Computer Architectures

What the specification says

describe classic Von Neumann architecture, identifying the need for, and the uses of, special registers in the functioning of a processor;
describe, in simple terms, the fetch/decode/execute cycle, and the effects of the stages of he cycle on specific registers;
discuss co-processor, parallel processor and array processor systems, their uses, advantages and disadvantages;
describe and distinguish between Reduced Instruction Set Computer (RISC) and Complex instruction Set Computer (CISC) architectures.

Von Neumann Architectures

John Von Neumann realised that the programs and their data were indistinguishable, and therefore can be stored in the same memory. He created a new architecture that contained a single processor which follows a linear sequence of the fetch-decode-execute cycle.

In order to do this it has a few special registers. A register is just an area to store data. The individual locations in the memory are registers, but as they have no special purpose they are not special registers. There are five special registers that are used to control the fetch-decode-execute cycle. They are outlined as below:

- PC - Program Counter
- CIR - Current Instruction Register
- MAR - Memory Address Register
- MDR - Memory Data Register
- Accumulator

The specification requires you to know the fetch-decode-execute cycle and how it links in with the special registers in some detail

Fetch

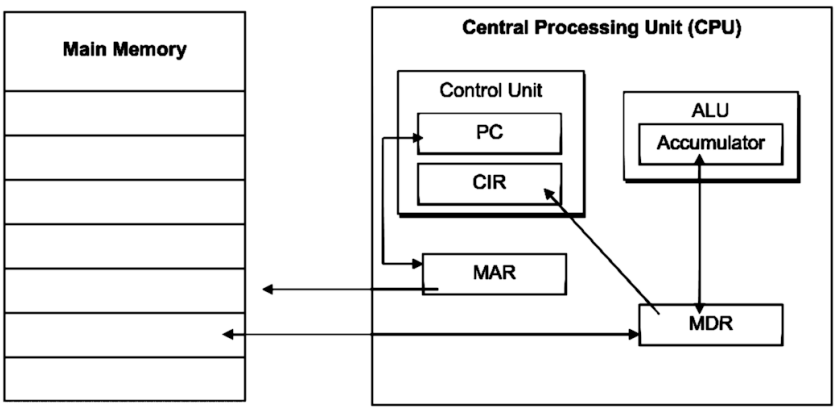
1. The PC stores the address of the next instruction which needs to be carried out
   As instructions are held sequentially in the memory, the value in the PC is incremented so that it always points to the next instruction.
2. When the next instruction is needed, it's address is copped from the PC and placed in the MAR
3. The data which is stored at the address in the MAR is then copied to the MDR
4. Once it is ready to be executed, the executable part if the instruction is copied into the CIR

Decode

1. The instruction in the CIR can now be split into two parts, the address and the operation
2. The address part can be placed in the MDR and the data fetched and put in the MAR.

Execute

1. The contents of both the memory address register and the memory data register are sent together to the central processor. The central processor contains all the parts that do the calculations, the main part being the CU (control unit) and the ALU (arithmetic logic unit), there are more parts to the central processor which have specific purposes as well.
2. The ALU will keep referring back to where the data and instructions are stored, while it is executing them, the MDR acts like a buffer, storing the data until it is needed
3. The CU will then follow the instructions, which will tell it where to fetch the data from, it will read the data and send the necessary signals to other parts of the computer.



Fetch-Decode-Execute Cycle

All instructions on the computer for nearly all types of architecture follow the fetch-decode-execute cycle. It is a simple principle developed in the 1940's. The specification requires you to know it in some detail, and know how each stage makes use of the special registers. It is outlined below:

1. Load the address that is in the program counter (PC) into the memory address register (MAR).
2. Increment the PC by 1.
3. Load the instruction that is in the memory address given by the MAR into the MDR
4. Load the instruction that is now in the MDR into the current instruction register (CIR).
5. Decode the instruction that is in the CIR.
6. If the instruction is a jump instruction then
    a. Load the address part of the instruction into the PC
    b. Reset by going to step 1.
7. Execute the instruction.
8. Reset by going to step 1.

Reduced Instruction Set and Complex Instruction Set Computers

When you used binary to represent instructions, there were two parts to each byte, the instruction and the data. For example 00110100, if the firs 3 bits represented the instruction, like ADD, SUB, DIV, MOD… or whatever, then the last 5 bits would represent the data address. There would only be nine possible instructions available though. This would be a very simple set, the more bits allowed for the operation, the more operations available and therefore the more complex the set becomes.

A complex instruction set computer (CISC) is designed to have operations and operation code for all things they will need to do. It needs to have slightly more complex physical architecture to allow for this wide range of possible instructions. It is easier to program, and requires less code. Because the instructions are already in existence, there is no need to store intermediate results, this uses less RAM. However the processor may need to use more cycles per instruction because the instructions are more complex.

Some processors however are designed to reduce the number of operations which saves space. These are called restricted (or reduced) instruction set computers (RISC). Just because they have fewer instructions doesn't mean they can do fewer things, it just means that the programmer has to be a bit cleverer. The physical architecture is therefore simpler because there are only a few assembly instructions which can be used. Also because each job will be made up of quite a lot of basic instructions, it is necessary to store intermediate results, which is half processed data, as a result more RAM will be required. Because each instruction is simple, it only takes one cycle by the processor to complete the instruction. Recently this has become a more preferred method, as overall it is more efficient, because the only two draw backs are using more RAM, which doesn't matter as it is getting increasingly cheaper, and that it is harder to program, but more tools are being developed which makes the programs easier to write.

Parallel Processing Architectures

The Von Neumann architecture is an example of serial processing, where one instruction is fetched, decoded and executed and then the next is fetched, decoded and executed and so on. Parallel processing however allows many instructions to be carried out at the same time. There are a number of different ways of doing this, depending on the use.

Array or Vector processing

This is a simple processor used for data that can be processed independently of one another – that means that a single instruction can be applied to multiple *bits* of data all at the same time, and none of the results of that data will be needed to process the next bit of data.

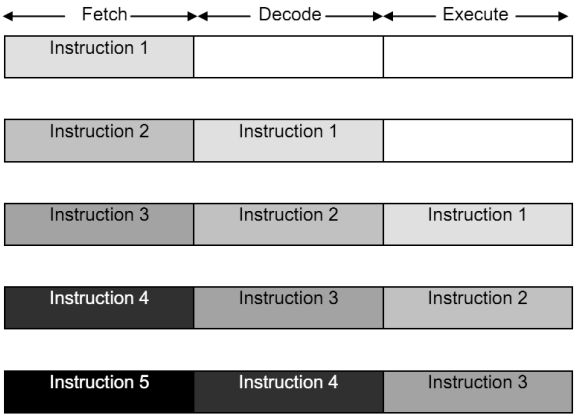
These types of processors are normally used for things like controlling input or output devices. They can be used to track the point of the mouse on a screen, or display the data on the monitor.

They are sometimes called array processors, because the data is stored in an array, similar to that used in programing, the array can have one to many dimensions. Array processors are an extension to the CPU's arithmetic unit. The only disadvantage to array processing is that it relies on the data sets all acting on the same instruction, and it is impossible to use the results of one data set to process the next, although this does not matter in their use.

It is also important to know, that an array processor is a single instruction multiple data (SIMD) processor, it should be clear what this means, lots of bits of data get processed with a single instruction.

Pipelining

Pipelining is another method of parallel processing. There are several processors each one does a different part of the fetch decode execute cycle, so the fetch-decode-execute cycle is staggered. This can be best illustrated with the diagram on the right.



As long as the pipelines can be kept full, it is making best use of the CPU. This is an example of single instruction single data (SISD) processor, again it should be quite clear why, the processor is processing a single instruction to a single bit of data.

Multi-core processors

This is where there a number of CPUs being applied to a job, with each part carrying out different parts. Each CPU is likely to be in effect a serial processor, although as there are many processing multiple instructions to multiple data it becomes a parallel processor when viewed as a whole.

These are likely to be used on a large scale in supercomputers, but also many personal computers have multiple cores. The limitation of multi-core processors would be that they are dependent on being able to cut jobs down into chunks, this can make it harder for the programmer to write code for them.

Processor Examples

This does not come into the specification, although it puts the above information into context and makes it easier to understand, it's also very interesting.

Vector processing

The Cray supercomputers all work with vector processors to an extreme, although they are much more expensive than alternative processing methods, and have limitations of what type of data they can process. Array processing is also used on a smaller scale for some I/O devices and games graphics.

Difference between code for serial processors and array/ vector processors

Serial Processor

```
execute this loop 10 times
  read the next instruction and decode it
  fetch this number
  fetch that number
  add them
  put the result here
end loop
```

Array/ Vector processor

```
read instruction and decode it
fetch these 10 numbers
fetch those 10 numbers
add them
put the results here
```

Multi-core supercomputer - Blue Gene

Blue Gene is one of the most powerful computers in the world, IBM started developing it in 1999, costing $100 million in research, and taking 5 years to complete, it became the world's most powerful supercomputer in 2004 (it is now 5th ranked by processing power). It is mainly used by universities and government research departments. Blue Gene is a good example of a multi-core computer to an extreme; it has 1496 cores (most home computers have about 2 cores). Supercomputers work basically the same was as normal multi-core processors, just to a much much larger scale, allowing for very big and accurate calculations to be mad quickly.

Top most powerful multi-core processor computers in the world since 1993

- Fujitsu K computer (Japan, June 2011 – present)
- NUDT Tianhe-1A (China, November 2010 - June 2011)
- Cray Jaguar (United States, November 2009 - November 2010)
- IBM Roadrunner (United States, June 2008 – November 2009)
- IBM Blue Gene/L (United States, November 2004 – June 2008)
- NEC Earth Simulator (Japan, June 2002 – November 2004)
- IBM ASCI White (United States, November 2000 – June 2002)
- Intel ASCI Red (United States, June 1997 – November 2000)
- Hitachi CP-PACS (Japan, November 1996 – June 1997)
- Hitachi SR2201 (Japan, June 1996 – November 1996)
- Fujitsu Numerical Wind Tunnel (Japan, November 1994 – June 1996)
- Intel Paragon XP/S140 (United States, June 1994 – November 1994)
- Fujitsu Numerical Wind Tunnel (Japan, November 1993 – June 1994)
- TMC CM-5 (United States, June 1993 – November 1993)

Von Neumann Architecture

- The most commonly used computer architecture, is serial, and so only processes one job at a time with one set of data
- All the instructions and the data in Von Neumann architecture are stored together in the same memory.
- In order to do this it makes use of the special registers, which are described on the next page.

Advantages	Disadvantages
▪ Nearly all types of data can be processed with the VN architecture ▪ Data that relies on the result on the previous operation is fine Cheaper than alternative methods of processing	▪ Can be slower than alternative methods ▪ Can be limited by the bus transfer rate ▪ Doesn't always make maximum use of the CPU ▪ Poorly written programs can get their data mixed up, because both programs and data share the same memory

PROCESSOR COMPONENTS

PC → MAR → MDR → CIR → ACC

PROGRAM COUNTER (PC)

- KEEPS CHECK OF WHEREABOUTS THE NEXT PROGRAM IS IN THE MEMORY.
- AFFTER ONE INSTRUCTION HAS BEEN CARRIED OUT, THE PC WILL BE ABBLE TO TELL THE PROCESSOR WHERE ABOUTS THE NEXT INSTRUCTION IS.
- THE INSTRUCTIOIONS ARE ALWAYS STORED IN ORDER IN THE PC.

MEMORY ADDRESS REGISTER (MAR)

- THIS IS WHERE THE ADDRESS THAT WAS READ FROM THE PC IS SENT.
- STORED HERE SO THAT THE PROCESSOR KNOWS WHERE ABOUTS IN THE MEMORY THE INSTRUCTION IS.

MEMORY DATA REGISTER (MDR)

- THE MEMORY IS SEARCHED TO FIND THE ADDRESS BEING HELD IN THE MAR, AND WHAT EVER IS UNDER THAT ADDRESS, MUST BE THE INSTRUCTION.
- THE INSTRUCTION IS THEN COPPIED, INTO THE MDR.

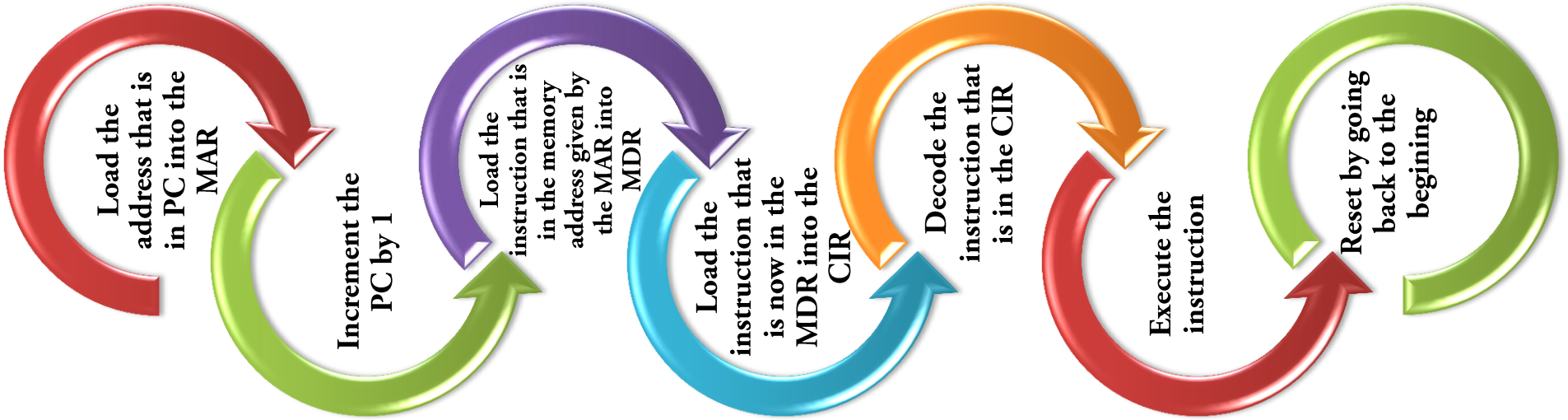
CURRENT INSTRUCTION REGISTER (CIR)

- THE INSTRUCTION THAT IS NOW IN THE MDR IS COPPIED INTO THE CIR.
- IT WILL BE SPLIT INTO TWO PARTS:
  - ONE PART WILL BE SENT TO THE COMPUTER TO BE DECODED SO THAT THE PROCESSOR KNOWS WHAT SORT OF INSTRUCTION IT IS, AND CAN SEND SIGNALS TO THE RELEVANT PARTS.
  - THE OTHER ART TELLS THE PROCESSOR WHERE ABOUTS IN THE MEMORY THE DATA THAT NEEDS TO BE USED IS.

THE ACCUMALATOR (ACC)

- THE ACCUMALATOR IS USED TO ACCUMALATE RESULTS, IT IS WHERE THE RESULTS FROM OTHER OPERSAIONS ARE STORED TEMPORARILY BEFORE BEING USED BY OTHER PROCESS'S.

The Fetch Decode Execute Cycle



Pipelining

Multiple CPUs carrying out a different part of the fetch-decode-execute cycle.

As long as it is kept full, makes maximum use of CPU

Single Instruction Single Data Computer (SISD)

Array Processing

Where data is arranged into arrays and all acted upon by the same instruction

Relies on the fact that all the sets of data are being acted on by the same instruction

Single Instruction Multiple Data Computer (SIMD)

Multi-core processing

Where several serial processors operate simultaniously on different parts of a job

Dependant on being able to cut problems down into chunks

Multiple Instruction Multiple Data Computer (MIMD)

SERIAL AS OPPOSED TO PARALLEL

Advantages

- Nearly all programs can run on with serial processors, there is no additional complex code to be written.
- All types of data are suitable for serial processing
- Data can use the previous result in the next operation
- Data sets are independant of each other
- Cheaper than parralel

Disadva

- The Von-Neumann bottle neck can slow data transfer in VN architectures.
- Both the program and data share the same memory, it is possible that trashing can occur with poorley written programs

PARALLEL AS OPPOSED TO SERIAL

Advantages

- Faster when handling large amounts of data
- Not limited by bus transfer rate
- Can make maximum use of the CPU
- Programs and data are stored seperatley usually

Disadva

- Only certain types of data are suitable for parralel processing
- Data that relies on the result of the previous operation can not be made parralel
- Each data set must be inderpendant of each other
- Usually more expensive

Past Exam Questions

Below are all the exam questions relating to computer architectures since the GCE computing from 2008 specification was released. The mark scheme answers and explanations are below.

Question 1

(a) Describe the effects of the fetch-execute cycle on the program counter (PC) and the memory address register (MAR).

[5]

(b) (i) State three features of a Complex Instruction Set Computer (CISC) architecture.

1

2

3

[3]

(ii) Explain one disadvantage, other than cost, of a CISC architecture compared with a Reduced Instruction Set Computer (RISC) architecture.

[2]

Question 2

In classic Von Neumann architecture, a number of registers are used.

(a) (i) Explain the term register.

[2]

(a)(ii) Give the correct names for two of the special registers used. (Do not use abbreviations.)

1

2

[2]

(b) Explain the advantages and disadvantages of parallel processor architecture compared with Von Neumann architecture.

[5]

Question 3

(a) State the three stages, in order, of the machine cycle in classic Von Neumann architecture.

_____

_____

_____

_____[2]

(b) Two computer architectures are Reduced Instruction Set Computer (RISC) and Complex Instruction Set Computer (CISC) architectures.

(b)(i) Complete the table to show how the statements apply to these architectures.

	RISC only (✓)	CISC only (✓)	both RISC and CISC (✓)
Has many addressing modes			
Many instructions are available			
Uses one or more register sets			
Uses only simple instructions			

[4]

(b)(ii) Compare the number of machine cycles used by RISC and CISC to complete a single instruction.

_____

_____

_____

_____ [2]

(c) An array processor is used in some systems.

(c)(i) Explain the term array processor.

_____

_____

_____

_____[2]

(c)(ii) Give one example of the type of task for which an array processor is most suitable.

_____

_____[1]

Past Exam Question Answers

Question 1

(a)

- PC holds address of next instruction
- PC passes this address to MAR…
- MAR holds address of instruction/data
- Instruction/data from address in MAR is loaded to MDR
- PC is incremented (in each cycle)
- PC is changed when there is a jump instruction…
- …by taking address from instruction in CIR

(b)(i)

- uses (complex) instructions each of which may take
- multiple cycles
- single register set
- instructions have variable format
- many instructions are available
- many addressing modes are available

(b)(ii)

- programs run more slowly…
- …due to the more complicated instructions/circuit

Question 2

(a)(i)

- a location in the processor
- used for a particular purpose
- (temporarily) stores data/or control information
- explained example of contents held by named register

(a)(ii)

- program counter
- memory address register
- memory data register/memory buffer register
- current instruction register
- index register
- interrupt register
- accumulator

(b)

advantages:

- allows faster processing
- more than one instruction (of a program) is processed at the same time
- different processors can handle different tasks/parts of same job

disadvantages:

- operating system is more complex...
- …to ensure synchronisation
- program has to be written in a suitable format
- Program is more difficult to test/write/debug

Question 3

(a)

- fetch, decode, execute
- correct order

(b)(i)

	RISC only (✓)	CISC only (✓)	both RISC and CISC (✓)
Has many addressing modes		✓	
Many instructions are available		✓	
Uses one or more register sets			✓
Uses only simple instructions	✓		

(b)(ii)

- RISC: each task may take many cycles
- CISC: a task may be completed in a single cycle
- …as instructions may be more complex than individual instructions in RIS

(c)(i)

- a processor that allows the same instruction to operate
- simultaneously…
- …on multiple data locations
- the same calculation on different data is very fast
- Single Instruction Multiple Data (SIMD)

(c)(ii)
(accept any example of a mathematical problem involving large number of similar calculations)

- eg weather forecasting / airflow simulation around new aircraft

Further Resources

Resources on the VRS (http://vrs.virtutools.com)

- This hand out
- The presentation that goes with it
- More revision posters
- More past exam questions
- User contributed documents

Quizzes (http://revisionquizzes.com)

- Serial Processors
- Parallel Processors
- Co-processors
- CISC and RISC
- Computer Architectures summary

Class resources (email me for copy)

- Team quiz
- Videos
- Presentation
- Further notes on computer architectures

Information Sources

The following sources were used in writing the above:

- AS/A2 Computing textbook
- Harvard website
- Wikipedia
- The sites mentioned in the links section

None of the information in this hand-out, the presentation, the quizzes or any on the revision notes was copied directly from another source without being noted.

Links for further reading

How Von Neumann Architecture works
http://bugclub.org/beginners/history/VonNeumann.html
Video presentation on busses in Von-Neumann
http://www.youtube.com/watch?v=eWhnMNjxYDQ
Interesting article about John Von-Neumann
http://www.maa.org/devlin/devlin_12_03.html
Simple Von-Neumann Vs Harvard architecture
http://www.pictutorials.com/Harvard_vs_Von_Nuemann_Architecture.htm