Homework #3
Due Wed. 10/3

1. (H&J 3.1)
   A program contains the following instruction mix:

   - 20% load/store instructions with execution time of 3 nsec each
   - 70% ALU instructions with execution time of 2.1 nsec each
   - 10% branch instructions with execution time of 2.3 nsec each

   (a) If the clock period is 500 psec, calculate the average CPI for the program.
   (b) What is the average MIPS rate of the program?
   (c) recalculate (a) with 40% load/store, 55% ALU, and 5% branch mix.

2. (H&J 3.2)
   In each of the following lines of MC68000 code, what value will be in the destination after execution of each instruction? Also, what addressing modes are used by each instruction? The following conditions apply:
   A0 = 0x1000 M[0x1000] = 0xABCD M[0xFFE] = 0xDCBA

   (a) MOVE.L #'AZB', D0
   (b) MOVE.L #0xABCDEF1, (A0)
   (c) MOVE.W A0, D0
   (d) MOVE.W -(A0), D0

3. (H&J 3.3)
   Consider the short section of MC68000 code below, which is executed with the initial conditions given. What values would PC, D0, D1, and D2 have after execution?
   Initial conditions:
   D0 = 0x1 D1 = 0x2 D2 = 0x10

   Code:

```
        ORG     $2000
        ADD     D0, D1
        DBLE    D2, $2000
```

4. (H&J 3.4) Consider the following MC68000 assembly language program. (The pseudo-operation, DC.W N, Define Constant Word, reserves space for *one* word in memory and initializes it to the specified value (N) at load time.)
   Code:

```
X       EQU     1000
        ORG     2000
Y       DC.W    X
Z       DC.W    Y
        ORG     3000
①       MOVE.W  #X, D0
        MOVE.W  Y, D1
        MOVE.W  Z, A0
        MOVE.W  (A0)+, D2
```

5. (H&J 3.8)
   Generate hexadecimal machine code for the following MC68000 instructions.

   (a) MOVE.W (A3), (A4)
   (b) MOVE.W 0xFF(A4, D5), 0x1000
   (c) ADD.L #1, D3
   (d) ROL #3, D4
   (e) BLE -255

6. (H&J 3.9)

   (a) At what address is LOOP located in example of Figure 3.6?
   (b) What is the size of the programs in bytes?
   (c) Encode the program in hexadecimal notation.

7. (H&J 3.13 (1)) Recode the example MC68000 program for array search in Figure 3.6. Compare and contrast the instruction counts and the data and program memory accesses in the CISC and RISC processors.

8. (H&J 3.14 (1)) Repeat Exercise 3.13 (1) but code for SRC.

9. (H&J 3.15) Translate the following SPARC instructions into hexadecimal.

   (a) `addcc %r3, %r5, %o7`
   (b) `sethi 0x1250FFF, %i1`
   (c) `ldub [%i7 + 0x6EE], %r22`

10. (H&J 3.20) Write SPARC assembly instructions sequences to simulate the 14 MC68000 addressing modes found in Figure 3.2. Your instructions should load one of the registers (e.g. %rd) using each of the addressing modes.