Q1

Illustrate the share and distributed memory architectures to run a data parallel task. Explain how you search NADRA's large database for male citizens having age > 85 utilizing data parallelism on the distributed memory architecture.

See share and distributed memory architectures as per lecture slides. In data parallelism, data is divided among different nodes and same program (Single Program Multiple Data SPMD) will be executed on each node to query male citizens having age > 85 years on data chunk on the node to generate partial results. A master (thread or process) will distribute data and aggregate partial results.

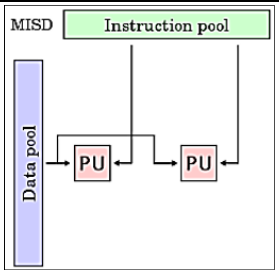
Consider a processor operating at a clock of 0.5 nsec. It is connected to a DRAM with a latency of 80 ns (no caches). Assume that the processor has four multiply-add units and is capable of executing four instructions in each cycle. Show working to calculate peak processor rating in GFLOPs. Why performance of a dot-product computation of two vectors reduces the peak rating? Show sample calculations.

- Clock time period is 0.5 nano seconds correspond to 2 GHz frequency (t=1/f). Since four instruction are executed in each cycle therefore it is capable of executing 8 GFLOPS.
- Peak rating will be reduce due to memory access as the processor have to wait 80ns to read an operand before executing the operation each instructions which access the memory.
- Dot-product multiply-add operation needs two memory load to read vector elements resulting in 100+100+1= 201 nsec. Therefore, operation involving memory operand runs at the speed of 2/201 ~ 9.9 MFLOPS

Q2

a) Define MISD Flynn's taxonomy and one application, which is, can be applied.

See diagram for the definition. Assume an encrypted input data stream, The MISD architecture applies different decryption algorithm (multiple instructions) in parallel to the single input data stream to speed-up code breaking.



b) Assume a parallel application running on CS Lab # 1 PCs cluster. Why you consider coarse-grain granularity in this parallel code? Explain.

CS Lab # 1 PCs has high computational capability and large (> 8GB) memory. Therefore, if the parallel solution run tasks that performs large computation before sharing partial results to other tasks than coarse-grain granularity seems a good fit. In case the parallel solution has small chunk of computations and hence large amount of partial results from many nodes need to be shared fine-grain granularity is also possible without any degradation of performance as we have a high-speed/high-bandwidth network between Lab # 1 PCs.

Show OpenMP C code snippet where 8 threads work together to computer sum of an 800K array of integers. Use any one thread to initialize the array. All threads will takes part in the summing computations. Sum in a global variable in the main () functions. Use OpenMP constructs used in Lab # 1 and Lab # 2 only.

```
#pragma omp parallel num_threads(8)
{
        If (omp_get_num_threads() == 0) { // only thread0 will run this code
           init_array(array, 800000);
        }
        #pragma omp barrier // all threads wait here. Only thread 0 init array

        int local_sum = 0; // thread local variable

        // sum array using team of threads using workload distribution.
        // Use workload distribution or divide loop iterations manually
        #pragma omp for
        for (int i = 0; i < 800000; i++) {
            local_sum += array[i];
        }
        // Use a critical section to update the global sum
        #pragma omp critical
        global_sum += local_sum;
}
```

Please note that the comments explain the context and use of openMP usages and side effects.