CMSC611: Advanced Computer Architecture
Extra Credit Homework 1

iAPX 86,88 (Intel Advanced Processor Architecture 8086/8088) is a very famous architecture dating back to 1980s, and the predecessor to the modern Intel CPU architectures. It is a two operand machine (source and destination). It supports source/destination operand combinations of register/memory, memory/register, memory/memory, immediate/register, and immediate/memory. Consider the following code segment and instruction set reference table for the iAPX 86,88. Assume the initial value for ARRAY[100] is 128 and for ARRAY[200] is 2048.

```
                MOV     AX, ARRAY[100]
                ADD     AX, 128
                MOV     CX, 4
                MUL     CX
                MOV     ARRAY[100], AX
AGAIN:          MOV     AX, ARRAY[200]
                SUB     AX, 256
                MOV     ARRAY[200], AX
                MOV     CX, AX
                MOV     AX, ARRAY[100]
                SUB     CX, AX
                JCXZ    AGAIN
```

Instruction	Operands	Clock Cycles
MOV destination, source	register, register	2
	register, immediate	4
	register, memory	12
	memory, register	13
ALU destination, source (for ADD and SUB in this case)	register, register	3
	register, immediate	4
	register, memory	13
	memory, register	24
	memory, immediate	25
MUL source (destination register is AX)	register	118
JCXZ label (jump to label if CX =0)	label	18

1. Compute the weighted average CPI for the code execution.
2. Operands in ALU can access local variables from memory by directly computing the effective address. We could use this to reduce the number of instructions in the code. For example, three instructions

```
MOV     AX, ARRAY[SI]
ADD     AX, 4
MOV     ARRAY[SI], AX
```

   can be combined into one instruction

```
ADD     ARRAY[SI], 4
```

   Show your new code (without changing the original code result) by reducing the number of original instructions and compute the new weighted average CPI. Could your new code improve the performance of the original one on the execution time?