**P7 and P8 unit 19**

**Task 7:** Describe how polling can be used to query whether a printer is ready to receive another page of data. Describe the alternative method where the printer raises an interrupt. Which of these two is the most efficient and why?

**Establishing communication**

A computer will need to communicate with the printer in order to send a printing job. So the computer will require to communicate with the control bus. So the control bus commands instruction to get the status of the printer then the printer might have busy flag or bit will say if the printer is prepared to obtain data from the buses. For example , the busy flag is like a counter representing whether the printer is being used; if it is being used it will show as ‘1’ but if the it’s a ‘0’, then the CPU can direct data.

**Polling method**

The CPU will know when to receive another page from the printer by using polling. So polling works by judging when to obtain a page from printer and so the polling method is frequently used for input/output. The CPU will check the status of the I/O device (printer’s busy flag) on a constant time until when the device is prepared. When the printer has finished printing a page, will then reset the busy flag to ‘0’. At this point the CPU in the computer system will then poll the printer to check for if it knows it can send another page.

CPU

CPU

CPU

CPU

PRINTER

Busy flag: 0

PRINTER

Busy flag: 1

PRINTER

PRINTER

**Explanation of the diagram**

1. First the CPU checks status of the printer.
2. Printer busy flag is 1 so CPU cannot send data.
3. The CPU once again checks the status of the printer.
4. Printer’s busy flag is 0 so the CPU can send data.

**Interrupting method**

There is another alternative method of receiving another page from the printer and the method is interrupt. So interrupt works by the printer will interrupt the CPU instead of the CPU polling the printer a regular interval, so interrupt method has to transfer data from the memory to the printers’ buffer. The buffer is memory which is connected to the printer and make the printer ready to receive another page. So the main purpose of the buffer is to help the communication between the CPU because it stores data that has been sent to the computer printer to print. However, it is unable to print because another print job is taking place. The print buffer permits the CPU to continue functioning smoothly even when operating something different while printing at the similar time. We see this when we send a page to print and there are extra pages being printed and so it prints the page in the order of how it was sent from the computer, when this happens the printer will have a busy flag that will be set to ‘1’ in which allows the CPU to operate something else.

**Which of these two is the most efficient and why?**

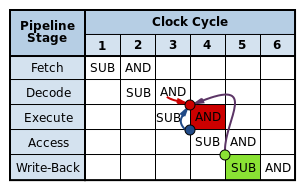
Both methods can function well but one method is better than the other. The best and efficient method is interrupt. The main reason for interrupt is better than polling is because it saves the CPU significant time, so it doesn’t have to keep going back and forth to check the busy flag. Another reason is that the CPU can do other job in the meantime and so it doesn’t waste time. Finally reason it is better than polling because it doesn’t have to face problems that might occur with the polling method since the CPU has to break and stop. This is because it has to make sure the printer has been poll; until it is finally prepared and this means the CPU won’t be capable to complete additional jobs correctly due to wasting time polling the printer.

**B. Explain how the processor uses pipelining to increase efficiency.**

**Pipelines:** An important factor in the way a CPU executes its operation is pipeline. So the main purpose of pipeline is essentially a queue into which CPU instructions are fed. So when an instructions are directed to the CPU, they are located into the pipeline in sequential order. The instructions within the pipeline are executed in the order that they are received.

**Pipeline (Efficiency)**

The processor uses pipeline to increase efficiency in the computer system. We know that a CPU holds numerous dissimilar parts. So the instruction can simply use one of the CPU's parts at a time. This means that all the other parts of the CPU remain sluggish as an instruction is executed and wastes of CPU resources. The solution is pipeline it permits instructions to be performed in a method that allows most of the CPU's mechanisms to work simultaneously and overlapping each instructions. So this means pipelines lets numerous instructions to be handled at a similar time. While one stage of an instruction is being processed, other instructions may be undergoing processing at a different stage. Without a pipeline, each instruction would have to wait for the previous one to finish before it could even be accessed. In the 21st century there are 4 processors that can have 20 stages of pipeline. So the superscalar processor can permit the arithmetic logic unit to handle 2 instructions in a parallel way and so this means it will be faster to process instructions.



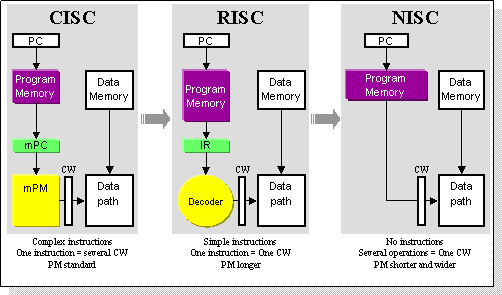
**How pipeline works (fetch execute cycle)?**

In order to understand how pipelining really works, you need to know about the fetch execute cycle. So the fetch execute cycle works by getting the first instruction which is called fetching. Fetching is the process of reading a part of a program from memory. Next process comes the decode sequence. Decoding involves examining the code that was saved throughout the fetch to see what type of process need to be performed, and what data is necessary to complete the process. After decode comes the execute stage of the instruction. This is where all mathematical operations are performed by the CPU's arithmetic logical unit. The memory access stage of the instruction is not continuously performed. It is only necessary if the CPU needs to read extra data from memory or write data to memory. For example if only four instructions have been placed into a CPU's pipeline. The CPU begins working on those instructions by performing the fetch stage of the first instruction. Once the fetch is complete, the CPU can move on to the decode stage of the first instruction.

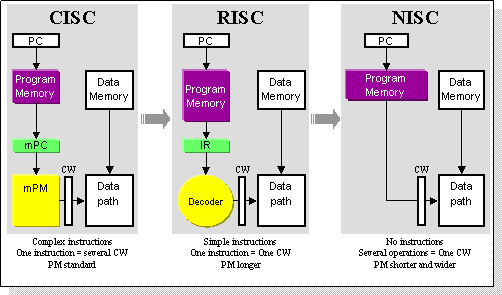
Also remember the CPU which handles the fetch instruction is freed up. Therefore, the CPU can begin working on the fetch stage of the second instruction at the same time, also it is working on the decode stage of the first instruction. When the CPU is ready to perform the execute stage of the first instruction, the fetch stage of the second instruction is done. Therefore, the CPU can begin working on the decode stage of the second instruction and the fetch stage of the third instruction.

**C. Compare Reduced Instruction Set Computer (RISC) chips and Complex Instruction Set Computer (CISC) chips**

**CISC:** stands for Complex Instruction Set Computerand was the first architecture used in a processor. So the main purpose of CISC is to try and handle task by using less memory slots in the main memory. So the architecture works by a single instruction can transmit out numerous operations for example load form memory, add, and store in memory. On the other hand the number of instruction had to been reduced by having multiple operations within a single instruction. In the past the purpose was to try and use as less memory in the processor because memory was very expensive. For example a 10MB was a big amount of memory in the past but in the 21st century 10MB is not sufficient memory in the computer system to hold bigger program, files and data.



**RISC:** stands forReduced Instruction Set Computer. The second architecture was intended to solve the problems of the CISC. This architecture works completely different from the CISC, instead of having more than one instruction within one memory slot in the main memory. So the main purpose of RISC is to provide minimal set of instruction that could be carrying out all important operations. So for example if you had three instructions which were loaded from the memory, add and stored in the memory then each of these instructions would go within one memory slot. This is because instructions where complex had to be reduced, so the solution was having few simple instructions that are the same length and permit memory access only with clear load and store instructions also this means each instruction performs less work.



**Comparison**: The main difference between RISC and CISC is the amount of computing cycles each of their instructions takes. For example CISC, each instruction might use large amount of numbers to process each cycles before completing the task. This is because the difference is the amount of number of cycles depends on the complexity and aim of their instructions. On the other hand in RISC, each instruction is only designed to complete a very small task. So if you want a difficult task done, then you require a lot of these instructions together. With CISC, each instruction is similar to a high level language code, so simply you require less instruction to achieve what you want as each instruction does a lot.

Another difference between RISC and CISC is the amount of available instructions. RISC uses little amount of stages might be required to divide instruction, dissimilar in CISC where a single instruction would be prepared to have multiple stages. Even though CISC might be easier for code programmers, it also has disadvantages because when using CISC might not be as well-organised and efficient then the RISC architecture. This is because inefficiencies in the CISC code will be used more than once, this leads to wasting cycles and resources. The advantages of RISC architecture it permits the programmer to take away pointless code and avoiding wasting cycles.

The disadvantage of CISC architecture it is not popular in the 21st century. This is because a memory tasks in one memory slot of the main memory (RAM) causes the computer to perform very slow and take a while to carry out the task. On the other hand RISC architecture has many advantages such as the computer will execute and perform instruction much faster compared to the CISC architecture, and also in the 21st century you can get a lot of memory for less money compared in the past.

Another difference between CISC and RISC how these two architecture are being used in the modern world. So the CISC has managed to gain early advantage in computing with the domination of Intel’s x86 architecture, which is the foundation for all previous modern computer architecture. In comparison the, RISC has managed to become one of the popular architecture within portable devices for example smart phones, tablets, GPS receivers, and other related devices. Finally to conclude I think in modern computing RISC architecture is better than CISC because of Acorn RISC Machines which is one of the popular and outstanding devices used in RISC also the higher efficiency of the RISC architecture which it provides for users.

**References**

http://www.differencebetween.net/technology/protocols-formats/difference-between-risc-and-cisc/

<http://kipirvine.com/asm/4th/sampleChapters/chapt_02.pdf>

http://www.techrepublic.com/article/solutionbase-how-pipelining-and-multiple-cores-help-speed-cpus/#.