(5) Question 1. Precision in decimal digits

20/0.1 is 200 alternatives, which is $2 \frac{1}{2}$ decimal digits

(2) Part 2a. Choose A-E

B) From memory to 6812, or from input device to 6812

(2) Part 2b. Choose A-E

A) Real time

(2) Part 2c. Choose A-E

C) FIFO queue

(2) Part 2d. Choose A-E

B) Drop out occurs after a right shift or a divide, and an intermediate result loses its ability to represent all of the values.

(2) Part 2e. Choose A-E

C) The two terms describe exactly the same synchronization method.

(2) Part 3a. Specify RegB

170+125=295, RegB = 295-256 = 39

(2) Part 3b. Specify 0 or 1

Consider as unsigned, 170+125=295 is too big, so C=1

Consider as signed, 86+125=39 is ok, so V=0

(1) Part 3c. Specify 0 or 1

(2) Part 4a. Give the value for xxx

2 (p is next to top)

(3) Part 4b. Give the value for yyy

0 (q is on top)

SP -> locals
X -> oldX
X+2 -> return
X+4 -> port zzz = +4

(5) Question 5. Give the value for zzz

zzz = SP -> locals
X -> oldX
X+2 -> return
X+4 -> port zzz = +4

After the pushes
SP -> $43$
$21$
$65$
$87$
pula makes RegA = $43
After the leas 2,sp
SP -> $87$
pulb makes RegB = $87
so Reg D = $4387

(5) Question 6. Give the value

$C000$ equals $-16384$

value = $-16384/100 = -163.84$

(5) Question 7. Give the hexadecimal value

After the pushes
SP -> $43$
$21$
$65$
$87$
pula makes RegA = $43
After the leas 2,sp
SP -> $87$
pulb makes RegB = $87
so Reg D = $4387

(5) Question 8. Give the op code

It is signed, so we need b1e

(5) Question 9. Simplified memory cycles (you may or may not need all 5 entries)

<table>
<thead>
<tr>
<th>R/W</th>
<th>Addr</th>
<th>Data</th>
</tr>
</thead>
<tbody>
<tr>
<td>R</td>
<td>4000</td>
<td>EE</td>
</tr>
<tr>
<td>R</td>
<td>4001</td>
<td>71</td>
</tr>
<tr>
<td>R</td>
<td>3900</td>
<td>00</td>
</tr>
<tr>
<td>R</td>
<td>3901</td>
<td>01</td>
</tr>
</tbody>
</table>
(5) Question 10. Show the assembly code

```assembly
ldd 13, x
```

(10) Question 11. Hand assemble this program

<table>
<thead>
<tr>
<th>Address</th>
<th>Machine code</th>
<th>Source code</th>
</tr>
</thead>
<tbody>
<tr>
<td>$3900</td>
<td></td>
<td>org  $3900</td>
</tr>
<tr>
<td>$3900</td>
<td>aa</td>
<td></td>
</tr>
<tr>
<td>$3902</td>
<td>bb</td>
<td>rmb  2</td>
</tr>
<tr>
<td>$0044</td>
<td>TCNT</td>
<td>equ  $0044</td>
</tr>
<tr>
<td>$4300</td>
<td></td>
<td>org  $4300</td>
</tr>
<tr>
<td>$4300</td>
<td>$DE $44</td>
<td></td>
</tr>
<tr>
<td>$4302</td>
<td>$CE $43 $0C</td>
<td></td>
</tr>
<tr>
<td>$4305</td>
<td>$A6 $02</td>
<td></td>
</tr>
<tr>
<td>$4307</td>
<td>$7A $39 $02</td>
<td></td>
</tr>
<tr>
<td>$430A</td>
<td>$20 $F6</td>
<td></td>
</tr>
<tr>
<td>$430C</td>
<td>$00 $01 $02 $03 $04</td>
<td>cc fcb 0,1,2,3,4</td>
</tr>
</tbody>
</table>

(15) Question 12. Show the assembly code

* Input one character from SCI terminal
* Inputs: none      Outputs: RegB is ASCII character
* Registers modified: CCR

```assembly
RDRF     equ  $20
SCI_InChar brclr SCISR1,#RDRF,SCI_InChar   Gadfly wait for RDRF set
ldab SCIDRL   ASCII character code
rts
```

(20) Question 13. Show the assembly code

```assembly
Main  lds  #$4000
      bset DDRM,#$03  PM1,PM0 outputs
      bclr DDRT,#$03  PT1,PT0 inputs
      ldx  #Stop       Reg X => current state
      *Linked list interpreter
LL    ldaa PTT
      anda  #$03
      cmpa  #1
      beq  is1
      cmpa  #2
      beq  is2
      cmpa  #3
      beq  is3
      ldsa          ; 0,2,4,6
      adda  #4       ; 4,6,8,10
      ldx  A,x      ; next
      bra  LL
is0   movb  0,x,PTM  ; output In=0
      adda  #4       ; 4,6,8,10
      ldx  A,x      ; next
      bra  LL
is1   movb  1,x,PTM  ; output In=1
      ldx  6,x      ; next In=1
      bra  LL
is2   movb  2,x,PTM  ; output In=2
      ldx  8,x      ; next In=2
      bra  LL
is3   movb  3,x,PTM  ; output In=3
      ldx  10,x     ; next In=3
      bra  LL
org  $FFFE
fdb  Main        reset vector
```