that you did in the lab, here are some more exercises that you could try using either of the simulators as given in the lab sheet. Some of these are direct extensions of the lab work.

1. Code and test a program that adds three input numbers together.
2. Code and test a program that reports the maximum of three input numbers.
3. Code and test a program that takes three input numbers and outputs them in ascending order.
4. Code and test a program that takes an input number and counts down from that number to 0.