ELEC 3275 - Computer Architecture
Assignment 11 - All Together now

This can be done with partners (4 people per group maximum).

Provided below are two tables. One (on the left) is a program, saved in machine code in the instruction memory. The other (on the right) is the memory pointed to by this program (and thus NEEDED by this program). For the sake of space, only 32 bits are shown in the data (assume the rest are 0).

0x0100	10010001000000000001001111100000
0x0104	10010001000101000000001111100001
0x0108	11111000010000000000000000100010
0x010c	11111000010000010000000000100011
0x0110	10001011000000011000000001000100
0x0114	11111000000000000000000000100100
0x0118	10010001000000000001000000100001
0x011c	11010001000000000000010000000000
0x0120	10110101111111111111111101000000
0x0124	11001011000001010000000010100101

0x0500	00000001000000010000000100000001
0x0508	00000010000000100000001000000010
0x0510	00000011000000110000001100000011
0x0518	00000100000001000000010000000100
0x0520	00000101000001010000010100000101
0x0528	00000110000001100000011000000110
0x0530	00000111000001110000011100000111
0x0538	00001000000010000000100000001000
0x0540	00001001000010010000100100001001

The parameters of the microprocessor pipeline and its datapath are listed below.

- Assume registers and data caches are all blank
- PC is initialized to 0x0100
- Assume instruction cache has the instructions prefetched (loaded) before you start.
- 5-stage pipeline with pipeline registers
- A memory access will take 1 cycle if the line you are loading/storing is a hit
- A memory access will take 10 cycles if the line is a miss
- The cache
  - can hold 2 words per line/way
  - is 2-way set associative
  - has 8 TOTAL entries/ways/lines
  - Uses an LRU replacement policy
- The microprocessor does have forwarding

Your task is to determine the latency of the program (e.g. how many cycles does it take to complete). To complete that, you will need to do a number of subtasks:

- Convert from machine code to ARM code, so it is readable
- Determine how the instructions flow through the pipeline, accounting for all stalls caused by hazards (data or structural) and cache accesses (could be a hit or a miss).
- Determine all cache accesses and the state of the cache at any given time
- Calculate the total number of cycles to get from the first instruction issues to the last one written back

Answer each of the following questions and SHOW YOUR WORK:

1. What is the ARM code?
2. What are the cache hits/misses?
3. Where are all the hazards in the code execution?
4. How many cycles does it take to run the code from beginning to end including all iterations)?
    1. I STRONGLY suggest using a spreadsheet for this.
5. What percentage of the time is spent waiting on the memory?

If you have questions, let me know as soon as possible. If you need to make an assumption in the work, WRITE IT DOWN.

You must also submit an evaluation of your team members. This is to be done individually and privately.

What to submit: The answers to all the questions as well as ALL work/proof of completion. If they are not digital (documents, spreadsheets, etc - which I highly recommend), you must make legible scans. If I can't read it, I can't grade it. Compile all files into 1-2 documents. Do NOT submit many picture files. Use doc, pdf, xls or their equivalent. The team member evaluation should be submitted in the Blackboard comments or in a document that you upload. Do not email them.