## *Part 1* (4 *points*)

1. Take the following memory addresses expressed in decimal: $6288_{10}$ and $5430_{10}$. Convert them to hexadecimal assuming a 32-bit address length. Which one of them can not represent the location of a properly aligned 64-bit operand? Justify your claims in detail. (2 points)
2. When discussing computer architecture, one often speaks about *structure* and *function*. Distinguish these two concepts and explain how a *computer system* can be seen in relation to each of them. (2 points)

## *Part 2* (4 *points*)

3. Consider the following code sequence for the pipelined version of MIPS 64 processor described in the lectures, where the equality test on the contents of the registers for the branch instruction is computed in the instruction decode (ID) stage

```
        L.D     F2,0(R4)
LOOP:   L.D     F0,0(R1)
        ADD.D   F4,F0,F2
        DADDUI  R1,R1,-8
        S.D     F4,0(R1)
        BNE     R1,R2,LOOP
        MUL.D   F6,F4,F2
```

   Assuming that neither the forwarding, nor the interlocking unit are present, intersperse NOP instructions so that the code can run correctly. What would be the changes on the code that would arise if both units were present? Justify all your claims in detail. (2 points)

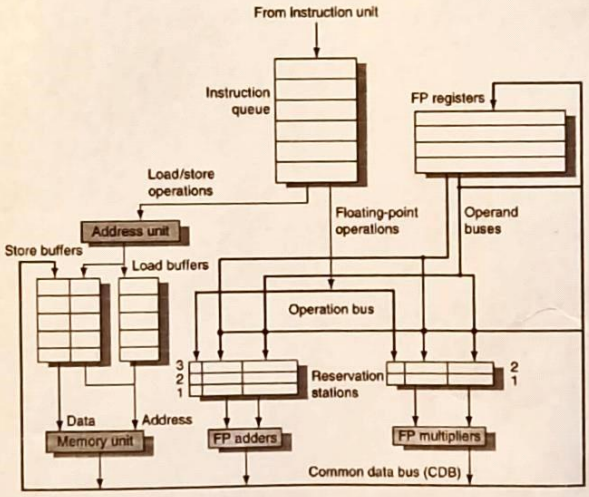
4. How does the branch delay slot feature work? Describe in detail what happens in both situations: when a branch is taken and when a branch is not taken. Although this feature does improve pipeline throughput, it is not an optimal solution. Why it is so? (2 points)

## *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 the different solutions in detail. (2 points)
6. A major hurdle on cache efficiency are *cache misses*. (2 points)
   i. What are they?
   ii. Describe the main types they may be divided in?
   iii. Which of those types are affected by the size of the memory block stored in each cache line? Justify your claims in detail.

## *Part 4* (4 *points*)

7. What are *name dependencies*? Two types must be considered. Which are they? Present an example of each. Why can they be solved by *register renaming*? (2 points)
8. The diagram bellow depicts the basic organization of a floating point unit using the Tomasulo's algorithm.




Explain in detail how the different types data hazards are dealt with in this organization and why the Tomasulo's algorithm is considered a dynamic scheduling technique. Justify your claims in detail. (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 32 X 8 grid of 16 X 32 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 there in total? Which is the separation of the elements of the array `zz` that are computed in the same warp? Justify all your claims in detail. (2 points)