CS - 1.1 SYSTEMS ARCHITECTURE

KEY CONCEPTS

Computer systems take data (input), process it and then output it.

Embedded systems are computers built in to other devices like washing machines. They are dedicated to a single task so they are efficient.

Clock speed: the number of instructions a processor can carry out per/second. Higher clockspeed = faster CPU..

Number of **Cores:** The more cores a CPU has the more instructions it can carry out at once (multitasking). More cores = faster processing.

Cache size: A larger cache gives the CPU faster access to more data

EXAM QUESTIONS

Explain how cache size, cores and clockspeed affect the performance of the CPU.

Define what is meant by an embedded system

What is the purpose of the ALU?

Explain the role of the CPU registers (MAR and MDR)

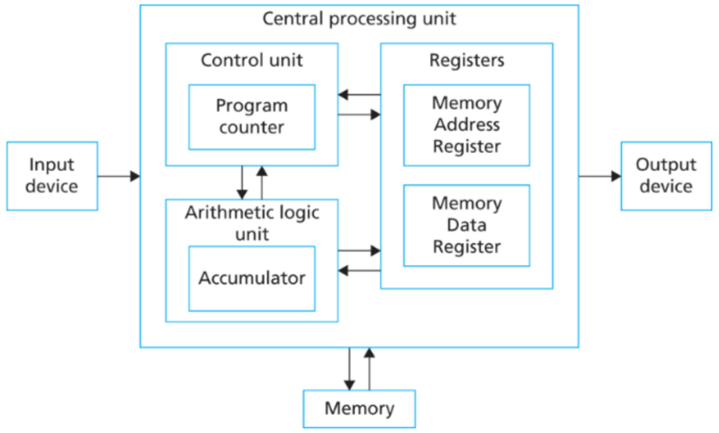
Explain how the fetch decode execute cycle works

Explain four events that occur during the FDE cycle.

FETCH – DECODE – EXECUTE CYCLE

CPU **fetches** instruction from the RAM (copies memory address to MAR, copies instruction to MDR & adds 1 to PC. CU **decodes** the instruction from the MDR Instruction is **executed** by the CU The next instructions is fetched and The cycle repeats.

THE CENTRAL PROCESSING UNIT (CPU)



Control Unit (CU): executes instructions and controls the flow of data in the CPU.

Program counter: holds the memory address for the instruction of each cycle.

Arithmetic Logic Unit (ALU): does all of the calculations and logic operations.

Accumulator: holds the immediate result of any calculations in the ALU.

Cache: very fast memory that stores regularly used data so that the CPU can access it quickly.

MAR (Memory Address Register): holds the address about to be used by the CPU.

MDR (Memory Data Register :) holds the actual data or instruction being processed by the CPU.

CS - 1.2 MEMORY and 1.3 STORAGE

RANDOM ACCESS MEMORY (RAM)

RAM is the computer’s main memory that holds the data, programs and files while they are being used.

RAM is volatile (power off = the data is lost)

The CPU will fetch instructions from the RAM in the fetch – decode – execute cycle.

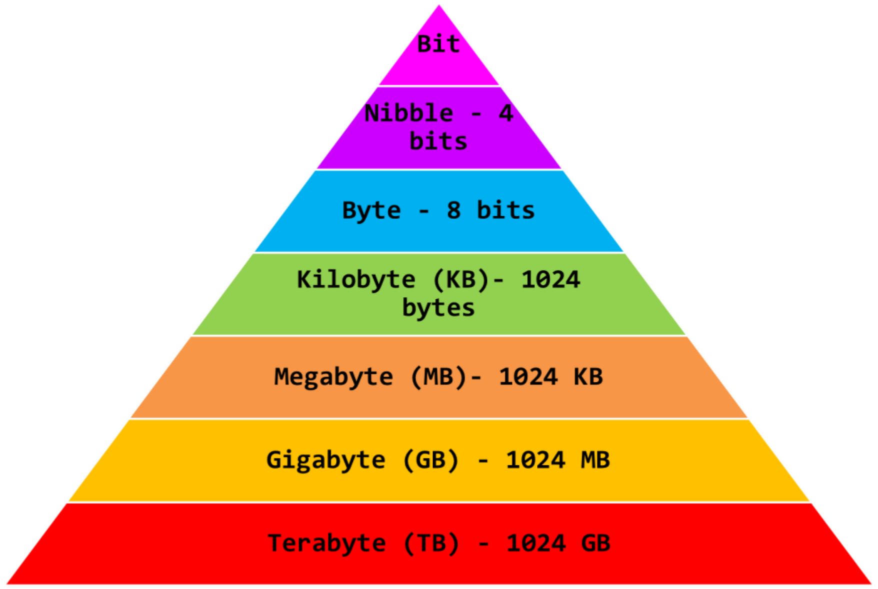
When the RAM is full the computer uses **VIRTUAL MEMORY**. It uses the secondary storage as temporary RAM so that the computer can continue running (but slowly).

READ ONLY MEMORY (ROM)

The ROM is on a chip build into the motherboard

It contains the BIOS (boot up sequence for the computer)

ROM is non-volatile (data still stored after power is off)



Bit

Nibble - 4 bits

Byte - 8 bits

Kilobyte (KB)- 1024 bytes

Megabyte (MB)- 1024 KB

Gigabyte (GB) - 1024 MB

Terabyte (TB) - 1024 GB

STORAGE CAPACITY

Some storage methods such as a HDD or SSD have a

TYPES OF STORAGE

Secondary Storage: where all data including the programs are stored when they are not being used.

Storage	Key Information
Hard Disk Drive (HDD)	Magnetic, has moving parts, large capacity, lower cost than SSD
Solid State Drive (SSD)	Flash memory, no moving parts, more robust than HDD, faster and more expensive than HDD
Flash memory	Eg: USB memory sticks, memory cards.
Optical Storage	Eg: CDs, DVDs. Cheap, portable and fairly robust.
Magnetic tape	Used for archive storage (back ups). Very large capacity, low cost, slow.

EXAM QUESTIONS

Explain how the RAM works with the CPU in the fetch –decode – execute cycle

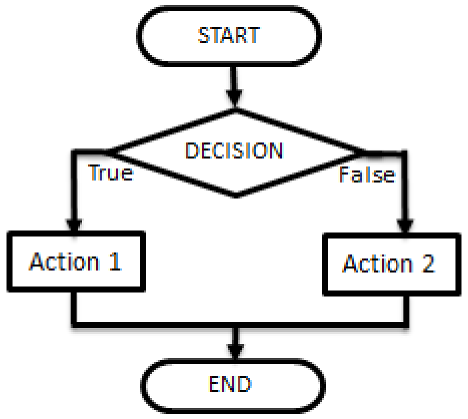
Explain the difference between volatile and non-volatile memory giving an example of each

Tom is buying a new laptop, he is not sure whether to get a magnetic HDD or SSD. Discuss the benefits and drawbacks of each.

CS - 2.1 ALGORITHMS

COMPUTATIONAL THINKING

Abstraction	•Focussing on just the important details of a problem
Decomposition	•Breaking a problem down into smaller parts so that it is easier to solve
Algorithmic thinking	•creating a step by step solution to a problem



PSEUDOCODE

```
START
IF the Decision = TRUE
THEN:
        Perform Action 1
ELSE
        Perform Action 2
ENDIF
END
```

SEARCHING ALGORITHMS

To find an item in a list, computers need to use a searching algorithm. A linear search and binary search are both examples of sorting algorithms.

Linear Search: Checks each item in the list one by one until it finds what it is looking for
+ Simple, list doesn't need to be ordered
- Not efficient, takes time with lots of data

Binary Search: Finds the middle item in an ordered list by doing (n+1)/2. IF the middle item is what it is searching for it stops. If not, it compares the item you are searching for to the middle item so that it knows whether to look in the first half or second half of the list. Then it repeats these steps until the item is found
+ More efficient than a linear search
- Only works on an ordered list, complex to program

SORTING ALGORITHMS

Sorting algorithms sort items into an ordered list.

Bubble Sort: Checks the first two items in a list, swaps them if they are in the wrong order and then moves onto the next two items and repeats the process. Once it has passed through the list once it goes through again until none of the items need swapping. + Simple. – Takes a long time

Merge Sort: Finds the middle item (n+1)/2 and splits the list in half. Repeats this step until the list is split into individual items (sub-lists). It them merges (joins) the sublists in pairs. Each time the sublists are paired they are sorted into the correct order. + Efficient – Slow

Insertion Sort: Looks at the second item in a list and compares it to the items that are in front of it, then inserts it into the right place. It then moves to the next item in the list and repeats these steps. + Quick for sorting small lists – slow with long lists

CS - 2.2 PROGRAMMING TECHNIQUES

DATA TYPES

Data Type	Definition
String	Text eg: “Hello”
Integer	Whole number eg: 32
Float/ Real	Decimal number eg: 1.2
Boolean	Two values eg: true or false
Character	A single character eg: b

Casting is when you want to change between data types. Eg – if you want to use

VARIABLES AND CONSTANTS

Variable – A value which may change while the program is running. Variables can be local or global.

Local Variable – a variable which can only be used within the structure they are declared in.

Global Variable – a variable which can be used in any part of the code after they are declared

Constant – A value which cannot be altered as the program is running.

OPERATORS

Operator/ Function	Definition
Exponentiation	Raises a number to a power eg: 2**3 OR 2 ^3 (=2^3)
Quotient/DIV	Gives the whole number after a division
Remainder/MOD	Gives the remainder part of a division
==	Is equal to
! or <>	Is not equal to
<	Is less than
>	Is more than
>=	Is more than or equal to
<=	Is less than or equal to

ARRAYS

One-Dimensional Arrays- this is like a list. In this example an array has been created called students. The list can hold 3 items (as shown).

```
array students [3]
students [0] = “Bob”
students [1] = “Dave”
students [2] = “Bob”
```

```
print(students[1])
```

This command would print the second item (1) From the array. It would print “Dave”.

Two-Dimensional Arrays – these are lists within lists (like a table)

```
Grades=[[“Bob”, “22%”, “44%”],
[“Dave”, “85%”, “100%”]]
```

	0	1	2
0	Bob	22%	44%
1	Dave	85%	100%

```
print(“Bob’s first test score was “ + Grades [0, 1]
```