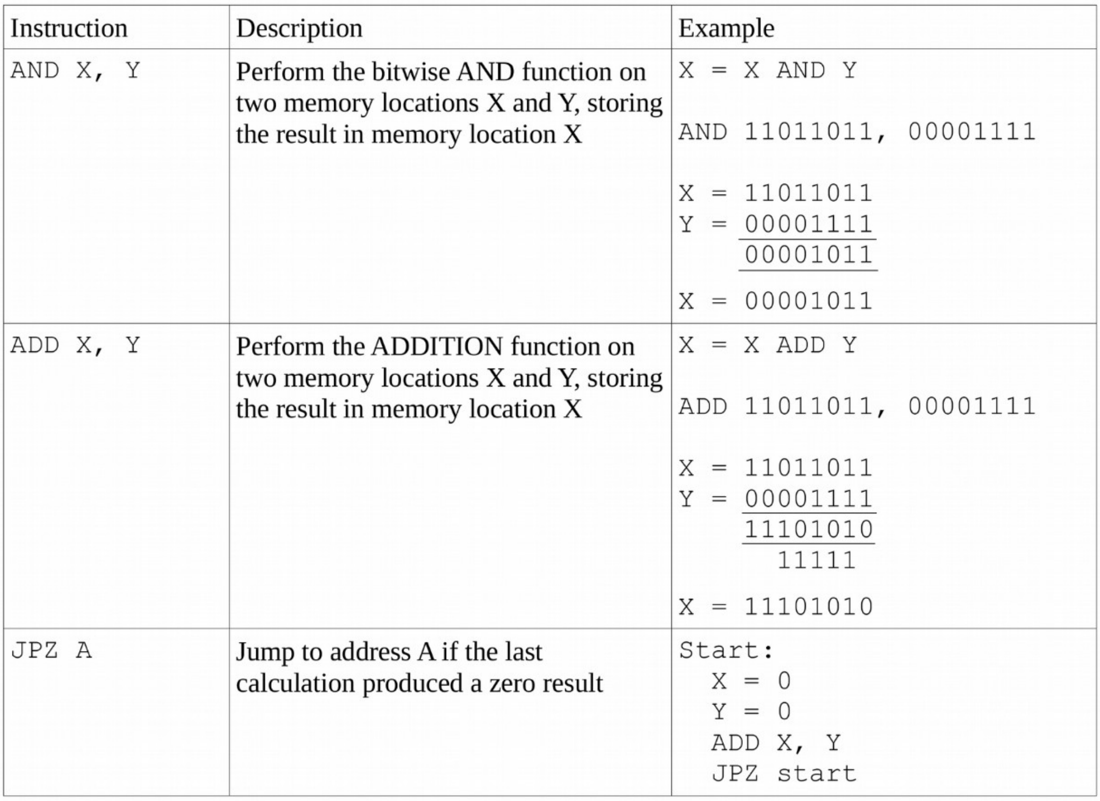
SYS1 : test your level quizzz

Using a very simple processor with only three instructions and no on-processor memory (registers), answer the following questions.

The processor’s instruction-set is shown in figure 1. External memory locations used to store variables are identified using letters i.e. memory locations X, Y etc. All variables are stored in external memory.



Instruction	Description	Example

AND X, Y	Perform the bitwise AND function on two memory locations X and Y, storing the result in memory location X	X = X AND Y
		AND 11011011, 00001111
		X = 11011011
		Y = 00001111
		00001011
		X = 00001011

ADD X, Y	Perform the ADDITION function on two memory locations X and Y, storing the result in memory location X	X = X ADD Y
		ADD 11011011, 00001111
		X = 11011011
		Y = 00001111
		11101010
		11111
		X = 11101010

JPZ A	Jump to address A if the last calculation produced a zero result	Start: X = 0 Y = 0 ADD X, Y JPZ start

Figure 1 : instruction-set

Q1) the memory locations X and Y are initialised with the values:

		2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
		128	64	32	16	8	4	2	1
X = 14	=	0	0	0	0	1	1	1	0
Y = 35	=	0	0	1	0	0	0	1	1

Consider the program below, what is the final value stored in memory location X? **Hint**, remember that the first variable i.e. X in this example, is overwritten with the result.

```
ADD X, Y
AND X, Y
```

Q2) write a program to initialise memory locations `X` and `Y` to the following values `X = 123` and `Y = 0x123` i.e. load these values into memory.

Hint, you can pass an instruction a hard-coded value e.g. `AND X, 67`, rather than two variables e.g. `AND X, Y`. Also, consider the two scenarios where the original value stored in the memory location is zero and when it is a non zero value. A number starting with `0x` indicates that the following value is in hexadecimal i.e. base 16, rather than the default base 10.

Q3) write a program to perform the following calculation.

```
X = X × 5
```

Hint, what is multiplication? You will need to use multiple instructions to implement the × function.

Q4) write a program to perform the following calculation. Your solution must use less than eight instructions.

```
X = X × 16
```

Q5) write a program to perform the following calculation i.e. the multiplication of two unknown variables.

```
X = X × Y
```

Hint, you will need to use the `JUMP` instruction. Jumps are made to labels (addresses) within a program i.e. a name (label) ending with a ":", as shown in figure 1. This example perform the pseudo code below, the `JPZ` instruction jumps to the label `start` if the result of the `ADD` instruction produces a zero result, otherwise the next instruction in the program is executed i.e. it gets stuck in an infinite loop.

```
START:
  X=0
  Y=0
  X=X+Y
  IF RESULT = 0 THEN START
```

Q6) write a program to perform the following calculation.

```
X = X OR Y
```

where the OR function performs a bitwise OR on the memory locations `X` and `Y`. This program should be equivalent to the bitwise OR instruction described below. Your solution must be less than 16 instructions. You may assume `X` and `Y` have already been loaded into memory.

Hint, this one is tricky :)

Instruction	Description	Example

`OR X, Y`	Perform the bitwise OR function on two memory locations X and Y, storing the result in memory location X	`X = X OR Y`
		
		`OR 11011011, 00001111`
		
		`X = 11011011`
		`Y = 00001111`
		`    11011111`
		
		`X = 11011111`

Figure 2 : bitwise OR

Your SYS1 starting level is given by the number of questions you can answer:

- Level 0 – 1 : beginner, not a problem, we will take you through the basics, from the ground up. **BUT** do make sure you do the labs and exercises, do make sure you understand the tasks in these labs, do pop round for a chat, or email me if you have questions, and all will be good. Do have a look through the background reading material, before the start of the module:

  http://simplecpudesign.com/simple_cpu_v1/index.html

- Level 2 – 3 : know the basics, should be a bit of revision in the early lectures, then we get into the good stuff i.e. building processors. You may want to look at the software tools i.e. Xilinx ISE (links on VLE, also installed on VDS and in labs). You may also want to look through the material related to the first processor we will be designing in the this module:

  http://simplecpudesign.com/simple_cpu_v1a_fpga/index.html

- Level 4 – 5 : good understanding, hopefully we will dig a little deeper than you have done in A-levels etc. Give you the chance to apply the knowledge you have gained through the development of a working system e.g. control the bug trap hardware in the lab, or develop your own processor, its instruction-set and using it to build a video game. You may also want to look through the material related to the final processor we will be designing in the this module:

  http://simplecpudesign.com/simple_cpu_v1d_fpga/index.html

- Level 6 : not expecting too many people to be able to answer all these questions, but if you can you will find SYS1 easy, will have to find some prizes :)