Johnathon Jack Moore Task 3 Questions 2/21/19

* Identifying the components on the raspberry PI B+

The most important component of the Pi is the CPU (which also happens to be the RAM in this case) which is the largest square chip located on the left half of the board underneath the heat sink. Along the sides there are connectors for USB (4 ports) HDMI, power, camera and display. There is also an Ethernet port next to the USB’s that has a corresponding controller nearby.

* How many cores does the Raspberry Pi B+ CPU have?

As the CPU is labeled as a Quad-Core CPU, it is safe to assume that there are four cores.

* List three main differences between X86 (CISC) and ARM Raspberry PI (RISC), Justify your answers and use your own words (do not copy and paste).
  + ARM’s RISC instruction set is far more basic and can only access memory to Load/Store. This means that ARM cannot use instructions other than Load/Store on memory directly leading to moving memory to registers and back to memory for simple changes which can feel redundant to someone writing the code.
  + ARM has the potential to be faster and more efficient, while writing X86 is much simpler due to the removal of some redundancy found in ARM’s simple instructions.
  + X86 has less total registers due to the trade off for complex instructions in dealing with memory, which may affect the scope of some larger codes.
* What is the difference between sequential and parallel computation and identify the practical significance of each?

In sequential computation, the code is broken down into parts and is executed one at a time through a single processor. Parallel computation sends its parts of code and solves them simultaneously on different processors. Parallel is important as it can execute code more efficiently than sequential, while Sequentials significance lies with ease of access especially when coding for hardware that may only have one core.

* Identify the basic form of data and task parallelism in computational problems.
  + Data Parallelism refers to problems that require the same computation to be used on multiple data items. Imagine how many computations will need to be done if a large group of data elements had to all be individually modified, The potential parallelism could be huge.
  + Task Parallelism refers to problems where the focus of the parallelism is on the functions and not the data itself.
* Explain the differences between processes and threads?

Threads are used for “lighter” tasks that allows a process to be broken up into independant parts, as well as the fact that they share the common memory of the process they are a part of, while processes are abstractions of a program that run with separate memory and are used primarily for heavier tasks. Processes can only be done one at a time with a single-core CPU

* What is OpenMP and what is OpenMP pragmas?

OpenMP is an API which is used for parallel programming within a multi-core CPU. The OpenMP pragmas are compiler directives that will cause the compiler to generate threaded code in order to carry the work out using parallelism.

* What applications benefit from multi-core (list four)?
  + Compilers
  + Database Servers
  + Multimedia Applications
  + Web Servers
* Why Multicore? (why not single core, list four)
  + It is used to bypass clock frequency limits of single-core
  + Several newer applications are multithreaded which essentially requires multi-cores.
  + General computer architecture trends are shifting toward parallelism.
  + Deeply pipelined circuits (Problems with more sub-problems) would be unnecessarily complicated with single-core