IIT JODHPUR

Minor-1 Examination (OPEN-BOOK): EEL7210 Hardware Software Co-Design

Guidelines (Total time: 60 minutes, Maximum Marks: 15):

- Please read the question paper very carefully (both sides of this paper). **NO clarification is required in any question.** In case of any doubt, assume whatever you wish to and state that.
- This is an OPEN-BOOK examination. However, usage of internet/any AI tool is **NOT** allowed.
- Please adopt mathematical reasoning as far as possible in your answers.

---

1. Consider that the software part of an application A is executed on a processor P and the hardware portion is executed on dedicated logic blocks LB. Further, because of difference in the speed of P and LB, there is a requirement of some buffers in the software-hardware interface between the processor and the dedicated logic blocks to adjust for this speed mismatch. Suppose the time-period of reading data from P by the interface is t_r and writing back the data to LB is t_w. There is a delay of 1 time-unit between two successive writes by the interface and there is a delay of 3 time-units between two successive reads by the interface. If N is the length of maximum data segment that is read by the software-hardware interface, find the number (n) of buffers (each of size one) to be supported in this software-hardware interface? What is the value of n if $1/t_r$, $1/t_w$ and N are known as 50 MHz, 30 MHz and 120 respectively? [2.5 + 0.5 Marks]

2. Suppose the goal of a design is to perform some encryption-related task. One of the step in this process is to do multiplication with prime numbers. However, the prime number identification has to be done in an on-chip manner i.e., it has to be computed in the hardware. Therefore, you can assume that a random stream of numbers is input to your system and your design should perform the detection of prime numbers from this stream. Develop the complete hardware design (datapath + controller) description for performing the prime number identification following a control flow graph (CFG) and data flow graph (DFG) computation approach? [3 + 2 Marks]

3. Let's say we have 3 matrices, A, B and C. We want to compute the accumulated sum, C = AB + C. In other words, we wish to design a system that can calculate C[n][m] = A[n][k]*B[k][m] + C[n][m]. Recall that this multiplication can be easily done in three nested loops over the variables n, m and k respectively. In order to optimize, we have decided to implement this multiplication in the following manner. Check if the below implementation is better than the conventional nested

```
for (int n = 0; n < N; ++n)
    for (int m = 0; m < M/T; ++m) {
        double acc[T]; // Tiles of size T
        for (int k = 0; k < K; ++k)
            double a = A[n][k];    // M/T reads
            for (int t = 0; t < T; ++t) {
                double prev = (k == 0) ? C[n][m*T+t] : acc[t];
                acc[t] = prev + a * B[k][m*T+t]; }
        for (int t = 0; t < T; ++t) // Write
            C[n][m*T+t] = acc[t]; } // out
```

loop implementation for computing C[n][m] = A[n][k]*B[k][m] + C[n][m]? What parameters are getting optimized here: throughput, latency, parallelization etc.? [3 marks]

4. The performance of a CPU is given by the MIPS (Million Instructions Per Second) metric, however, FLOPS (Floating Point Instructions Per Second) is better for performance measurement as it tries to capture the CPU's ability to handle floating-point operations. Consider the design of a heterogeneous digital system where we are asked to follow a hardware-software co-design approach for executing a deep learning (DL) application. A DL system needs to solve a lot of floating-point operations for tasks such as matrix multiplications/convolutions etc. For the software execution part, we have 3 options to choose from: P_1 is a pipeline processor with a performance of 8 MIPS per MHz, P_2 is a non-pipeline processor with a performance of 3 MIPS per MHz and P_3 is an out-of-order processor with a a performance of 1 GFLOPS. Simplistically, we can assume that each floating-point operation translates to 6 instructions on an average. Additionally, it is given to us that if we use P number of P_3 type processor, we are allowed to use at the maximum $2P$ number of P_1 type processor and $4P$ number of P_2 type processor. The performance numbers of P_1, P_2 and P_3 have been obtained when they are executed at a clock frequency of 600 MHz. P_1 has an area requirement of 4000 units, P_2 has an area requirement of 2000 units and P_3 has an area requirement of 8000 units. For the hardware part, we have two variants of floating-point units, called as FPU_1 and FPU_2 that take an area of 2600 and 3200 units respectively. FPU_1 is 1.5 times faster than that of the single instance of slowest processor for floating-point operations and FPU_2 is 2.5 times faster than that of the single instance of the slowest processor for floating-point operations. Devise a hardware-software partitioned solution (i.e., number of FPU_1, FPU_2 and the processor choices) if an application consists of a total of 6000 Giga floating point operations so that this application can be completed as quickly as possible and the total system fits within an area budget of 11200 to 14800 units? Find the total time and area taken by your hardware-software partitioned solution to complete this DL application execution? [3 + 1 Marks]
