

EE3376 – Embedded Systems - Fall 2007 - Exam 1 – MacDonald – Page 1

Student Name: _____ **Student ID:** _____

Show work with intermediate steps. Showing only the answer will get no credit.

Problem 1 – 30 points

Assume “*ldx [D,X]*” is the next instruction in line and is executed with the initial values shown in the table below.

a – Fill in all blank entries in the table before and after the execution of the instruction.

	Initial Value (before execution)	Final Value (After execution)
Accumulator A	\$00	
Accumulator B	\$03	
Accumulator D		
Accumulator X	\$4001	
Accumulator Y	\$4002	
Bit C of CCR	0	
Bit V of CCR	0	
Bit N of CCR	0	
Bit Z of CCR	0	
Memory \$4000	\$00	
Memory \$4001	\$01	
Memory \$4002	\$40	
Memory \$4003	\$06	
Memory \$4004	\$40	
Memory \$4005	\$00	
Memory \$4006	\$42	
Memory \$4007	\$43	

b – Can you think of a reason in C or Java that you might want this addressing mode?

Student Name: _____ Student ID: _____

Problem 2 30 points –

- a) Provide the machine code (hand assemble) the instructions:

lra
lra
staa 1,X+

Address	Contents
\$0C00	
\$0C01	
\$0C02	
\$0C03	
\$0C04	

- b) Given the contents of eeprom below and assuming the contents are instructions, create (hand dis-assemble) the corresponding assembly program.
Hint: more than one instruction is encoded below. Boundaries are important. The instructions are contrived so don't look for them to do something reasonable.

Address	Contents
\$0C00	\$75
\$0C01	\$10
\$0C02	\$00
\$0C03	\$55
\$0C04	\$3d

Student Name: _____ Student ID: _____

Problem 3

```

PORTB          EQU  $0001
PORTB_DDR     EQU  $0003
BLOCK_COUNT   EQU  $02
something:    clra
              movb #$FF, PORTB_DDR
loop:         ldab  #$55
              stab  A, X
              ldab  A, X
              cmpb  #$55
              bne   BADBIT
              inca
              cmpa  #BLOCK_COUNT
              bne   loop
              movb  #$00, PORTB
              rts
BADBIT       movb  #$FF, PORTB
              rts
    
```

	Initial value	After either rts
Register A	\$80	
Register B	\$FF	
Register D		
Register X	\$3000	
Register Y	\$3001	
Memory \$0001	\$08	
Memory \$0003	\$0A	
Memory \$3000	\$08	
Memory \$3001	\$0A	
Memory \$3002	\$10	
Memory \$3003	\$02	
Memory \$3004	\$ff	

5 point bonus: This code is not a contrived example. What is the point of this subroutine?