One 8.5 by 11 inch paper, two sided crib sheet. You will have two pages similar to Quiz 1 2016. Since the exam is closed book without calculator, you will need to know some basic constants:

- $2^0 = 1 = 0x01$
- $2^1 = 2 = 0x02$
- $2^2 = 4 = 0x04$
- $2^3 = 8 = 0x08$
- $2^4 = 16 = 0x10$
- $2^5 = 32 = 0x20$
- $2^6 = 64 = 0x40$
- $2^7 = 128 = 0x80$
- $2^8 = 256 = 0x100$
- $2^9 = 512 = 0x200$
- $2^{10} = 1024 = 0x400$
- $2^{11} = 2048 = 0x800$
- $2^{12} = 4096 = 0x1000$
- $2^{13} = 8192 = 0x2000$
- $2^{14} = 16384 = 0x4000$
- $2^{15} = 32768 = 0x8000$
- $2^{16} = 65536$

Range of integers for 8-bit signed (-128 to +127), 8-bit unsigned (0 to 255)

Range of integers for 16-bit signed (-32768 to +32767), 16-bit unsigned (0 to 65535)

**Fixed-point numbers (decimal/binary, signed/unsigned, 8/16/32 bit)**

1) Convert value to integer. E.g., What integer is stored in the computer, when the value 2.1 is stored in 16-bit unsigned binary fixed-point, with a resolution of $2^{-10}$? Answer: $I = 2.1 \times 1024$, which is about 2150. I will make the math so easy a calculator will not be needed.

2) Convert integer to value. E.g., What is the value of an 8-bit signed decimal fixed-point number (resolution is 0.1) if the integer stored in memory is $-123$? Answer: $-123 \times 0.1$ equals $-12.3$.

3) Basic concepts of range, resolution and rounding.

4) Given the range and resolution, choose the format

**Programming (C on the Arm Cortex M) Techniques**

1) How to check for overflow when performing integer calculations in C. Answer: promote to higher precision and check intermediate results.

2) Stack picture: function call, function parameters, locals, interrupt service stack frame

3) Memory allocation: global, local, constants, and reset/interrupt vectors

4) Modularity linking call graph to #include 'file.h', private versus public

5) **const volatile static**

6) Data flow graph

7) Debugging: dumps, monitors, scans, breaks, filter, profiling

8) Putting prototypes to public functions in the 'file.h', implementations in the 'file.c'

**I/O concepts (on the TM4C123): focus on concepts and don’t memorize details**

1) Direction register. Friendly means setting only the bits that are needed, leaving the rest unchanged.

2) Concept of bit-specific access to GPIO ports. Not details, but the concepts of friendly and critical sections

3) Using SysTick to measure elapsed time, and to create a time delay

4) Periodic timer interrupts: rate, and vector (jitter)

**Detailed I/O programming (on the TM4C123)**

1) FIFO implementation

2) Periodic timer interrupts using Timer0A and/or SysTick

3) Debugging dumps, profiles

4) Mailbox and semaphore

5) Edge-triggered interrupt on Port A (Fall 2015 was Port B, in class we did Port F)

**Interfacing**

1) Switches with debouncing and LEDs

2) PN2222 interface (saturated mode, off mode, selection of base resistor);

3) Speaker (where software creates a squarewave)

4) Simple circuit model for N-channel BJT and/or Darlington transistors in on/off applications

**IoT**

IP addresses, DNS (what, why), UDP (what, why), TCP (what, why), sockets (types, what, why), Steps involved in client-server paradigm
Definitions (match definition with the following terms)
Real-time, friendly, latency, ROM, RAM, computer, CPU, Harvard architecture, von Neumann architecture processor, ALU, BIU, CU, registers, bus, embedded computer, microprocessor, microcomputer, microcontroller, I\text{OL}, I\text{IH}, I\text{IL}, I\text{IH}, V\text{OL}, V\text{OH}, V\text{IL}, V\text{IH}, capacitive loading, nonvolatile, open collector, tristate, memory-mapped I/O, dynamic efficiency, static efficiency, functional debugging, performance debugging, non-intrusiveness, profile, desk check, instrument, stabilize, scan, break, thread, busy wait, CPU bound, I/O bound, atomic, critical section, reentrant, interrupt vector, interrupt acknowledge, interrupt arm, interrupt enable, interrupt priority, board support package, functional abstraction, complexity abstraction, cohesion, coupling, latency, power budget, probability mass function, Central Limit Theorem

Variables
- RAM versus ROM
- Public scope versus private scope
- Temporary versus permanent allocation
- Precision

Know voltage current power in R, L or C. Time constant, step response (RC RL LC circuits)

Old Quizzes and Exams (replace RTI/output compare with SysTick or Timer0A)
Definitions
- Spring 2003 Final, Question 12, desk checking
- Spring 2003 Final, Question 14, interface latency
- Spring 2003 Final, Question 15, polled interrupt
- Spring 2003 Final, Question 16, private
- Spring 2003 Final, Question 17, intrusive
- Spring 2003 Quiz 1, Question 12, scanpoint
- Spring 2003 Quiz 1, Question 13, stabilization
- Spring 2003 Quiz 1, Question 14, profile
- Spring 2003 Quiz 1, Question 15, busy-waiting
- Spring 2003 Quiz 1, Question 1-5, Where in memory are variables allocated?
- Spring 2003 Quiz 1, Question 16, real time system
- Spring 2003 Quiz 1, Question 17, interrupt acknowledge
- Spring 2003 Quiz 1, Question 18, ALU
- Spring 2003 Quiz 1, Question 19, volatile
- Spring 2003 Quiz 1, Question 20, output low current
- Spring 2003 Quiz 1, Question 21, nonintrusive
- Spring 2003 Quiz 1, Question 22, friendly, set direction register, toggle output
- Spring 2003 Quiz 1, Question 6, Data flow graph
- Spring 2004 Quiz 1, Question 1, Which variables are stored on the stack?
- Spring 2004 Quiz 1, Questions 6,9,12, volatile, open collector, busy-waiting
- Spring 2004 Quiz 1, Questions 8,10,13, nonintrusive, stabilization, real time
- Spring 2004 Quiz 2, Question 3, FIFO principles
- Spring 2005 Final, Question 10, Fixed point
- Spring 2005 Final, Question 16, Desk check
- Spring 2005 Final, Question 17, Bandwidth
- Spring 2005 Final, Question 18, Latency
- Spring 2005 Final, Question 19, Polled interrupt
- Spring 2005 Final, Question 2, Critical section
- Spring 2005 Final, Question 20, Private
- Spring 2005 Final, Question 21, Nonintrusive
- Spring 2005 Final, Question 22, Buffered I/O
- Spring 2005 Final, Question 24, Associative principle
- Spring 2005 Final, Question 3, What does \texttt{short} mean?
- Spring 2005 Final, Question 4, What does \texttt{const} mean?
- Spring 2005 Final, Question 5, What does \texttt{static} mean?
Spring 2005 Final, Question 7, Debugging instruments
Spring 2005 Quiz 1, Question 2, Intrusive
Spring 2005 Quiz 1, Question 3, Open collector
Spring 2005 Quiz 1, Question 4, Stabilizing
Spring 2005 Quiz 1, Question 5, Tristate logic
Spring 2005 Quiz 1, Question 6, Latency
Spring 2006 Quiz 1, Question 1, I/O bound
Spring 2006 Quiz 1, Question 2, vectored interrupt
Spring 2006 Quiz 1, Question 3, intrusive
Spring 2006 Quiz 1, Question 4, stabilization
Spring 2006 Quiz 1, Question 5, tristate
Spring 2006 Quiz 1, Question 6, latency
Spring 2006 Quiz 1, Question 7, buffered I/O
Spring 2007 Quiz 1, Questions 1-5, Intrusive
Fall 2007 Quiz 1, Question 1, Intrusive
Fall 2007 Quiz 1, Question 2, Debugging instrument, which interrupt occurs first?
Spring 2007 Quiz 1, Questions 8-12, Where are variables allocated?
Fall 2007 Quiz 1, Question 4, Interface latency
Spring 2008 Quiz 1, Question 2, Intrusive
Spring 2008 Quiz 1, Question 6, Where are local variables allocated?
Spring 2008 Quiz 1, Question 8, $V_{IH}$, $V_{IL}$
Fall 2008 Quiz 1, Question 6, Where are variables allocated?
Fall 2009 Quiz 1, Question 1, latency
Fall 2009 Quiz 1, Question 2, reentrant
Fall 2010 Quiz 1, Intrusive debugging, looking at assembly code
Fall 2011 Quiz 1, debugging profile
Fall 2011 Quiz 1, semaphore application
Fall 2012 Quiz 1, Questions 1-2, What is LR, stabilize, profile, intrusive?
Fall 2012 Quiz 1, Question 3, Is the FIFO reentrant?
Fall 2012 Quiz 1, Question 5, Scope of a variable
Fall 2012 Quiz 1, Question 6, Voltage and current across a capacitor
Fall 2015 Quiz 1, Question 1, Probability Mass Function, Central Limit Theorem
Fall 2017 Quiz 1, Question 7, UDP vs TCP
Fall 2016 Quiz 1, Question 1, Central Limit Theorem
Fall 2016 Quiz 1, Question 1, const, static
Fall 2016 Quiz 1, Question 1, Connection socket

Interrupts (replace RTI/output compare with SysTick or Timer0A)
Fall 2012 Quiz 1, Question 7, Write code to measure time jitter using SysTick
Fall 2014 Quiz 1, Question 1, debugging, FIFOs and CPU bound
Fall 2014 Quiz 1, Question 2, Critical section
Fall 2014 Quiz 1, Question 4, Edge-triggered interrupts
Fall 2015 Quiz 1, Question 2, Critical section
Fall 2015 Quiz 1, Question 5, SysTick interrupts, debounce switches
Fall 2016 Quiz 1, Question 2, Critical section
Fall 2016 Quiz 1, Question 6, Edge-triggered interrupts
Fall 2016 Quiz 1, Question 7, debugging interrupts
Fall 2016 Quiz 1, Question 8, SysTick interrupts

Fixed-Point
Spring 2005 Final, Question 30, Fixed point multiply
Spring 2004 Quiz 1, Questions 2,3,5, Fixed-point numbers
Spring 2005 Quiz 1, Questions 1,7, Fixed-point numbers
Spring 2003 Quiz 1, Questions 7-10, Fixed-point numbers
Spring 2006 Quiz 1, Question 13, How to store the integer part of a fixed-point number
Spring 2007 Quiz 1, Question 6, How to store the integer part of a fixed-point number
Fall 2007 Quiz 1, Question 5, Choosing fixed-point format
Spring 2008 Quiz 1, Question 3, Choosing fixed-point format
Spring 2008 Quiz 1, Question 4, Writing fixed-point math software
Fall 2008 Quiz 1, Question 4, Fixed-point math, implemented in C.
Fall 2009 Quiz 1, Question 4, binary fixed point.
Fall 2010 Quiz 1, Question 5, Fixed-point math, implemented in C
Fall 2011 Quiz 1, Question 6, Fixed-point math, implemented in C
Fall 2012 Quiz 1, Question 4, binary fixed point
Fall 2015 Quiz 1, Question 8, fixed point divide
Fall 2016 Quiz 1, Question 5, decimal fixed point

Hardware interfacing
Spring 2003 Final, Question 5, Solid state relay
Spring 2003 Quiz 2, Question 3, LED interface
Spring 2004 Final, Question 4, Solid state relay
Spring 2004 Quiz 2, Question 4, LED interface
Spring 2005 Quiz 2, Question 4, LED interface
Spring 2006 Quiz 1, Question 11, LED interface
Spring 2006 Quiz 1, Question 14, Switch debouncing using interrupts
Spring 2007 Quiz 1, Question 14, LED interface
Fall 2007 Quiz 1, Question 6, LED interface
Spring 2008 Quiz 1, Question 7, LED interface
Fall 2008 Quiz 1, Question 3, LED interface.
Fall 2009 Quiz 1, Question 3, $V_{OL}$, $V_{IL}$, $V_{OH}$, $V_{IH}$ interfacing. 
Fall 2011 Quiz 1, Question 3, Xbee interface ($V_{OL}$, $V_{IL}$, $V_{OH}$, $V_{IH}$) 
Fall 2011 Quiz 1, Question 7, LED interface (100 mA).
Fall 2012 Quiz 1, Question 6, V and I for ideal capacitor
Fall 2014 Quiz 1, Question 3, $V_{OL}$, $V_{IL}$, $V_{OH}$, $V_{IH}$ interfacing
Fall 2014 Quiz 1, Question 5, V and I for ideal capacitor
Fall 2014 Quiz 1, Question 7, BJT (treat it like a speaker)
Fall 2015 Quiz 1, Question 3, $V_{OL}$, $V_{IL}$, $V_{OH}$, $V_{IH}$ interfacing
Fall 2015 Quiz 1, Question 5, BJT interface of a speaker
Fall 2016 Quiz 1, Question 3, $V_{OL}$, $V_{IL}$, $V_{OH}$, $V_{IH}$ interfacing
Fall 2016 Quiz 1, Question 4, V and I for ideal capacitor
Fall 2016 Quiz 1, Question 9, BJT interface of a 50-mA LED
You may use the I/O definitions from either header file \texttt{tm4c123gh6pm.h} in your answers without showing the define statements. E.g., these apply to all software

\begin{verbatim}
#include "tm4c123gh6pm.h"
\end{verbatim}

These tables will be supplied (along the exam will use NPN and not PNP transistors)

<table>
<thead>
<tr>
<th>Parameter</th>
<th>PN2222 (I_C=150mA)</th>
<th>PN2907 (I_C=150mA)</th>
<th>2N2222 (I_C=500mA)</th>
<th>2N2907 (I_C=500mA)</th>
<th>TIP120 (I_C=3A)</th>
<th>TIP125 (I_C=3A)</th>
</tr>
</thead>
<tbody>
<tr>
<td>$h_{fe}$</td>
<td>100</td>
<td>40</td>
<td>1000</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$V_{BE\text{sat}}$</td>
<td>0.6</td>
<td>2</td>
<td>2.5 V</td>
<td>2.5 V</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$V_{CE}$ at saturation</td>
<td>0.3</td>
<td>1</td>
<td>2 V</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

These parameters will be included for the TM4C12xx microcontroller (no 12mA mode will be used)

$I_{OL} = 8\text{mA}, \quad I_{OH} = 8\text{mA}, \quad I_{IL} = 2\mu\text{A}, \quad I_{IH} = 2\mu\text{A}$,

$V_{OL} = 0.4\text{V}, \quad V_{OH} = 2.4\text{V}, \quad V_{IL} = 1.3\text{V}, \quad V_{IH} = 2.0 \text{V}$

These parameters will be included for the TM4C12xx microcontroller (no 12mA mode will be used)

$I_{OL} = 8\text{mA}, \quad I_{OH} = 8\text{mA}, \quad I_{IL} = 2\mu\text{A}, \quad I_{IH} = 2\mu\text{A}$,

$V_{OL} = 0.4\text{V}, \quad V_{OH} = 2.4\text{V}, \quad V_{IL} = 1.3\text{V}, \quad V_{IH} = 2.0 \text{V}$

Address | 31-24 | 23-17 | 16 | 15-3 | 2 | 1 | 0 | Name                  
------- |-------|-------|----|------|---|----|---|----------------------|
$0000E010$ | 0 | 0 | COUNT | 0 | CLK SRC | INTEN | ENABLE | NVIC ST CTRL R |
$0000E014$ | 0 | 24-bit RELOAD value | NVIC ST RELOAD R |
$0000E018$ | 0 | 24-bit CURRENT value of SysTick counter | NVIC ST CURRENT R |

Address | 31-29 | 28-24 | 23-21 | 20-8 | 7-5 | 4-0 | Name                  
------- |-------|-------|-------|------|-----|-----|----------------------|
$0000ED20$ | TICK | 0 | PENDSV | 0 | DEBUG | 0 | NVIC SYS PRI3 R |

Address | 31-3 | 2-0 | Name                  
------- |------|-----|----------------------|
$4003.0000$ | GPTMCFG | TIMER0_CFG_R |
$4003.0004$ | TAAMS | TACMR | TAMR | TIMER0_TAMR_R |
$4003.001C$ | TBPWML | TBOTE | TBEVENT | TBEN | TAPWML | TAOTE | TAEVENT | TREN | TIMER0_CTL_R |
$4003.0018$ | CAEIM | CBMIM | TBTOIM | CMIM | TATOIM | TIMEM0.IMR_R |
$4003.001C$ | CBERIS | CBMRIS | TBTRIS | CAMIS | TATISR | TIMER0_RIS_R |
$4003.001C$ | CBEMIS | CBMMIS | TBTOFS | CAMIS | TATOMIS | TIMER0_MIS_R |
$4003.0020$ | CBECINT | CBMCINT | TBCINT | CAMCINT | TATOCINT | TIMER0_ICR_R |
$4003.0028$ | TAILRH | TAILRL | TIMER0_TAILR_R |
$4003.0030$ | TAMRH | TAMRL | TAMATCHR_R |
$4003.0038$ | TAPS | TAPR | TIMER0_TAPR_R |
$4003.0040$ | TAPSM | TAPMR | TIMER0_TAPMR_R |
$4003.0048$ | TARH | TAR | TIMER0_TAR_R |
<table>
<thead>
<tr>
<th>Address</th>
<th>31 – 29</th>
<th>23 – 21</th>
<th>15 – 13</th>
<th>7 – 5</th>
<th>Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>0xE000E400</td>
<td>GPIO Port D</td>
<td>GPIO Port C</td>
<td>GPIO Port B</td>
<td>GPIO Port A</td>
<td>NVIC PRI0 R</td>
</tr>
<tr>
<td>0xE000E404</td>
<td>SS10, Rx Tx</td>
<td>UART1, Rx Tx</td>
<td>UART0, Rx Tx</td>
<td>GPIO Port E</td>
<td>NVIC PRI1 R</td>
</tr>
<tr>
<td>0xE000E408</td>
<td>PWM Gen 1</td>
<td>PWM Gen 0</td>
<td>PWM Fault</td>
<td>I2C0</td>
<td>NVIC PRI2 R</td>
</tr>
<tr>
<td>0xE000E40C</td>
<td>ADC Seq 1</td>
<td>ADC Seq 0</td>
<td>Quad Encoder</td>
<td>PWM Gen 2</td>
<td>NVIC PRI3 R</td>
</tr>
<tr>
<td>0xE000E410</td>
<td>Timer 0A</td>
<td>Watchdog</td>
<td>ADC Seq 3</td>
<td>ADC Seq 2</td>
<td>NVIC PRI4 R</td>
</tr>
<tr>
<td>0xE000E414</td>
<td>Timer 2A</td>
<td>Timer 1B</td>
<td>Timer 1A</td>
<td>Timer 0B</td>
<td>NVIC PRI5 R</td>
</tr>
<tr>
<td>0xE000E418</td>
<td>Comp 2</td>
<td>Comp 1</td>
<td>Comp 0</td>
<td>Timer 2B</td>
<td>NVIC PRI6 R</td>
</tr>
<tr>
<td>0xE000ED20</td>
<td>SysTick</td>
<td>PendSV</td>
<td>--</td>
<td>Debug</td>
<td>NVIC_SYS_PRI3 R</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Address</th>
<th>31</th>
<th>30</th>
<th>29-7</th>
<th>6</th>
<th>5</th>
<th>4</th>
<th>3</th>
<th>2</th>
<th>1</th>
<th>0</th>
<th>Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>0xE000E100</td>
<td>G</td>
<td>F</td>
<td>...</td>
<td>UART1</td>
<td>UART0</td>
<td>E</td>
<td>D</td>
<td>C</td>
<td>B</td>
<td>A</td>
<td>NVIC EN0 R</td>
</tr>
<tr>
<td>0xE000E104</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>UART2</td>
<td>H</td>
<td>NVIC EN1 R</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>