Part 1 (4 *points*)

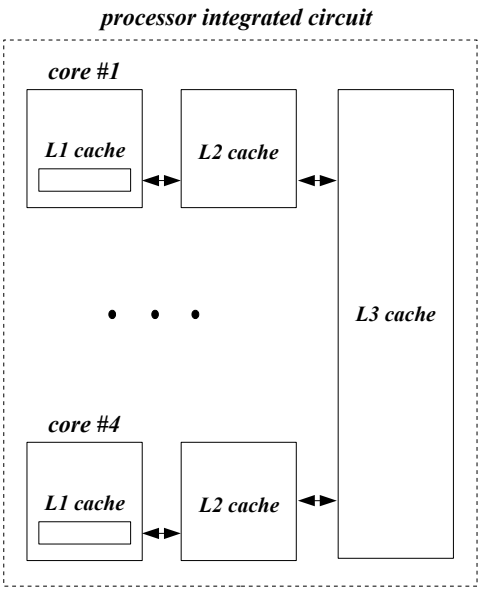
1. Take the following memory addresses expressed in decimal: 3674_{10} and 5932_{10}. Convert them to hexadecimal assuming a 32-bit address length (all computations must be carried out on the exam paper). Which one can not represent the location of a properly aligned 32-bit operand? Justify your claims in detail. (2 points)
2. Write the general equation that solves the following problem: What is the speed up that can be achieved when an application is run in a N processor computer system, where the fraction P of the application execution time in a single processor is concurrent and may be run in parallel. What is the name of the law the equation portrays? Why does one say it is a law of *diminished returns*? (2 points)

Part 2 (4 *points*)

3. Explain in detail how a K stage *pipelined implementation* of a given processor can increase the throughput of instruction execution when compared to a *non-pipelined implementation* of the same processor. However, for the implementation to work properly, care should be taken to solve all the hazards that may arise. What are those hazards and how are they solved? (2 points)
4. Sketch the *5-stage classical processor pipeline* with an integer execution unit for addition and subtraction and for the logical and shift operations, an integer multiplication unit and an integer division unit. Explain why the integer multiplication unit is usually pipelined, but the division unit is not. Explain also why in such an organization instructions can not be executed *out of order*. (2 points)

Part 3 (4 *points*)

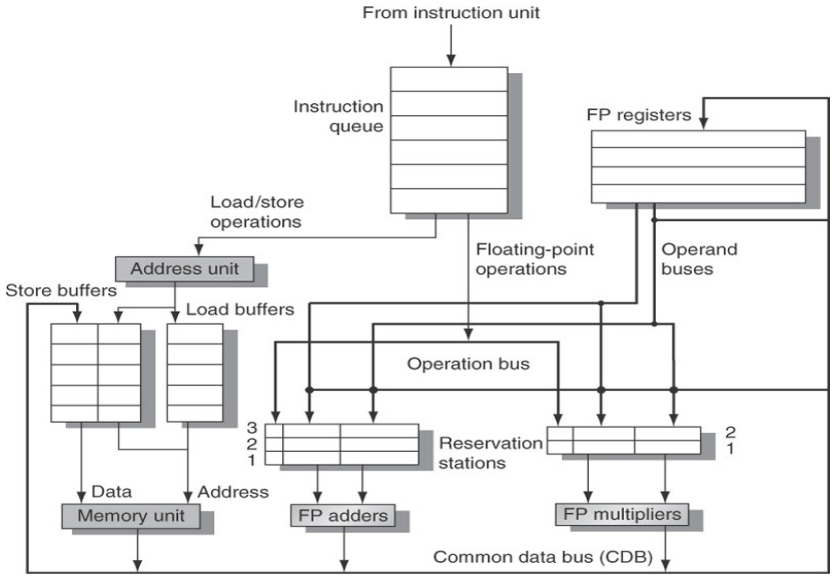
5. In order to speed up memory access, a high speed special memory, called *cache*, is placed between the processor and main memory. However, since main memory capacity is much larger than cache size, many memory blocks will overlap at the same cache location during program execution. How may the cache be organized to deal with this problem? Describe them in detail. (2 points)
6. The diagram below depicts the cache hierarchy for a multicore processor. (2 points)



i. Why three cache levels are typically used?
ii. Why level 1 is usually divided in an instruction and a data cache?
iii. What kind of *write policy* is usually applied to them?

Part 4 (4 *points*)

7. Sketch in detail the organization of a (8,2) correlating branch predictor which also takes into consideration the lower 4 bits of the branch instruction address and explain how it works? Assume a 32-bit instruction address. What is the size in bits of the branch prediction buffer? Justify your claims in detail. (2 points)
8. The diagram below depicts the basic organization of a floating point unit using the Tomasulo's algorithm.



Explain in detail how the different types of data hazards are solved in this organization. (2 points)

Part 5 (4 *points*)

9. Explain why a graphics processing unit (GPU) can be considered to be a MIMD computer based on SIMD processors. Supplement your text with a schematics that turns your explanation more clear. (2 points)
10. Assume the following CUDA C computation kernel that is run in a 16 X 2 grid of 8 X 16 blocks of threads as the launching configuration.

```
__global__ static void kernel (float *xx, float *yy, float *zz, int N)
{ int x, y, idx;
  x = threadIdx.x + blockDim.x * blockIdx.x;
  y = threadIdx.y + blockDim.y * blockIdx.y;
  idx = blockDim.y * gridDim.y * x + y;
  zz[idx] = xx[idx] * yy[idx % N];
}
```

How many warps are in each block of threads? Which is the separation of the elements of the array `zz` that are computed in the same warp? Justify your claims in detail. (2 points)