2	**One** mark per mark point (**Max 4)** MP1 low number of instruction formats //low number of instruction sets MP2 uses single-clock cycle instructions MP3 uses fixed length instructions MP4 uses many general-purpose registers MP5 works well with pipelining MP6 hard-wired control unit MP7 makes extensive use of RAM MP8 uses a low number of addressing modes MP9 the design emphasis is on the software.

8	**One** mark per mark point (**Max 4)** MP1 many instruction formats possible MP2 large instruction set MP3 many addressing modes available MP4 uses variable length/multi-operation instructions MP5 multi-clock cycle instructions MP6 complex decoding of instructions MP7 uses complex circuits MP8 makes frequent use of cache memory MP9 uses programmable control unit // uses micro-programmed control unit // uses hardwired control unit MP10 hardware needs to be able to handle more complex instructions convert into sub-instructions // Design emphasis is on the hardware.

11(a)	**One** mark per mark point (**Max 2**) MP1 Uses **hard-wired code/control units** MP2 Uses relatively few instructions / simple instructions MP3 Uses relatively few addressing modes MP4 Makes use of a single-cycle for each instruction MP5 Makes use of fixed length / fixed format instructions MP6 Makes use of general-purpose registers MP7 Pipelining is **straightforward to apply** MP8 The design emphasis is on the **software** MP9 Processor chips require few transistors.
11(b)	**One** mark per mark point (**Max 3**) MP1 Once the processor detects an interrupt at the start/end of the fetch-execute cycle MP2 … the current program is temporarily stopped and the **status of each register** stored on the stack. MP3 After the interrupt has been serviced/the Interrupt Service Routine (ISR) has been executed … MP4 … the **registers** can be restored to its original status before the interrupt was detected // … the data can be restored **from the stack**.
11(c)	**One** mark per mark point (**Max 3**) MP1 Pipelining adds an additional complexity // there could be a number of instructions still in the pipeline when the interrupt is received MP2 All the instructions currently in operation are usually discarded except for the last one/the one at write back MP3 … the interrupt handler routine is applied to the remaining instruction. MP4 Once the interrupt has been serviced the processor can restart with the next instruction in the sequence.

9(a)	**One** mark per mark point for up to **two** benefits (**Max 2**) MP1 COMPATIBILTY e.g. Applications that aren't compatible with the host computer can be run on the virtual machine // It is possible to emulate old software on a new system by running a compatible guest operating system as a virtual machine // Software can be tried on different OS on the same hardware. MP2 PROTECTION e.g. The guest operating system has no effect on anything outside the virtual machine other virtual machines or the host computer//Virtual machines are useful for testing as they will not crash the host computer if something goes wrong // Easier to recover if software causes a system crash as virtual machine software protects the host system. MP3 COST e.g. No need to buy extra computers / hardware as multiple virtual machines can be implemented on the same hardware. **One** mark per mark point for up to **two** limitations (**Max 2**) MP4 PERFORMANCE e.g. The performance of the guest operating system will not be as good on a virtual machine as it would be on its own compatible machine because of the extra code / using more RAM/memory space // The performance of the VM is dependent on the capabilities of the host computer // Response times cannot be accurately measured using a virtual machine. MP5 COMPLEXITY e.g. Building an in-house virtual machine can be expensive, time consuming and complex to maintain / set-up. MP6 HARDWARE/SOFTWARE ISSUES e.g. Some hardware/software can't be emulated with a virtual machine // Some of the host machine's hardware can't be directly accessed by the virtual machine.
9(b)	**One** mark per mark point – host operating system (**Max 2**) MP1 The host operating system is the normal operating system for the host computer / machine. MP2 It has control of all the resources of the host computer / machine. // It can access the physical resources of the host computer / machine. MP3 It provides a user interface to operate the virtual machine software. MP4 It also runs the virtual machine software. **One** mark per mark point – guest operating system (**Max 2**) MP5 The guest operating system **runs** within the virtual machine. MP6 … it controls the virtual hardware/software during the emulation. // It accesses the actual hardware through the virtual machine and host operating system. MP7 It provides a virtual user interface for the emulated hardware/software. MP8 The guest operating system runs under the control of the host operating system.

5	**One** mark per mark point – SISD (**Max 2**) MP1 Single Instruction, Single Data (architecture). // Data is taken from a single source and a single instruction is performed on the data. MP2 Contains **one** processor, a control unit and a memory unit. MP3 …that executes instructions sequentially. **One** mark per mark point – MIMD (**Max 2**) MP4 Multiple Instruction, Multiple Data (architecture). // At any time, any processor can execute different instructions on different sets of data. MP5 Contains **many** processors MP6 …that operate **asynchronously / independently**.

10	**One** mark per mark point – SIMD (**Max 2**)
	MP1 Single Instruction, Multiple Data (architecture) // Performs the same operation on multiple different data streams simultaneously.
	MP2 The instructions can be performed sequentially, taking advantage of **pipelining**.
	MP3 Parallel computers **with** multiple processors.
	
	One mark per mark point – MISD (**Max 2**)
	MP4 Multiple Instruction, Single Data (architecture) // Performs different operations on the same data stream.
	MP5 Each processor works on the same data stream independently.
	MP6 Parallel computers **with** multiple processors.