Review

R103 – Binary, Numbers, Dry Run

Starter – CompArch Recap

Define the following components of general purpose computers:

1. System clock
2. Data bus
3. Address bus
4. Register
5. General purpose register

Extension: Can you name all 5 of the special purpose registers used in the FDE cycle?

Processors 3

Factors that affect performance

Specification

4.7.3.6 Interrupts

Content	Additional information
Describe the role of interrupts and interrupt service routines (ISRs); their effect on the Fetch-Execute cycle; and the need to save the volatile environment while the interrupt is being serviced.	

4.7.3.7 Factors affecting processor performance

Content	Additional information
Explain the effect on processor performance of: • multiple cores • cache memory • clock speed • word length • address bus width • data bus width.	

Processor performance

In the context of processors, what does **performance** mean? Discuss in pairs.

- What does high performance look like?
- What about low performance?
- Is there more than one definition?

Processor performance

- **Speed/throughput**: number of instructions executed per second

Processor performance

- **Speed/throughput**: number of instructions executed per second
- **Scalability**: could this processor cope with, say, more memory?

Processor performance

- **Speed/throughput**: number of instructions executed per second
- **Scalability**: could this processor cope with, say, more memory?
- **Response time**: how long does it take for a single request to be completed?

Processor performance

- **Speed/throughput**: number of instructions executed per second
- **Scalability**: could this processor cope with, say, more memory?
- **Response time**: how long does it take for a single request to be completed?
- **Power consumption**
- Many more...

Effect on performance

Write down 3 factors that affect processor performance that you researched as part of your pre-work.

Clock speed

- A quartz-controlled oscillator supplies a fixed rate timing signal – this is the **system clock**

Clock speed

- A quartz-controlled oscillator supplies a fixed rate timing signal – this is the **system clock**
- **Clock speed** is the frequency at which the system clock ticks

Clock speed

- A quartz-controlled oscillator supplies a fixed rate timing signal – this is the **system clock**
- **Clock speed** is the frequency at which the system clock ticks
- A processor executes a particular instruction in a set number of **clock ticks** or **clock cycles**
- Set by manufacturer, but can be increased by ‘overclocking’

Effects of increasing clock speed

- Enables more instructions to be executed per unit of time (second)
  - Double the clock speed means double the number of instructions being executed per second

Effects of increasing clock speed

- Enables more instructions to be executed per unit of time (second)
  - Double the clock speed means double the number of instructions being executed per second
- Heat dissipation issues increase as clock speed increases

Effects of increasing clock speed

- Enables more instructions to be executed per unit of time (second)
  - Double the clock speed means double the number of instructions being executed per second
- Heat dissipation issues increase as clock speed increases
- There are limits to this
  - Speed of electricity
  - All machine code instructions take a minimum amount of time to execute

Word

- A **word** is a fixed-size unit of data used across the computer
- The **word length** is the number of bits in a word; it varies between computers

Word

- A **word** is a fixed-size unit of data used across the computer
- The **word length** is the number of bits in a word; it varies between computers
- Typically either 32 or 64 bits
- Registers are usually one word in size
- Memory addresses are often one word in size
- Word length affects instruction operand lengths too

Increasing the word length

- Larger word size means that the registers can store more bits
  - This means the processor can process more bits at once
- It also means that more space is wasted when storing small values

Increasing the word length

- Larger word size means that the registers can store more bits
  - This means the processor can process more bits at once
- It also means that more space is wasted when storing small values
- If the size of an instruction operand of a memory value was greater than the word length, then complicated value-splitting would need to happen

Bus width

- The number of parallel signal lines or wires allocated to a bus
- Don’t forget that there are different types of buses

Increasing address bus width

- Enables the processor to access a larger number of main memory locations, so there is less need to use secondary storage
- **Scalability** increases and **speed** increases under high memory usage

Increasing data bus width

- Enables more bits to be transferred between the processor and main memory at one time
  - Fewer read/write operations are needed
- If the data bus is not large enough to hold a whole instruction, then the time to fetch it is doubled (at least) as the instruction needs to be split

Cache memory

- A small amount of very fast memory that stores
  - copies of data from frequently used main memory locations
  - data to be written to main memory
  - pre-fetched instructions

Increasing the amount of cache memory

- Cache memory is faster to access than main memory
- The more that can be stored in cache memory, the less frequently main memory needs to be accessed
- Increases **speed** and **response time**

Multicore processors

- Increase the actual number of processors, but keep the clock speeds low to minimise heat
- Multiple tasks can be run at the same time, or a single task can be split over several processors
- Increases **throughput**, but greatly increases **complexity**

Other ideas

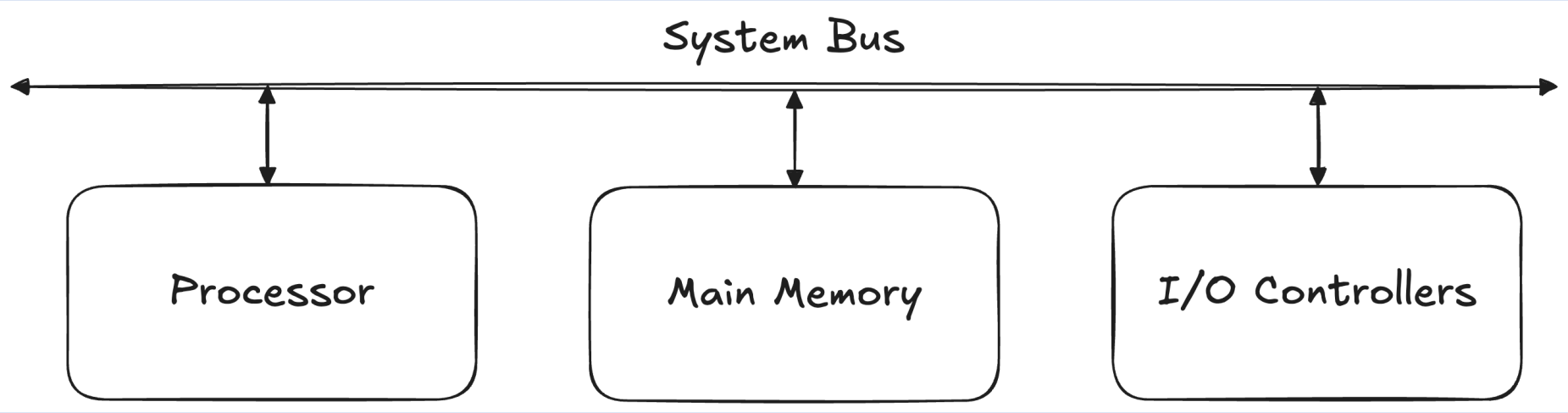
- Increase the number of **general purpose registers** so that more intermediate results/variables can be kept in processor registers rather than in main memory
- Change the **type of cache memory** – some types can be accessed faster
- **Pipelined processors** allow the next instruction to be in the ‘fetch’ stage while the current instruction is in the ‘execute stage’. Increases **throughput**

Recap

Draw the three-box model

Recap

Draw the three-box model



Processor interrupts

- When your program executes the line `Console.ReadKey()`, it tells the processor to do something else more useful until a key is pressed
  - How does it know when a key is pressed?

Processor interrupts

- When your program executes the line `Console.ReadKey()`, it tells the processor to do something else more useful until a key is pressed
  - How does it know when a key is pressed?
- An **interrupt** is a signal that is sent to the processor to request its immediate attention

Processor interrupts

- When your program executes the line `Console.ReadKey()`, it tells the processor to do something else more useful until a key is pressed
  - How does it know when a key is pressed?
- An **interrupt** is a signal that is sent to the processor to request its immediate attention
- When an interrupt is received, the processor suspends what it is doing and runs the process associated with the interrupt

Examples of processor interrupts

- A hardware device has data ready to process
- A device has completed a task it was asked to do
- A process needs a function to be performed
- A pre-set timer has expired
- A hardware failure has occurred

Handling processor interrupts

- For every ‘kind’ of interrupt, the operating system has an **interrupt service routine/ISR/interrupt handler**
- These are small programs designed to respond to each interrupt
- They can be provided by either the operating system itself or I/O device drivers

Handling processor interrupts

- For every ‘kind’ of interrupt, the operating system has an **interrupt service routine/ISR/interrupt handler**
- These are small programs designed to respond to each interrupt
- They can be provided by either the operating system itself or I/O device drivers
- Consider the example of a key being pressed
  - The ISR will determine which key was pressed by communicating with the keyboard via its I/O controller
  - The pressed key will be stored in the keyboard buffer
  - A process waiting for keyboard input can later check the buffer to find out which key was pressed

How the fetch-execute cycle is affected

- The fetch-execute cycle needs to be paused for an interrupt to be serviced

How the fetch-execute cycle is affected

- The fetch-execute cycle needs to be paused for an interrupt to be serviced
- The cycle for the current instruction is completed, and then the volatile environment for the running process is saved
  - The **volatile environment** of a process is a snapshot of the information needed to execute a process from its current instruction. This includes special purpose and general purpose registers (including the Program Counter)

How the fetch-execute cycle is affected

- The fetch-execute cycle needs to be paused for an interrupt to be serviced
- The cycle for the current instruction is completed, and then the volatile environment for the running process is saved
  - The **volatile environment** of a process is a snapshot of the information needed to execute a process from its current instruction. This includes special purpose and general purpose registers (including the Program Counter)
- Then the ISR runs, after which the volatile environment is loaded back into the processor and the fetch-execute cycle resumes

Procedures with Parameters 2

Starter

True or false?

1. You cannot have a local variable with the same name as a global variable
2. Local variables are only visible to the code within the procedure that they are defined
3. Subroutine is another word for procedure
4. A call to a procedure is a statement in its own right

Starter

True or false?

1. You cannot have a local variable with the same name as a global variable False
2. Local variables are only visible to the code within the procedure that they are defined
3. Subroutine is another word for procedure
4. A call to a procedure is a statement in its own right

Starter

True or false?

1. You cannot have a local variable with the same name as a global variable False
2. Local variables are only visible to the code within the procedure that they are defined True
3. Subroutine is another word for procedure
4. A call to a procedure is a statement in its own right

Starter

True or false?

1. You cannot have a local variable with the same name as a global variable False
2. Local variables are only visible to the code within the procedure that they are defined True
3. Subroutine is another word for procedure True
4. A call to a procedure is a statement in its own right

Starter

True or false?

1. You cannot have a local variable with the same name as a global variable False
2. Local variables are only visible to the code within the procedure that they are defined True
3. Subroutine is another word for procedure True
4. A call to a procedure is a statement in its own right True

Starter

Identify the 6 errors in the following code.

```
class Program
{
  static void Main(string num)
  {
    int total;
    for (i = 0; i < num; i++)
    {
      total += i
    }
    Console.WriteLine(total);
   }
  static void Main(string[] args)
  {
   Console.Write("Enter num: ");
   int input = Console.ReadLine();
   Main(input);
   Console.ReadKey();
  }
}
```

PRIMM

- Predict
- Run
- Investigate
- Modify
- Make

Subroutines with Parameters PRIMM

This worksheet is on Google Classroom.