University Of Portsmouth
BSc (Hons) Computer Science
First Year

Architecture and Operating Systems (Computer)
M30943
September 2022 - May 2023
20 Credits

Thomas Boxall
up2108121@myport.ac.uk

Contents

S.1. Introduction to Module (26-09-22) **2**

S.2. Binary Arithmetic (26-09-2022) **3**

S.1. Introduction to Module

26-09-22 16:00 Farzad RB LT1

Division of the Module

This module is split into two parts: computer (this part) which is worth 70% and maths (the other part) which is worth 30%. The two parts are run completely independently of each other. The only time they come together is when the final overall score is calculated.
There are two separate Moodle pages (one for Computer and one for Maths)

Computer Module assessments

For the Computer section of the module, there are two assessments. One is in January 2023, which will be a Computer Based Test (covering content taught in the first teaching block). It is worth 30% of the over module score. The second is in the May/June 2023 assessment period. It will be computer based. This assessment will be worth 40% of the overall module score.
Both assessments are closed book however a formula sheet will be provided for the January assessment. Nothing is provided for the May/June assessment.
The pass mark for the entire module is 40%, this score is generated from all the computer assessments AND all the maths assessments.

Module structure

There will be a one hour lecture per week, where content is introduced to us. This will be delivered using worksheets for the first 10 weeks.
There will also a practical session each week where the cohort is split into smaller groups. These sessions will be a chance to practice the ideas introduced in the lectures. There will be more members of staff around at the practical sessions to help out.

More Information on Practical Sessions

There are practical session guidelines available in the induction slides or on Moodle.

Content in each Week

There is a teaching plan on Moodle which outlines the content covered each week as well as the weeks in which the exams will be held.

S.2. Binary Arithmetic

26-09-2022 16:15 Farzad RB LT1

Number Systems

There are a number of different number systems and different methods to convert between them.

Denary (Base 10)

Used most commonly, this is the one most people learn.

10^x	10^3	10^2	10^1	10^0
$10^x =$	1000	100	10	1
	4	2	5	1

The total of the numbers above would be calculated in the following way:
$4251 = (1000 \times 4) + (100 \times 2) + (10 \times 5) + (1 \times 1)$
Denary is also known as base 10, this means each column can have one of ten possible values (0, 1, 2, 3, 4, 5, 6, 7, 8, 9)

Binary (Base 2)

This is base 2, this means each column can have one of two possible values (0, 1). The columns are also different. Moving from right to left, the columns double each time.

2^x	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
$2^x =$	128	64	32	16	8	4	2	1
	1	0	1	1	0	0	1	1

The largest value which can be stored in 8-bits of binary is 11111111_2 or 255_{10}.

Hexadecimal (Base 16)

Also known as Hex. Using this method, numbers up to 255 can be stored in two characters. This is used a lot in computing, especially in graphics and website development. Each column can have one of 16 values (1 2 3 4 5 6 7 8 9 A B C D E F). The letters are used to represent two-digit numbers as seen below.

Hex:	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
Decimal:	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15

To calculate the value held in a Hex number, we calculate in a similar way to Denary and Binary as seen below.

16^x	16^3	16^2	16^1	16^0
$16^x =$	4096	256	16	1
	D	3	C	E

$D3CE = (13 \times 4096) + (3 \times 256) + (12 \times 16) + (14 \times 1) = 54222$

Converting Between Number Systems

Binary To Denary

Add together all the columns in which there is a 1. Using the example shown in the binary section, the total would be 179.

Denary To Binary

This is the reverse of binary to denary. Work from right to left seeing if the value will fit into the column, if it won't then mark down an zero and move onto the next.

Denary to Hex

The easiest way to do this is to go via Binary. Convert the number into binary, then split the binary into two nibbles. The values inputted in the previous step don't need to change. With the two nibbles of (4, 2, 1, 0), convert each of them back into denary, giving two individual digits, then convert each of those into Hex.

Binary Addition

Basic Rules

There are four basic rules to binary addition:

```
0 + 0 =  0
0 + 1 =  1
1 + 1 =  1
1 + 1 = 10
```

The last one (1+1) is a special case; strictly speaking, the answer is 0 with the 1 carried over. This is particularly useful in digital circuitry.

Binary Addition Example

Add 100 + 011

1. Draw out the binary addition columns

$$\begin{array}{rccc} & 1 & 1 & 0 \\ + & 0 & 1 & 1 \\ \hline \end{array}$$

2. Start with the right-most column and add the digits

$$\begin{array}{rccc} & 1 & 1 & 0 \\ + & 0 & 1 & 1 \\ \hline & & & 1 \end{array}$$

3. Move to the next column. Add those digits together. As this is 1+1 = 0 carry 1, we write a little 1 in the next column as a carry.

$$\begin{array}{rccc} & {}^{1}1 & 1 & 0 \\ + & 0 & 1 & 1 \\ \hline & & 0 & 1 \end{array}$$

4. Move to the next column and add that. Remember to add the carry (making the sum $1+1+0$). This results in 0 carry 1, so, again, we add a little 1 in the next column. It doesn't matter that there aren't any other numbers there to be added, we will make a column!

$$\begin{array}{ccccc} & {}^{1} & {}^{1}1 & 1 & 0 \\ + & & 0 & 1 & 1 \\ \hline & & 0 & 0 & 1 \end{array}$$

5. Add the final column

$$\begin{array}{ccccc} & {}^{1} & {}^{1}1 & 1 & 0 \\ + & & 0 & 1 & 1 \\ \hline & 1 & 0 & 0 & 1 \end{array}$$

This gives us out final answer of $110 + 011 = 1001$

Binary Multiplication

There are 4 basic rules for binary multiplication.

```
0 x 0 = 0
0 x 1 = 0
1 x 0 = 0
1 x 1 = 1
```

Binary Multiplication Example

Multiply 10×11

1. Draw out the multiplication grid as you would for a standard column multiplication with decimal numbers.

$$\begin{array}{cccc} & & 1 & 1 \\ \text{x} & & 1 & 0 \\ \hline \end{array}$$

2. Take the right-most digit of the bottom binary number, we will multiply it with each of the digits above and place their results directly underneath.
$0 \times 1 = 0$, place this directly under the 0
$0 \times 1 = 0$, place this to the left.

$$\begin{array}{cccc} & & 1 & 1 \\ \text{x} & & 1 & 0 \\ \hline & & 0 & 0 \end{array}$$

3. Next, move to the next digit on the bottom row, repeat the same process as before.

$$\begin{array}{cccc} & & 1 & 1 \\ \text{x} & & 1 & 0 \\ \hline & & 0 & 0 \\ & 1 & 1 & \end{array}$$

4. We then add our two answer rows together, using the rules of binary addition.

		1	1
x		1	0
		0	0
+	1	1	
	1	1	0

This gives us the final answer of $11 \times 10 = 110$

Binary Subtraction

At this stage, there are four basic rules for binary subtraction. At a later stage, there will be negative numbers introduced when we look at signed binary so more rules will be introduced.

```
 0 - 0 = 0
 1 - 0 = 1
 1 - 1 = 0
10 - 1 = 1
```

Binary Subtraction Example

Subtract $110 - 001$

1. Draw out the subtraction columns as you would for a standard decimal column subtraction

	1	1	0
-	0	0	1

2. Start at the right hand most column $(0 - 1)$. This is something which we can't do, so we have to borrow 1 from the left hand column. To represent this, cross out the borrowed digit, replace with a 0 and in the current column, add a little 1 to the left.
   This leaves us with $10 - 1$, which we can do and know equals 1.

	1	${}^{0}\cancel{1}$	${}^{1}0$
-	0	0	1
			1

3. Now move to the next column, and perform that operation. This is the middle column which is $0 - 0 = 0$.

	1	${}^{0}\cancel{1}$	${}^{1}0$
-	0	0	1
		0	1

4. Move to the next column, and perform that operation. This is the left column which is $1 - 0 = 1$

	1	${}^{0}\cancel{1}$	${}^{1}0$
-	0	0	1
	1	0	1

This gives us the final answer of $110 - 001 = 101$

Binary Division

Binary division follows much the same procedure as 'bus stop' decimal division.

Binary Division Example

Divide $110 \div 10$

1. Draw out the division columns as you would for a standard decimal 'bus stop' division.

$$\begin{array}{ccccc} \cline{3-5} 1 & 0 & \multicolumn{1}{|c}{1} & 1 & 0 \end{array}$$

2. Start by looking for factors and find 11 is greater than 10. We then write the number of times the value goes into 11 at the top, and the value itself underneath.

$$\begin{array}{ccccc} & & & 1 & \\ \cline{3-5} 1 & 0 & \multicolumn{1}{|c}{1} & 1 & 0 \\ & & 1 & 0 & \end{array}$$

3. We then subtract to see if there is a remainder. $(11 - 10 = 01)$
   The remainder is written up on the top line

$$\begin{array}{ccccc} & & & 1 & 1 \\ \cline{3-5} 1 & 0 & \multicolumn{1}{|c}{1} & 1 & 0 \\ & - & 1 & 0 & \\ \cline{3-4} & & 0 & 1 & \end{array}$$

4. We then bring down the final digit in the division (0), to where we are working.

$$\begin{array}{ccccc} & & & 1 & 1 \\ \cline{3-5} 1 & 0 & \multicolumn{1}{|c}{1} & 1 & 0 \\ & & 1 & 0 & \\ \cline{3-4} & & 0 & 1 & 0 \end{array}$$

5. Now, we look to see if our divisor can fit in again. It does fit again, so we subtract it. $(010 - 10 = 0)$ It leaves no remainder.

$$\begin{array}{ccccc} & & & 1 & 1 \\ \cline{3-5} 1 & 0 & \multicolumn{1}{|c}{1} & 1 & 0 \\ & & 1 & 0 & \\ \cline{3-4} & & 0 & 1 & 0 \\ & - & & 1 & 0 \\ \cline{4-5} & & & & 0 \end{array}$$

This gives us the answer of $110 \div 10 = 11$.