Task 3: Parallel Programming Skill

a)

**- Identifying the components on the raspberry PI B+:**

+ 2 USB port

+ Ethernet

+ 2 Power port

+ Camera

+ HDMI

+ Ethernet Controller

+ CPU/RAM

+ display

**- How many cores does the Raspberry Pi’s B+ CPU has:**

4 cores

**- List three main differences between X86 (CISC) and ARM Raspberry PI (RISC).Justify you answer and use your own words:**

+ Main differences: instruction set

+ Intel (CISC):

. has a larger, more feature-rich instruction set , allows many complex instruction to access memory => more operations, addressing modes but less registers than RISC

. used in normal PCS, workstation, servers

. little-endian format

+ ARM (RISC)

. simplified instruction set, more general purpose register than CISC.

. use instruction only operates on register

. load/store memory model for memory access

. adv: instruction executed more quickly, greater spd

. disadv: less instruction. great (emphasis) depends on efficient writing

. has 2 modes: ARM and Thumb

. was little-endian b4 3. After that become BI- endian

. feature setting allows switchable endianness

.conditional execution

**- What is the difference between sequential and parallel computation and identify the practical significance of each?**

+ Serial Computing: (Sequential computing)

. Traditionally, software has been written for serial computation

. A problem is broken into a discrete series of instructions

. Instructions are executed sequentially one after another

. Executed on a single processor

. Only one instruction may execute at any moment in time

. Practical Significance: easy instruction, old applications

+ Parallel Computing:

.In the simplest sense, parallel computing is the simultaneous use of multiple compute resources to solve a computational problem

. A problem is broken into discrete parts that can be solved concurrently

. Each part is further broken down to a series of instructions

. Instructions from each part execute simultaneously on different processors

. An overall control/coordination mechanism is employed

Practical Significance: fast execution, new applications

**- identify the basic form of data and task parallelism in computational problem:**

+ Data parallelism:

. refers to a broad category of parallelism, in which same computation applied to multiple data items

. amount of available parallelism proportional to input size => tremendous amount of potential parallelism

. give programmers flexibility to write scalable parallel program.

+ Task parallelism:

. applies to solutions where parallelism is organized around the function to be perform rather than the data

. emphasis is on the functional decomposition, can be implemented with either task or threads

. challenges are to balance the work and to make sure all work contributes to the result

. task parallelism does not scale well as data parallelism in most cases

**- Differences btw processes and thread:**

+ process: abstraction of a running program

. process does not share memory with each other

+ threads: a lightweight process that allows a single executable/process to be decomposed to smaller , independent part

.all threads share a common memory of the process they belong to.

**- OpenMP and OpenMP pragmas:**

+ OpenMP: is a library/language for programming multicore architecture; uses implicit

multithreading model in which the library handles thread creation and management, thus making the programmer’s task much simpler and less error-prone; is a standard compilers who implement it must adhere to .

+ OpenMP pragmas: compilers directive that enable the compiler to generate threaded code.

**- What applications benefit from multi-core (list four)?**

+ Database servers

+ Web servers (download software while running anti virus program)

+ Compiler

+ Multi media application ( editing a photo while recording a tv show in a recorder)

+ Scientific App ( CAD/CAM)

+ in general, applications with thread-lvl parallelism

**- Why multicore?**

+ single core only operates one process at a time. ( process being the abstraction of a running program.)

=> multi cores = multi process operate at once (concurrency)

+ increase the throughput of the system

+ difficult to make single core clock frequencies to be higher

+ Deeply Pipeline circuits:

+ many new application are multithreaded

+ General trend for comp architecture: more parallelism