COM1006/COM1090 Devices and Networks (Autumn)
Tutorial Sheet #10: Accelerating Performance

1. Consider the following Motorola 68K instructions:

	D0	D1	D2	D3
`CLR.L  D0`				
`MOVE.L #$0000CAFE,D1`				
`ADD.B  D1,D0`				
`MOVE.L D1,D2`				
`CLR.L  D3`				
`AND    #$000F,D1`				

   a) Fill in the hexadecimal values of registers D0-D3 after each instruction in the above table. **Do this by hand.** Leave cells empty if a register has not changed. Assume that initial values are unknown.

   b) Now implement these instructions in EASy68K and check your answers.

2. Consider the following Motorola 68K instruction sequences being executed in a four-stage pipeline with stages: instruction fetch (IF), operand fetch (OF), execute (E), and operand store (OS). Assume internal forwarding is **not** being used.
   Determine for each instruction sequence, whether a data dependency occurs, causing a pipeline stall (bubble). If so, name the register involved.

   Hint: for longer sequences it might help to draw a timing diagram with phases IF, OF, E, OS, like the one on Lecture 12, Slide 11, to see what phase is executed at what time.

```
MOVE   D1,D2          MOVE   D1,D2
MOVE   D1,D3          MOVE   D2,D3


MOVE   D1,D2          MOVE   D1,D2
MOVE   D3,D2          ADD    D3,D2


MOVE   D1,D2          MOVE   D1,D2
AND    D3,D4          AND    D3,D4
SUB    D2,D3          CLR    D1
CLR    D1             SUB    D2,D3
```

3. Explain the problem arising when the pipeline from 3. executes the following instruction sequences. How is this problem called?

```
          SUB   D1,D2                CMP   D1,D2
          JMP   Loop                BNE   Loop
          CLR   D3                  CLR   D3
          MOVE  (A1),D4             MOVE  (A1),D4
          ...                       ...
          ...                       ...
          ...                       ...
Loop:     MULU  D6,D7         Loop: MULU  D6,D7
```

How can this problem be solved?
Which of the two sequences is easier to deal with, and why?

4. A computer has a small main memory of 32 words organised into 4 sets of 4 lines, and a direct-mapped cache. The 5-bit memory address from the CPU has the form $S_1S_0L_1L_0W$, where bits S_1 and S_0 index the set, bits L_1 and L_0 index the line, and bit W indexes the word.

   For this computer, explain what happens when the following memory addresses are accessed in sequence. Assume that the cache is initially empty.

   00110
   10110
   10111
   01010

   Show how the tag fields in cache memory are updated, and state whether an address led to a **hit** or a **miss** in regard to cache memory.