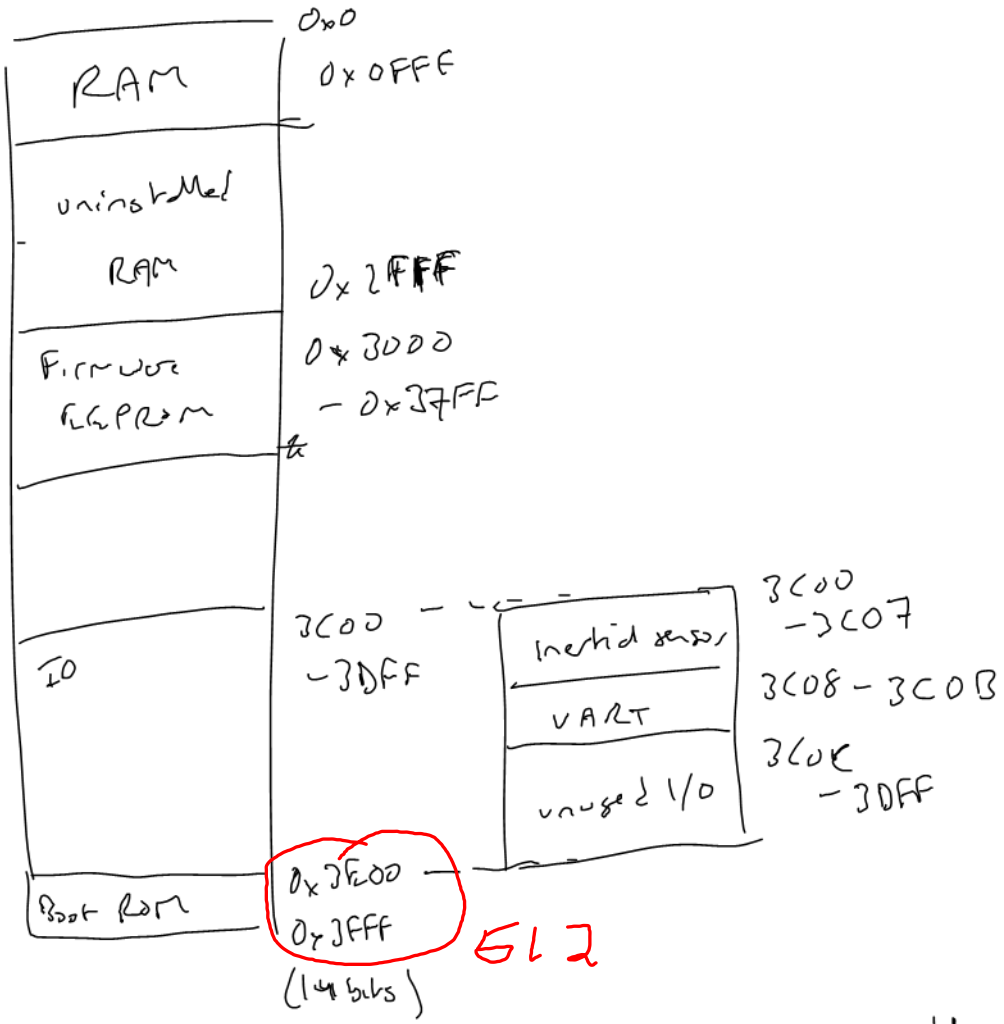
EE3302 s1 exam — 2016-2017

Q1 (a) (i) 10 addresses for — 3C00 - 3DFF

⇒ num addresses = 3DFF + 1 = 3E00
− 3C00
0x 200 = 512 decimal

(ii)



placed I/O devices on integer multiple of size for allow basic addr dec methods to be used.

Q1 (a) (iii) Addr decoding

Boot ROM

3E00 — 11 1110 0000 0000
3FFF — 11 1111 1111 1111
11 111X ——— X

Assume active low chip selects!

$\overline{\text{BootROM\_CS}} = (A_{13} A_{12} A_{11} A_{10} A_9)$

EEPROM

3000 — 11 0000 0000 0000
37FF — 11 0111 1111 1111
11 0X1 ——— X

$\overline{\text{EEPROM\_CS}} = (A_{13} A_{12} \overline{A_{11}})$

Internal RAM

0000 — 00 ——— 0
0FFF — 00 1111 1111 1111
00 X ——— X

$\overline{\text{RAM\_CS}} = (\overline{A_{13} \cdot A_{12}})$

Inertial sensor

3C00 — 11 1100 0000 0000
3C07 — 11 1100 0000 0111
11 1100 0000 0XXX

$\overline{\text{Inertial\_CS}} = (A_{13} A_{12} A_{11} A_{10} \overline{A_9} \overline{A_8} \overline{A_7} \overline{A_6} \overline{A_5} \overline{A_4} \overline{A_3})$

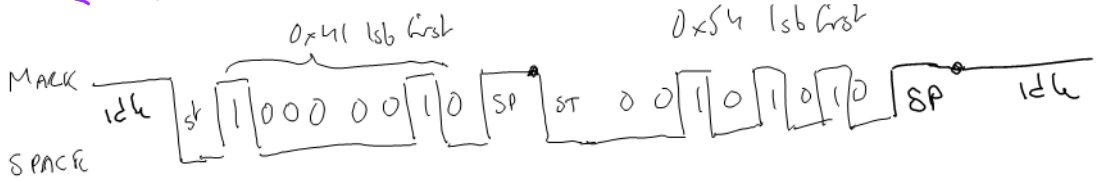
Q1 (b) (i) 115200, 8, N, 1

baud rate | bits/async word | parity type (N = none) | num stop bits per async word

(ii) each async word will be

start bit + 8 bit data (lsb first) + 1 stop bit
(there's no parity in this system)

0x41 0100 0001 ✓
0x54 0101 0100 ✓



(iii) sym rate = bit rate = 115200

$\Rightarrow T_{sym} = T_{bit} = \frac{1}{115200}$

$\Rightarrow T_{word} = \frac{10}{115200}$

$T_{msg} = \frac{19 \times 10}{115200}$ ✓

10 msg/sec = total time sending msgs in one sec = $\frac{10 \times 19 \times 10}{115200}$

= 16.49 ms

⇒ guard time is ~ 1.65 ms

10 x (10 bit x 19)

16.49 ms

Q2 (a) (i) see notes

(ii) " "

(iii) " "

(iv) see notes

(b)

T1CON = 0x2D – 0010 1101

(i) bitmask = 0011 0000 = 0x30

(ii) prescale : 0x03 – 0000 0011

tmp1 = T1CON & ~bitmask;

```
  0010 1101
& 1100 1111
-----------
  0000 1101
```

tmp2 = prescale << 4;

```
  0000 0011
  0011 0000
```

T1CON = tmp1 | tmp2:

```
  0000 1101
| 0011 0000
-----------
  0011 1101   = 0x3D
```

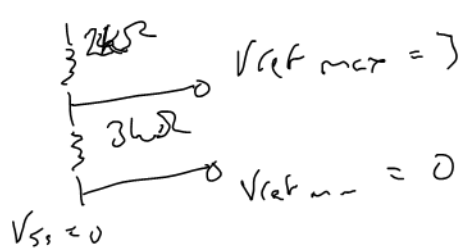
Q2 (c)

(i) bits/sample gives us the resolution/step size of ADC

Voltage reference values determine the voltage range that is divided into $2^{N\text{bits}}$ steps.

(ii) Change 0–3V

Suggest – resistor divider



(iii) $$\text{resolution} = \frac{V_{ref+} - V_{ref-}}{2^B} = \frac{3-0}{28} = \frac{3}{256} \approx 11.7\text{mV}$$

(iv) $R_{FSR} = 20000\Omega$ $\quad V_{out} = 5\left(\frac{10k}{10k+20k}\right) = 1.66\text{ V}$

$$\left\lfloor \frac{1.66 - V_{ref-}}{LSb} \right\rfloor = \left\lfloor \frac{1.66V - 0}{11.7mV} \right\rfloor = 142 \text{ (truncated for 142...)}$$

(v) $ADC = 208$

$$\Rightarrow V_{in} = (208.LSb) + V_{ref-}$$
$$= 208 \times 11.7mV + 0V$$
$$= 2.4375\text{ V}$$

$$2.4375 = 5\left(\frac{10k}{10k + R_{FSR}}\right)$$

$$\Rightarrow R_{FSR} = 10k\left(\frac{5}{2.4375} - 1\right) = 10.51k\Omega$$

double check

Q3 (a)

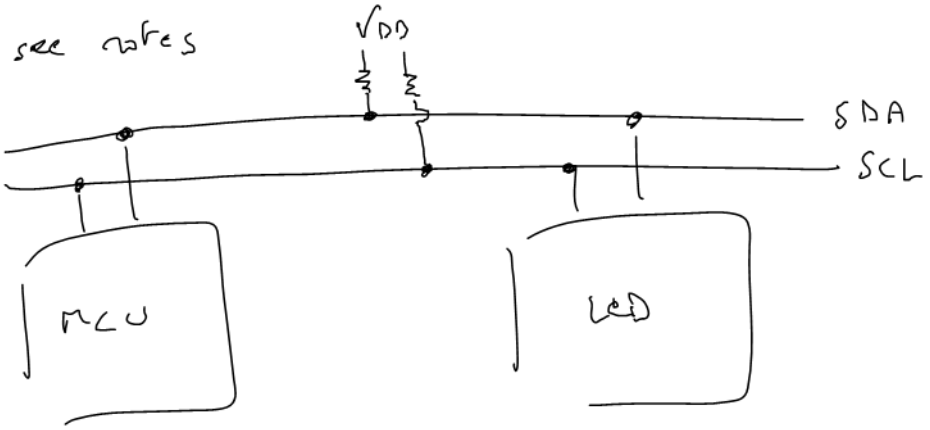
```
updateLed()
    constant NEXT_FLASH_TICKS = 35
    constant FLASH_DURATION_TICKS = 5
    static nextFlashCounter = 1
    static ledOnCounter = 0

    decrement nextFlashCounter
    if (nextFlashCounter == 0)
        set LED = HIGH        // switch on LED
        set ledOnCounter = FLASH_DURATION_TICKS
        set nextFlashCounter = NEXT_FLASH_TICKS
    else if (ledOnCounter > 0)
        decrement ledOnCounter
        if (ledOnCounter == 0)
            set LED = LOW     // switch off LED
```

Q3 (b) sec notes

Q3 (c) (i) see notes

(ii)

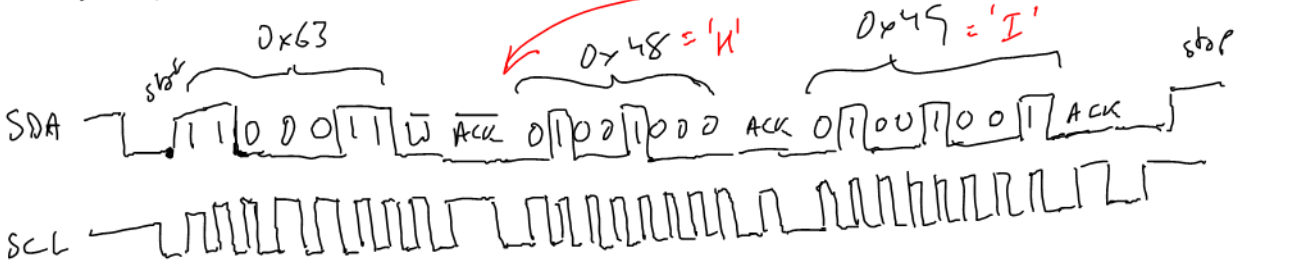


(iii) Protocol elements:

ST cond — master
+ I2C addr (0x63)
+ W
+ ACK
+ LCD register (0x08) + ACK — slave
+ 'H' + ACK
+ 'I' + ACK
+ Stop cond

(iv) data sent MSB first

forgot to put in LCD register 2 3



Q4 (a) (i) see notes

(ii) see notes

(iii) see notes

(b) (i) see notes

(ii) $f_{osc} = 20\,MHz$

$f_{src} = \frac{f_{osc}}{4} = 5\,MHz$

$\Rightarrow T_{src} = \frac{1}{f_{src}} = \frac{1}{5\,MHz} = 0.2\,\mu s$

$T_{src} \times prescale \times N \times postscale$

$N_{max} = 2^{16} = 65536$

prescale

1x : $0.2\,\mu s \times 1 \times 65536 \times 1$

$= 13.1072\,ms$ is max timeout

resolution $= 0.2\,\mu s \times 1 = 0.2\,\mu s$

8x : $0.2\,\mu s \times 8 \times 65536 \times 1 = 104.8576\,ms$

resolution $= 0.2 \times 8 = 1.6\,\mu s$

(iii) $30\,Hz$ rate $\Rightarrow$ duration $= \frac{1}{30}\,s = 33.33\,ms$

Allow $1.5\,\mu s$ for resetting timer etc

$\Rightarrow$ desired timeout $= 33333.\dot{3}\,\mu s - 1.5\,\mu s$

$= 33331.8\dot{3}\,\mu s$

$$\text{desired scale} = \frac{33331.8}{0.2 \times 65536} = 2.54$$

Nearest prescale value is 4x

Q4(b) (iii) (cont.)

$$N_{intr} = \text{round}\left(\frac{33331.8}{0.2 \times 4 \times 1}\right) = 41665$$

$$\text{Timer register value} = 2^8 - N_{intr} = 23871$$
$$= 65536 - 41665$$

(iv) Actual timeout will be

$$0.2 \times 4 \times 41665 \times 1 = 33332\ \mu s$$

Add in $1.5\mu s$ extra work and calc freq

$$f_{actual} = \frac{1}{33332\mu s + 1.5\mu s} = 29.99985\ Hz$$

$$\% \text{ error} = \left(1 - \frac{f_{actual}}{f_{desired}}\right) \times 100 = \left(1 - \frac{29.99985}{30}\right) \times 100$$

$$= 0.0005\ \%$$

Limited by the resolution of the timer with this prescale factor ($0.8\mu s$)