CE1921 HOMEWORK HW6	SPRING 2019	NAME: Julian Singkham

- This homework is **due on printed paper at the start of the second lecture in Week 7.**
- Your solution for each problem must show your work to receive partial credit. Failure to show your thought process will not earn points for incorrect answers.

1. Two instructions have been stored in instruction memory using big-endian format. These binary numbers were created by the assembler from assembly language statements typed by the programmer. **Determine** the assembly language statements typed by the programmer by disassembling the binary numbers back into their text form.

ADDRESS	HEX BYTE	
0	E0	MUL R9, R4, R5
1	09	
2	05	
3	94	
4	E5	
5	92	
6	C0	LDR R12,[R2,#0]
7	00	

2. Implement the following Java code in ARM assembly language. You can use the ADD and the SUB instruction only. You are not allowed to use multiply. **Hint:** the second operand of ADD can be rapidly multiplied by powers-of-2 by shifting it using the barrel shifter. How many places do you shift a number to multiply by 64? (See problem 4 in the previous homework assignment for syntax reference.) You must use the assigned registers noted in the comments. Your solution must be well-commented ARM assembly code and the Keil uVision4 simulation showing the final values after executing to the infinite loop. Write hand-written comments on the simulation that describe how you know that R9 contains the correct answer.

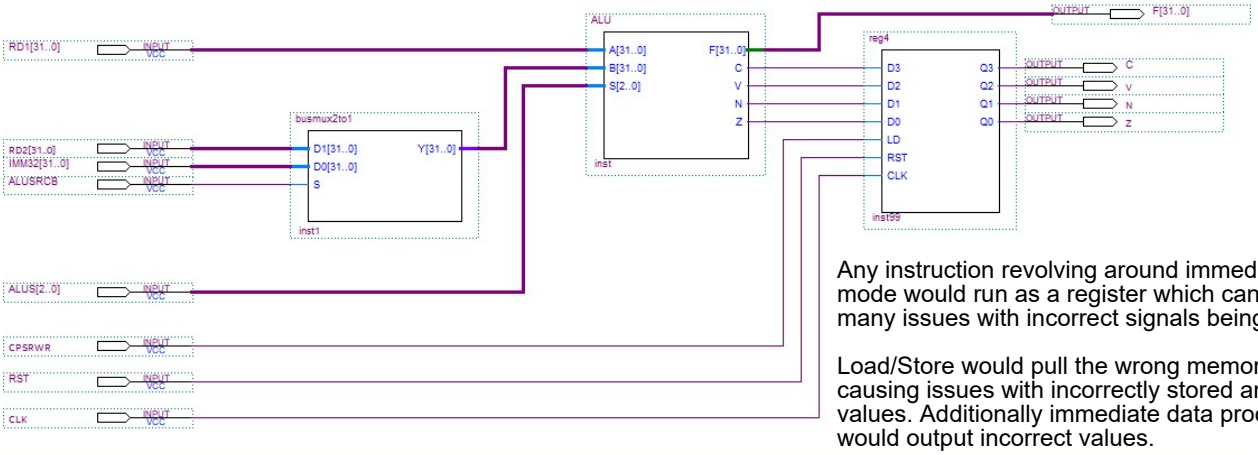
```
// implement only the algorithm inside main
// the class enclosure is shown just to provide completeness
// x is stored in ARM saved value register R9
// iterator j is stored in ARM temp register R12
class Main {

  public static void main(String[] args) {

    int x = 0;
    int j = 10;
    do
    {
      x = x + 64*j;
      j = j - 1;
    } while(j>0);

   while(true); // infinite loop
  }
}
```

3. Circuit reliability is a complex function of requirements analysis, design, fabrication, and unexpected events. Fabrication faults can lead to signals that do not change value as expected. Binary signals can be **stuck-at-logic-0** or **stuck-at-logic-1** because of fabrication faults. Consider an ARM CPU that has control signal ALUSRCB stuck-at-logic-1 in the execute stage circuit. **Describe** the impact on the data processing instructions in the instruction set. In other words, what instructions from your reference card would or would not work?



Any instruction revolving around immediate mode would run as a register which can cause many issues with incorrect signals being sent.

Load/Store would pull the wrong memory address causing issues with incorrectly stored and retrieved values. Additionally immediate data processing would output incorrect values.

4. **Calculate** the total storage requirement in bytes if 32,000 ARMv4 instructions are assembled. **Use** the modern units kibibyte (KiB), Mebibyte (MiB), Gibibyte (GiB), etc. **Show** all work to receive partial credit.

(32,000*4)/1024 = 125 Kibibytes