Edinburgh Napier UNIVERSITY

Systems & Services CSN08101

Introduction to the Architecture block

Contents:

- Introduction to the Architecture section
- Useful URLs and books
- Introduction to PC hardware
- Computer buses and data transfers

Edinburgh Napier UNIVERSITY

CSN08101 - Systems and Services

Weeks: 2-15 - Assessment Week (15 Sep 2014-21 Dec 2014)

	9:00	10:00	11:00	12:00	13:00	14:00	15:00	16:00	17:00
Mon								CSN08101.P1c <8-13> Systems & Services Merch.C6.co, Merch.C27.co Mr Jim Jackson;Mr Alastair Soutar Practical Weeks: 8-13	
								CSN08101.P1c <2-7> Systems & Services Merch.C6.co, Merch.C27.co Dr Alistair Armitage;Mr Frank Greig Practical Weeks: 2-7	
Tue			CSN08101.T1 <8-13> Systems & Services Merch.C27.co Mr Jim Jackson;Mr Alastair Soutar Practical Weeks: 8-13	CSN08101.L1 <2-7> Systems & Services Merch.A17 Dr Alistair Armitage;Mr Frank Greig Lecture Weeks: 2-7					
			CSN08101.T1 <2-7> Systems & Services Merch.C27.co Dr Alistair Armitage;Mr Frank Greig Practical Weeks: 2-7	CSN08101.L1 <8-13> Systems & Services Merch.A17 Mr Jim Jackson;Mr Alastair Soutar Lecture Weeks: 8-13					
Wed	CSN08101.P1a <8-13> Systems & Services Merch.C6.co, Merch.C27.co Mr Jim Jackson;Mr Alastair Soutar Practical Weeks: 8-13								
	CSN08101.P1a <2-7> Systems & Services Merch.C6.co, Merch.C27.co Dr Alistair Armitage;Mr Frank Greig Practical Weeks: 2-7								
Thu						CSN08101.P1b <8-13> Systems & Services Merch.C6.co, Merch.C27.co Mr Jim Jackson;Mr Alastair Soutar Practical Weeks: 8-13			CSN08101.L2 <2-7> Systems & Services Merch.A17 Dr Alistair Armitage;Mr Frank Greig Lecture Weeks: 2-7
						CSN08101.P1b <2-7> Systems & Services Merch.C6.co, Merch.C27.co Dr Alistair Armitage;Mr Frank Greig Practical Weeks: 2-7			CSN08101.L2 <8-13> Systems & Services Merch.A17 Mr Jim Jackson;Mr Alastair Soutar Lecture Weeks: 8-13
Fri									

Edinburgh Napier UNIVERSITY

Assessments

ASSESSMENTS	WEIGHT	MATERIAL ASSESSED	Week	Time and Date
1A. Class test	50%	Hardware (units 1 - 6)	7	17:00, 23rd Oct 2014, A17
1B. Coursework	50%	Linux scripting and Operating systems	12/13	Part A – practical in lab Sessions; Part B – class test date 27th Nov (TBC), A17

Week begin	Wk	Lecture 1 Tuesday 12:00 1hr A17	Lecture 2 Thursday 17:00 1hr A17	Practical C6/C27	Practical/ Tutorial C6/C27
15/09/14	2	Unit 1: revision of hardware, Fetch execute AA	Unit 2a: PC architecture & mother-boards AA	Lab: CPU simulation (No Monday class: local holiday)	Tutorial 1: Number systems & basic architecture
22/09/14	3	Unit 2b: the Pentium Architecture AA	Unit 3a: RISC architecture AA	Lab: CPU-ID	Tutorial 2: PC motherboards
29/09/14	4	Unit 3: dram & cache AA	Unit 4: I/O Methods FG	Pentium assembly	Tutorial 3: Assembly lang.
06/10/14	5	Unit 4: I/O Methods FG	Unit 5: Bus Systems FG	Lab: GPIO	Tutorial 4: I/O Methods
1310/14	6	Unit 5: Bus Systems FG	Unit 6: I/O Devices FG	Lab: Serial I/O	Tutorial 5: Bus Systems
20/10/14	7	Unit 6: Interface Devices FG	HW Test (Units 1-6)	Lab: Analogue & Interrupts	Tutorial 6: Interface Devices & Test Revision
27/10/14	8	Intro to O/S AS OS Command line revision JJ	Linux utility commands JJ	Lab: C27 Practical Linux JJ	
03/11/14	9	Shell Scripts I JJ	Shell Scripts II JJ	Lab: C27 Linux Scripts JJ	
10/11/14	10	Processes ASr	Scheduling ASr	Lab: C27 Linux Coursework I JJ	
17/11/14	11	Memory ASr	Filestores ASr	Lab: C27 Linux Coursework II JJ	
24/11/14	12	Revision AS	Mini Class Test	Coursework submission/ demo JJ/ASr	
01/12/14	13	HW Test resit AA/FG	Mini Test resit JJ/ASr	Demo / Re- Submission JJ/ASr	
0812/14	14				

Systems & Services

Edinburgh Napier UNIVERSITY

- Two main topics:
  A. Computer architecture. (units 1-6)
  B. Operating systems. (units 7-12)

- Computer Architecture is further divided:
  - Units 1-3: PC architecture & the Pentium (Alistair Armitage)
  - Units 4-6: microcontrollers & I/O (Frank Greig)

- Learning Outcomes: Block A (Computer Architecture):
  - “Analyse the architecture of a range of processor based system”.
  - “Compare and contrast the different approaches to system architecture”.

Practical Outcome

- To be able to understand hardware specifications.
- To be able to evaluate alternative PC solutions.
- To understand current trends in computer architecture.
- To have an introductory knowledge of some non-PC computing systems (Microcontrollers, mobile systems).

Useful URLs

Edinburgh Napier UNIVERSITY

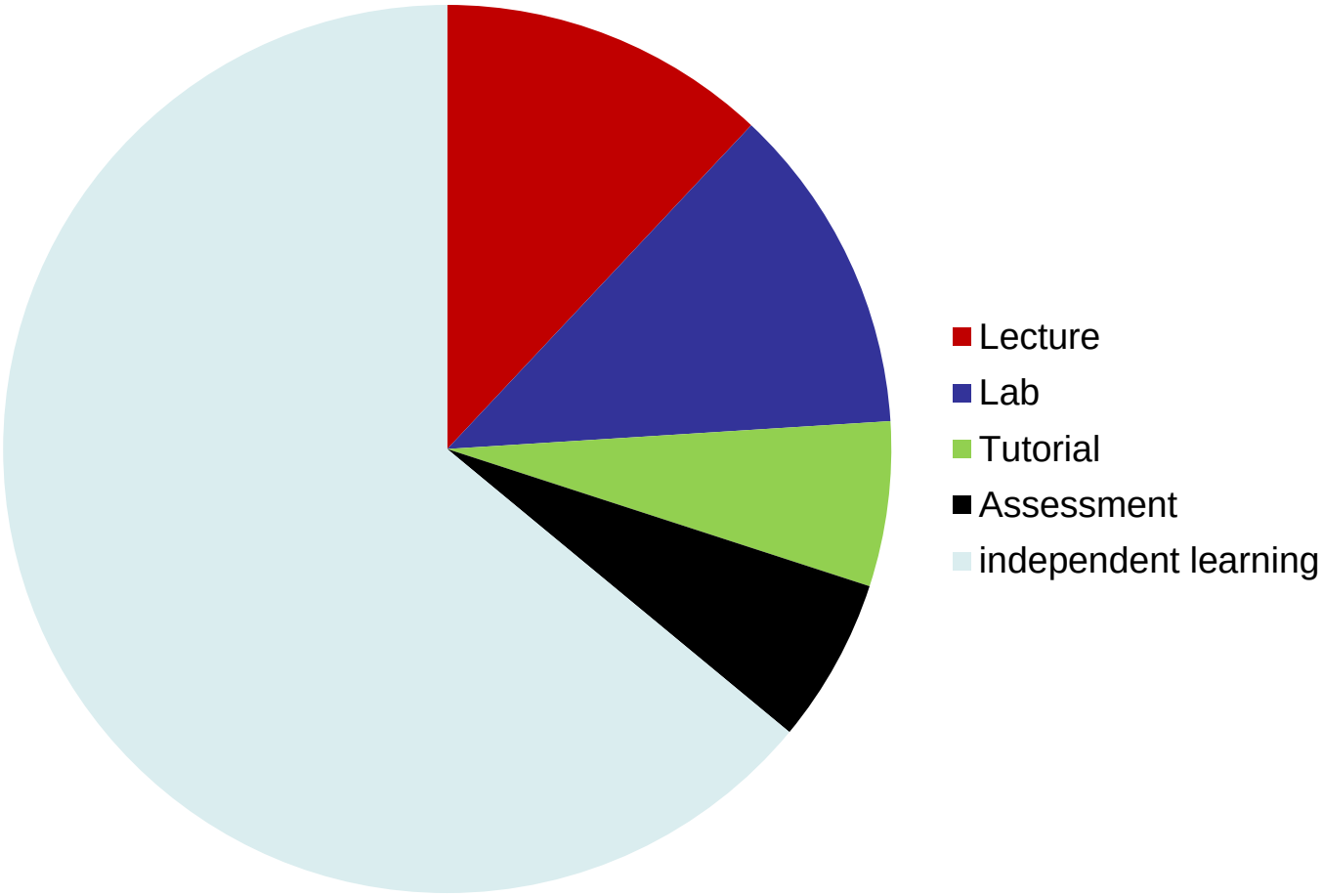
- http://www.intel.com/ Processor manufacturers
- http://www.amd.com/
- http://www.dell.com/uk/ PC manufacturers/ distributors
- http://www.viglen.co.uk/
- http://www.pcworld.co.uk/
- http://arstechnica.com/ guides
- http://www.tomshardware.com/

Booklist

Edinburgh Napier UNIVERSITY

- Paper books do tend to date quickly. You are probably better using online resources. Wikipedia sometimes gets some bad press in academic circles, but is a pretty good starting point. Here are some traditional books if you are interested:
  - R. Williams ‘Computer Systems Architecture, a networking approach’ Addison-Wesley/Pearson Education, 2nd Ed 2006, ISBN 978-0-321-34079-5 (Dated, but has the Pentium assembly language)
  - ‘Structured computer organization’, Tanenbaum, 0-13-020435-8.
  - ‘Computer organization and architecture’, Stallings, 0-13-081294-3.
  - ‘Computer architecture, design and performance’, Wilkinson, 0-13-518200-X.

Study Hours (200)



Edinburgh Napier UNIVERSITY

An Introduction to PC Hardware

Contents:

Processor.

Memory.

Busses.

I/O.

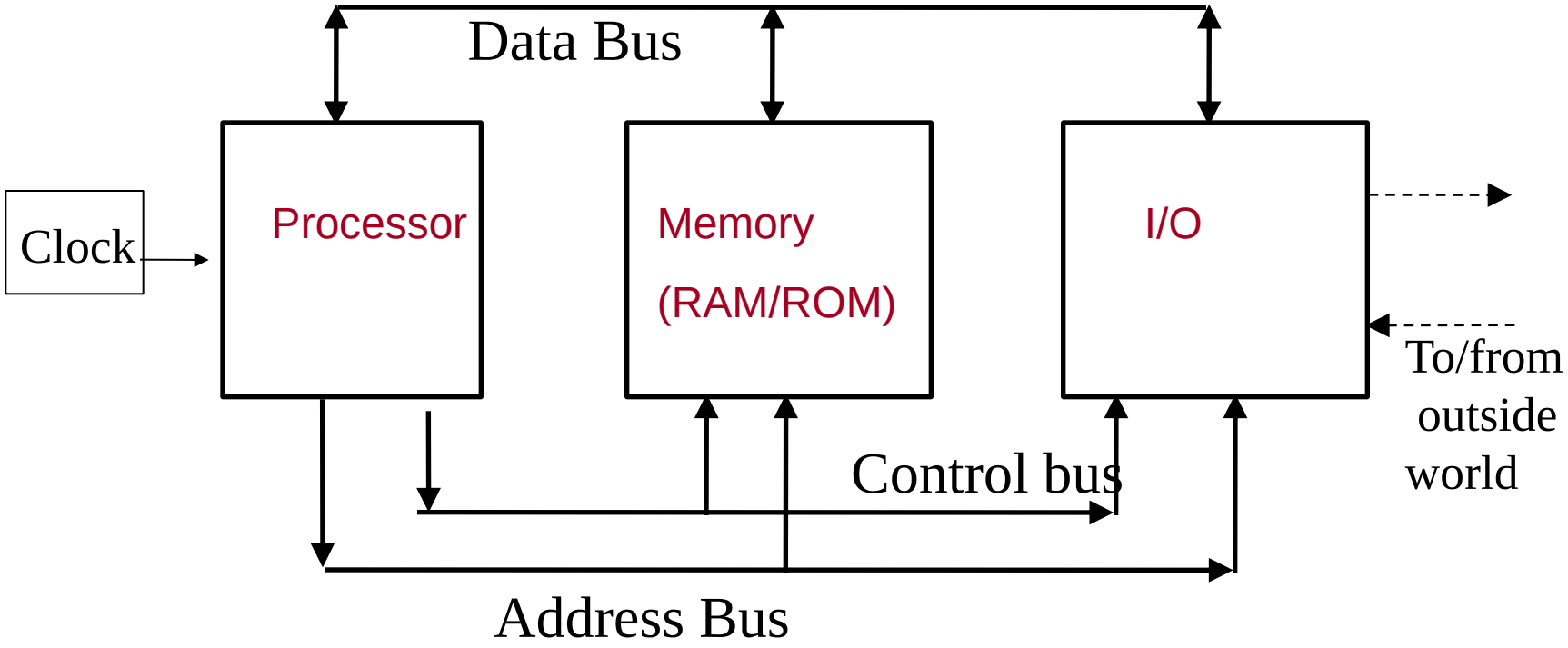
Overview

Edinburgh Napier UNIVERSITY

- This section will briefly introduce some of the main features of computer hardware.
- The main idea is to identify the key considerations in computer systems.
- One problem is that there is a wide range of computer systems, from small embedded microprocessors to complicated PCs. We will look at two examples: the 8051 family of low cost microcontrollers and the Pentium family used in most PCs.
- Many text books start with a simplified view of the components at the heart of the computer system.

Edinburgh Napier UNIVERSITY

The 'Text book' picture



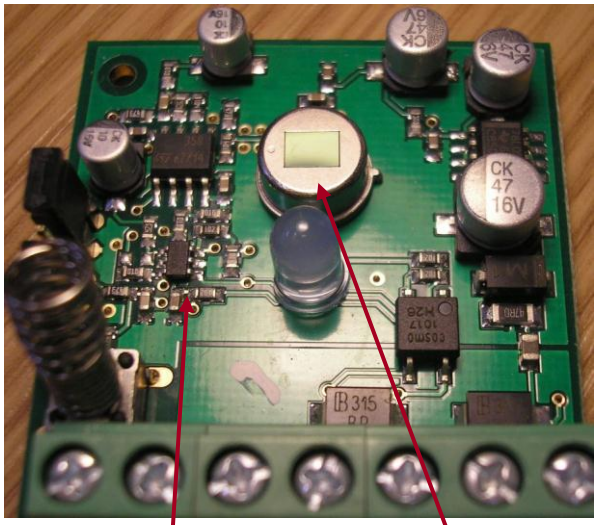
Edinburgh Napier UNIVERSITY

A simple text book microprocessor

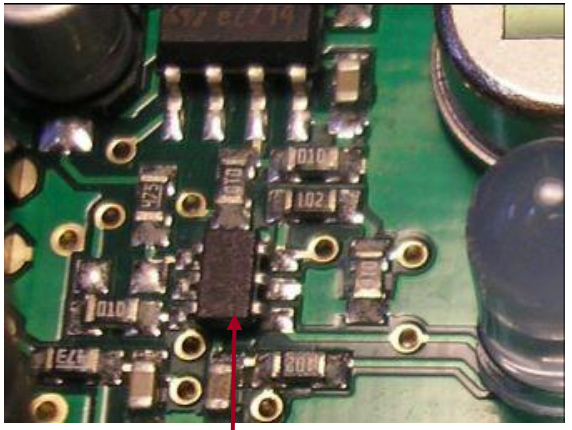
- The preceding picture shows, in an idealised form, the processor chip, and how it communicates with memory chips and peripherals.
- The processor communicates directly with the semiconductor memory chips (ROM & RAM).
- A simple embedded microprocessor could look quite similar to this simple picture (except that the three components might be combined into one chip).
- What a user sees of a PC is quite different:

A simple embedded system

A £15 PIR
security detector



Microcontroller & IR sensor



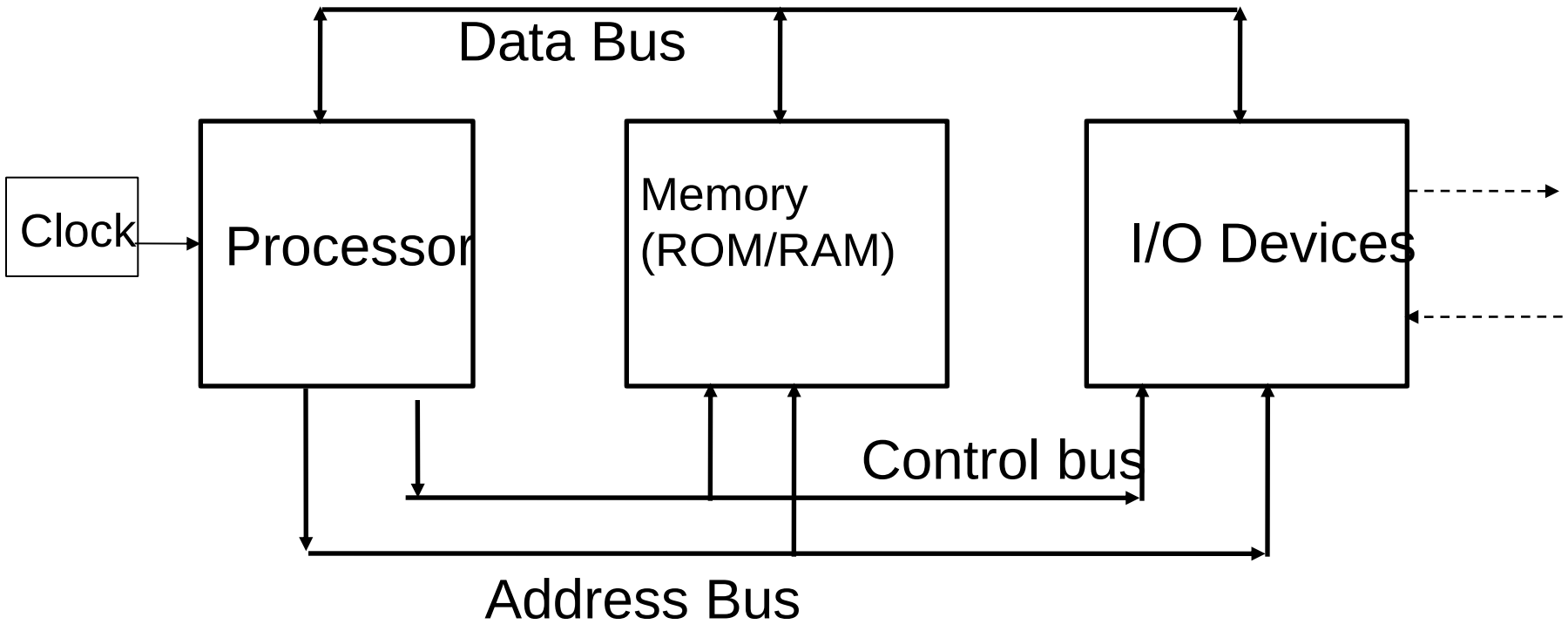
A 30p Microcontroller

Processors & Busses

Edinburgh Napier UNIVERSITY

- The processor is the heart of the system. In a simple embedded processor system the processor will control all the other circuits.
- The processor communicates with the rest of the system via buses. There are main types: Address, Data & Control.
- In a small embedded processor, these buses may be all there is.
- In a desktop PC, the situation is more complicated, with the same main types of bus, but different levels operating at different speeds.
- We will be looking in more detail at how PC systems use a hierarchy of buses.

Basic System Architecture



Clock

Edinburgh Napier UNIVERSITY

- The processor is driven by a fixed frequency clock. The frequency is determined by a crystal and is very accurate (as in wrist watches). What is the frequency of current PCs?

  2 – 3 GHz

- Would you know how to get the clock period (the length of a clock cycle) from the frequency?

  $T = 1/F$

- The clock frequency gives a rough measure of the performance of a computer, but is not the only factor determining performance.

Buses

Edinburgh Napier UNIVERSITY

- The Buses are the main means of communication within a processor system. Each bus is a collection of similar signals: usually conveyed by copper tracks and pins on the motherboard.
- The control bus is a collection of signals (mostly from the processor) by which the processor controls the rest of the system. For instance the processor may send signals to write to an Input/Output (I/O) device. Or, it may send signals to read from the memory chips.

Address bus

Edinburgh Napier UNIVERSITY

- The address bus is used by the processor to select a particular chip, and the location within a chip that is to be used.
- Normally, it is only the processor that generates addresses, so it is shown as a unidirectional bus.
- The more address lines there are, the more locations can be addressed. With 2 lines there are 4 combinations (00, 01, 10, 11).
- Each additional line doubles the number of combinations (adding a third line would give me the previous four combinations with a 1 at the front, and the same four with a 0 at the front).

Powers of 2

- Early processors had 16 address lines. This gave 2^{16} possible combinations of addresses i.e. 65536. This is usually referred to as 64 kbytes.
- When the PC first came out, it had 20 address lines. 2^{20} gives approximately 1 million combinations: 1 Mbyte.
- What is the current PC memory size?

Number of Address lines	Number of memory locations	Abbreviation
1	2	
8	256	
10	**1,024**	**1 K**
16	65,536	64 K
20	**1,048,576**	**1 M**
24	16,777,216	16 M
30	**1,073,741,824**	**1 G**
32	4,294,967,296	4 G

A Shortcut

Edinburgh Napier UNIVERSITY

- There is a simple pattern:
  - 10 lines give 1 K (roughly 1 thousand)
  - 20 lines give 1 M (roughly 1 million)
  - 30 lines give 1 G (roughly 1 billion)
  - 40 lines give 1 T (roughly 1 trillion)
- If you have a number nearby you can find the power of two from this. For instance, how many addresses do 22 lines give? (What is 2^{22}?)

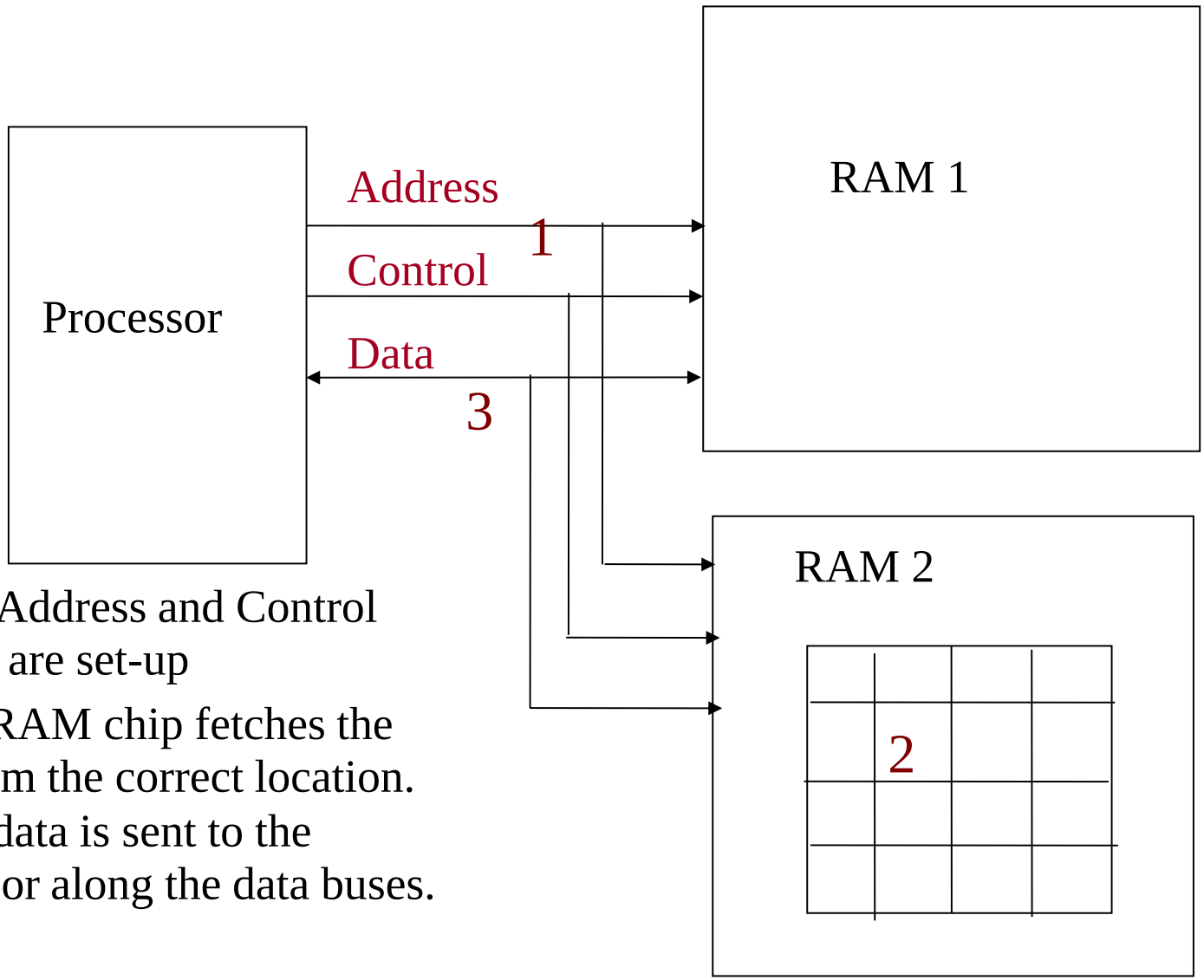
20 lines would give 1 M, so 21 would give 2 M, and 22 would give 4 M.

The data bus

Edinburgh Napier UNIVERSITY

- The data bus is used to transmit data from one point to another. Usually multiples of 8 bits are sent at a time. So, 32 bits of data may be sent together. This is a classic example of a parallel bus: data travels together in parallel.
- This mode of transfer only works well over short distances (say between chips on a circuit board). For longer distances there are problems with getting the data bits all to arrive at the same time. Instead, data is sent along a single line over long distances.

Example: reading from RAM



1. The Address and Control signals are set-up
2. The RAM chip fetches the data from the correct location.
3. The data is sent to the processor along the data buses.

Edinburgh Napier UNIVERSITY

Reads and Writes

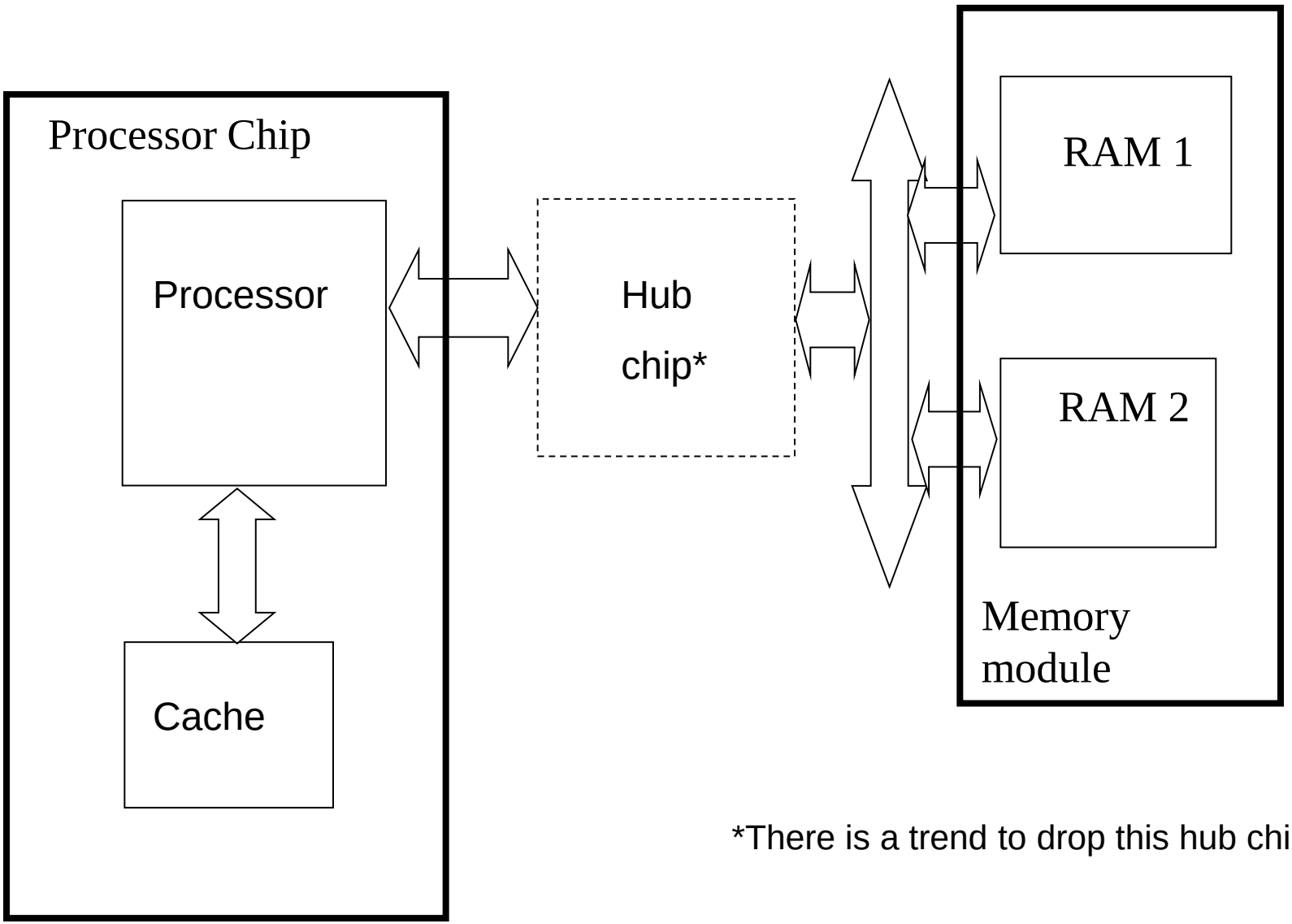
- In a PC data may go from RAM to the processor or vice versa. This is controlled by the processor using control lines.
- As everything is viewed from the point of view of the processor, data going out of RAM to the processor is a ‘Read’, Data going to the RAM from the processor is a ‘Write’.
- The processor has to read its instructions from memory before it can execute them: there are a lot of reads.
- Sometimes the instruction results in data being read from RAM, or stored in the RAM. This can result in traffic going either way.
- PCs use the classic ‘Von Neumann’ architecture: instructions and data are stored in one common block of RAM. What would be the implications of storing them in separate blocks (the ‘Harvard’ architecture)?

They could be accessed separately, but two sets of buses would be needed.

Data Transfer in a PC

Edinburgh Napier UNIVERSITY

- Data transfers to and from RAM are slightly different in a modern PC:
  - The processor comes in a module with local cache; many data transfers use this, and don't involve the RAM.
  - RAM chips come in a module containing a number of chips.
  - The processor transfers data via a 'hub' or 'bridge' chip that handles the memory (and graphics).
    - (Note in the diagram that all the buses have been combined into one to avoid cluttering the picture).



*There is a trend to drop this hub chip.

Edinburgh Napier UNIVERSITY

RAM and ROM

- There are two key types of memory chip. RAM is the main one used in PCs. It has the disadvantage that it is volatile (it loses data when the power is switched off). However, it can be made cheaply and has the advantage that data can be written to it very rapidly (a few nanoseconds). Because of this, it is the main type of memory used in PCs; there will be more on this later.
- ROM has the advantage that it doesn't lose data when power is removed. Depending on the type it can be slow, or impossible, to change the data stored in it. In PCs it is used to hold some low-level I/O routines (the BIOS) and a small amount of software that runs as the PC starts up (before the main software can be loaded from disk). Flash disks also use this, but these act like disk drives, not main memory.

Edinburgh Napier UNIVERSITY

The Fetch (Decode) Execute cycle

- Driven by the clock, the processor continually **fetches** instructions from memory.
- A special register, called the Program Counter or Instruction Pointer holds the location in memory (i.e. the address) of the “next instruction”.
- Once the instructions are inside the processor they are **decoded** to work out what needs to be done.
- They are then **executed**. This may involve different things: performing arithmetic, storing results, making decisions.
- This is referred to as the Fetch-Decode-Execute cycle (often abbreviated to Fetch-Execute).

The Execute Operations

Edinburgh Napier UNIVERSITY

- Instruction Execute:
  - The Instruction Decoder examines the contents of the Instruction Register and sends appropriate signals to other parts of the CPU to carry out the actions specified by the instruction.
  - There will be different types of instruction, so exactly what happens depends on the instruction. For instance, the instruction may involve arithmetic, and is executed entirely inside the processor. It may involve moving data in or out of the processor, using the buses again.
- This cycle of Fetch and Execute repeats indefinitely.

Example Instructions

Edinburgh Napier UNIVERSITY

- Reading operands from registers in the Arithmetic Logic Unit. Or reading them from the memory, using the buses.
- Causing the circuits of the Arithmetic Logic Unit to perform arithmetic or other computations.
- Storing data values from the ALU into registers. Or storing in memory using the the buses.
- Changing the value of the Program Counter (Jump or Branch instructions).

Instruction type 1a: the 8051

Edinburgh Napier UNIVERSITY

- Each instruction does a relatively simple thing, such as add a number to the contents of the A register.
- Typically it will have two sections: the 'opcode' and the operand. Opcodes are effectively verbs: MOV, ADD, INC (for MOVe, ADD, INCrement).
- The operand is the object(s) that are being worked on. As an example, there is an instruction to add the contents of the R6 register to the A register:

  ADD A,R6

- The instructions may take 1,2 or 3 bytes to store.

Edinburgh Napier UNIVERSITY

Instruction Type 1b: the Pentium

- A processor, such as the Pentium family, has many hundreds of possible instructions. Far more than the 8051.
- Some of these are very specialised (e.g. 3-dimensional graphics calculations).
- The instructions, when stored in memory, can take up different amounts of room (1-3 bytes).
- This type of processor. Like the 8051, is known as a Complex Instruction Set Computer (CISC).
- Every new generation of Pentium has to be able to run old software; this complicates new designs.

Questions

Edinburgh Napier UNIVERSITY

- Why do most people never need to learn the 8051 or Pentium instructions?
- Because compilers/interpreters/high-level libraries convert for us (e.g. java to Pentium instructions).
- Can an old compiler, written for an early version of the Pentium, still work for a new Pentium, with all its new instructions?
- Yes, but it won't take advantage of the new features. (biggest change was going from the 16-bit world (Win 3.1/8086) to 32 bits (80386, Win 95 or later).

Instruction Type – 2

Edinburgh Napier UNIVERSITY

- An alternative approach is to use a much smaller set of regular instructions. This is known as a Reduced Instruction Set Computer: RISC.
- Each is of a fixed length (usually 32 bits), making the fetch operation much simpler.
- The operations are simpler, so achieve less, but can run more quickly.
- Compilers have to be more sophisticated in order to do the same work with simpler instructions.
- This approach is used by the ARM processor in many mobile phones and tablets (actually many processors are hybrids).

Object code

Edinburgh Napier UNIVERSITY

- The code that runs on the processor is referred to as object code (or machine code). As different processor use different instructions, the object code for different processors varies considerably.
- In one family, such as the Pentiums, the code will be the same. However, Pentium code will not run on an 8051 (and vice versa).
- In the processor, the object code is stored as binary. This is too difficult for humans to understand, so is normally displayed as Hex.
- A slightly more friendly version is the use of Assembly language mnemonics.

Assembly language

Edinburgh Napier UNIVERSITY

ADDRESS	OBJECT	ASSEMBLY LANGUAGE
000F	7F05	MOV R7,#0X05
0011	7E1A	MOV R6,#0X1A
0013	EF	MOV A,R7
0014	2E	ADD A,R6

1. Move the number 5 into Register 7
2. Move the number 1A into Register 6
3. Copy the contents of R7 into A
4. Add the contents of R6 to A

This could have come from a line of C: x = 5 + 26;

The contents of RAM

Remember also that the code will be stored in binary; hex is just used to show the object code more compactly.