Comments on HW 1

2f2. Write two instructions with the same STAB opcode to store the contents of accumulator B to memory location $0023. (Use two different addressing modes.)

Technically, the opcode is the hexadecimal machine instruction, hence it is not possible to write two instructions using the same opcode and two different addressing modes. The answers that the authors wanted were STAB $0023 (Extended Direct Mode) and STAB $23 (Direct Mode). However, the opcode for the first is 7B whereas for the second it is 5B. The authors should have said “mnemonic” or “assembly instruction” instead of “opcode”.

2f9. Write a program segment that initializes memory locations $0000 and $0001 with [the values] $12 and $10 using [assembler] directives.

For the Cosmic compiler we use in lab, we would replace the FCB directive in the authors’ answer

```
ORG $0000
FCB $12
FCB $10
```

with the `dc.b` directive.

2a2. Suppose that accumulator A contains $45. If we perform instruction SUBA #$44, which of the following branch instructions will be activated? BHE, BNE, BLT, BHT, BLE, BNZ.

Well, since BHE, BLT, and BNZ are not part of the 68HC12 instruction set, they certainly won’t be activated. The author probably meant BGE, BLS, and BNE. Then the answer would be BGE, BNE, and BHT.

Why? If A contains $45 and we subtract $44 from it we have $01 in A. We check Appendix A (68HC12 Instruction Set) and find that the SUBA instruction affects the N, Z, V, and C flags. Accordingly, the lower nibble of the CCR will be $0000.

Referring to pp. 55-56 of the text, we then see that if the next instruction after the SUBA instruction were

**BGE** (Branch if Greater than or Equal),

**BNE** (Branch if Not Equal), or

**BHS** (Branch if Higher or Same),

the branch would occur.
2a8. Write a program segment to add $20 and $40 and store the result to memory location $4000.

Obviously there are lots of possible answers here. For example, instead of adding immediate $40 to register A (ADDA #$40), we could first place the $20 into register B (LDAB #$40) and then add the value in B to that in A (ABA). Since this would take two extra clock cycles (see Appendix A) to do the same thing, it would not be as efficient, so we would generally prefer the single instruction. However, if we were going to need the value $20 in register B for subsequent instructions, it might make sense to load it here.