1. (10pt) Determine how many times the instruction “DBNE” in the sub-program below is executed if NUM=$432.

DLY:  
LDY  #NUM
LOOP:  
NOP
DBNE Y,LOOP
RTS

How long will it take to execute the above program? (Assume MC9S12C32 with 24 MHz bus clock.)

We want to create exactly 0.5000 msec. delay with the above program. Determine the value for NUM in hexadecimal number. (Assume MC9S12C32 with 24 MHz bus clock.)

How many times will the “DBNE” instruction be executed if the NUM is equated to $00.
2. (5pt) Write a short subroutine to change MC9S12C32’s SCI port baud rate to 115.2Kbaud. Assume 24MHz bus clock for MC9S12C32.

3. (15pt) Write a short program to generate a square wave signal on the PORTA bit 0. Your program must use the interrupt: the Timer Module Channel 2 Output Compare interrupt generation at every 125usec (8KHz rate). Your program must include the interrupt service routine.
4. (15pt) Write a short program to convert the signal at AN7 pin to 8 bit binary and transmit it through the SCI port. Convert the signal every 125usec, repeat the conversion and transmission forever. Your program must include all the necessary subroutines.
5. (15pt) Show the memory contents of the stack after the Real Time Interrupt (RTI) has been occurred to MC9S12C32. The RTI occurred during the execution of instruction at line 19. The interrupt handling has been initiated and the first instruction has been executed. Show the memory contents of the stack during the beginning of the execution of the first instruction from the interrupt service routine at ISRSWI.

```
01 $FFFE  ORG $FFFE
02 $FFFE F000  FDB $F000
03 $FF6  ORG $FF6
04 $FF6 F100  FDB $F100
05 $FF0  ORG $FF0
06 $FF0 F200  FDB $F200
07
08 $F000  ORG $F000
09 $F000 CF0C00  MAIN: LDS #$0C00
10 $F003 CCFF00  LDD #$FF00
11 $F006 B702  TAP
12 $F008 CEF0F0  LDX #$F0F0
13 $F00B CDABCD  LDY #$ABCD
14 $F00E 10EF  CLI
15 $F010 16F014  JSR SUB1
16 $F013 3F  SWI
17
18 $F014 A7  SUB1: NOP
19 $F015 A7  LOOP1: NOP
20 $F016 20FD  BRA LOOP1
21 $F018 3D  RTS
22
23 $F200  ORG $F200
24 $F200 3F  ISRRTI: SWI
25 $F201 A7  NOP
26  ***
27  ***
28 $F202 0B  RTIEND: RTI
29
30 $F100  ORG $F100
31 $F100 A7  ISRSWI: NOP
32  ***
33  ***
34 $F101 0B  SWIEND: RTI
35
36 $F102  END
```

$0BF0
$0BF8
$0C00
6. (20pt) Shown below is a MC9S12C32 memory interface.

(a) How many address lines are required by a 1K X 8 memory chip?

(b) RAM memory beginning address: ________________
RAM memory ending address: ________________

ROM memory beginning address: ________________
ROM memory ending address: ________________ (Specify all addresses in Hex.)

(c) From the above memory circuit, which memory chip(s) will be enabled during the data read cycle of:

   the "LDX $3456" instruction? (Specify one of a, b, c, d, none )
   the "LDY $5678" instruction? (Specify one of a, b, c, d, none )
   the "LDAA $ABCD" instruction? (Specify one of a, b, c, d, none )

(d) Design the address decoder circuit which will map 8k bytes of RAM starting at address $4000 and 8k bytes of ROM ending at address $BFFF. Label all inputs to the decoder. Also, label the two decoder outputs RAMselect and ROMselect.

For part (c) above, make the following change:
'Specify one of' -> 'Choose ones that are enabled'
7. (10pt) Signed binary numbers (2’s complement form) are received from the port C of the MC9S12C32 and they are stored on the stack. The two numbers are needed to be added and error must be indicated if it occurs. Given the main program below, write a short subroutine which will add the two numbers stored on the stack. The result on the stack is two bytes: one byte is the sum of two numbers and the other byte is the error indicator ($00$ for no error, $FF$ for error). Penalty applies if your subroutine is more than 12 lines. Your subroutine must be transparent to its caller.

ADD2:

```
MAIN:

LDAA PORTC
PSHA

LDAA PORTC
PSHA

JSR ADD2
PULA
STAA ANSW
PULA
BNE ERROR
```

```
```
8. (10pt) Write a short subroutine to add two 24 bit numbers in memory locations labeled N1 and N2. Store the result in N1. Design your subroutine as short as possible. Your subroutine must be transparent to its caller.

<table>
<thead>
<tr>
<th>MSB</th>
<th>LSB</th>
</tr>
</thead>
<tbody>
<tr>
<td>N1</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>N2</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Note: The diagram above represents memory locations N1 and N2, showing their bit positions as MSB (Most Significant Bit) and LSB (Least Significant Bit).