Labs 1,2,3,4               Lectures 1-9 (no FSM, stepper, SPI, DAC)
One 8.5 by 11 inch paper, two sided crib sheet.
Since the exam is closed book without calculator, you will need to know some basic constants:

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

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 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) \texttt{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 TM4C12xx): 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) Masking input bits to check for individual signals, making individual output bits high, low, or toggle.
4) Using SysTick to measure elapsed time, and to create a time delay
4) Periodic timer interrupts: rate, and vector (jitter)
5) UART input/output (busy-wait and interrupt synchronization) (concepts but no detailed programming)
6) Concepts of PWM, input capture, and input capture interrupts (concepts but no detailed programming)

Detailed I/O programming (on the TM4C12xx)
1) FIFO implementation
2) Digital controller implementation
3) Periodic timer interrupts using Timer0A and/or SysTick
4) Debugging dumps, profiles
5) Mailbox and semaphore
6) Edge-triggered interrupt on Port B

Interfacing
1) Switches (debouncing) and LEDs
2) DC motors (model for motor coil, back emf);
3) Speaker (where software creates a squarewave)
4) Solid state relay (just like an LED)
5) Solenoid or electromagnetic relay (just like one coil of a DC motor)
6) Simple circuit model for N-channel BJT and/or Darlington transistors in on/off applications
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, read cycle, write cycle, address bus, data bus, control bus, embedded computer, microprocessor, microcomputer, microcontroller, \( I_{OL}, I_{OH}, I_{IL}, I_{IH}, V_{OL}, V_{OH}, V_{IL}, V_{IH} \), 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, board support package, functional abstraction, complexity abstraction, cohesion, coupling, latency, back emf.

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. $V_{HH}$, $V_{IL}$
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

**Interrupts (replace RTI/output compare with SysTick or Timer0A)**

Spring 2003 Final, Question 19e, acknowledge RTI
Spring 2003 Final, Question 3, What happens if a ISR does not acknowledge?
Spring 2005 Final, Question 1, Items on the stack during the execution of an interrupt service routine
Spring 2005 Final, Question 9, What three conditions cause a RTI interrupt?
Spring 2007 Quiz 1, Question 7, Critical section
Spring 2007 Quiz 1, Question 13, What causes an output compare interrupt?
Fall 2007 Quiz 1, Question 3, Debugging instrument, critical section
Fall 2007 Quiz 1, Question 9, Output compare software
Spring 2008 Quiz 1, Question 5, How do we make a system real time?
Fall 2008 Quiz 1, Question 2, Critical section.
Fall 2010 Quiz 1, Question 2, Write code to see which interrupt occurs first
Fall 2011 Quiz 1, Question 4, Critical section in ISR debugging.
Fall 2012 Quiz 1, Question 7, Write code to measure time jitter using SysTick

**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

FSM (no FSM in Fall 2014)
- Spring 2003 Final, Question 20, FSM controller
- Spring 2003 Quiz 2, Question 2, FSM controller
- Spring 2004 Final, Question 11, FSM controller
- Spring 2004 Quiz 2, Question 6, FSM controller
- Spring 2005 Final, Question 25, FSM analysis
- Spring 2005 Quiz 2, Question 7, Output compare-driven FSM
- Spring 2006 Quiz 2, Question 7, Output compare-driven FSM
- Spring 2007 Quiz 1, Question 16, Output compare-driven FSM
- Fall 2007 Quiz 1, Question 8, Drawing a FSM state graph
- Fall 2008 Quiz 1, Question 5, Interpret a FSM (what is the output given the input)
- Fall 2008 Quiz 1, Question 7, Drawing a FSM state graph, implement using interrupts
- Spring 2008 Quiz 1, Question 9, Output compare-driven FSM
- Fall 2009 Quiz 1, Question 5, Drawing a FSM state graph.
- Fall 2010 Quiz 1, Question 3, Drawing a FSM state graph
- Fall 2011 Quiz 1, Question 5, Convert between Mealy and Moore FSM
- Fall 2012 Quiz 1, Question 8, FSM design (problem statement to graph)

Hardware interfacing
- Spring 2003 Final, Question 5, Solid state relay
- Spring 2003 Quiz 2, Question 3, LED interface
- Spring 2003 Quiz 2, Question 4, Motor interface
- Spring 2004 Final, Question 4, Solid state relay
- Spring 2004 Quiz 2, Question 2 Solenoid interface
- Spring 2004 Quiz 2, Question 4, LED interface
- Spring 2005 Final, Question 29, relay interface
- Spring 2005 Quiz 2, Question 2, Solenoid interface
- Spring 2005 Quiz 2, Question 4, LED interface
- Spring 2005 Quiz 2, Question 6, Stepper motor fundamentals
- 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
- Spring 2007 Quiz 1, Question 15, Stepper interface
- Fall 2007 Quiz 1, Question 6, LED interface
- Fall 2007 Quiz 1, Question 7, Solenoid interface
- Spring 2008 Quiz 1, Question 7, LED interface
- Fall 2008 Quiz 1, Question 7, electromagnetic relay 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 2009 Quiz 1, Question 6, stepper motor jerk.
- Fall 2009 Quiz 1, Question 6, electromagnetic relay interface.
- Fall 2010 Quiz 1, Question 6, DC motor interface
- 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 9, DC motor interface using a TIP120
You may use the I/O definitions from either header file tm4c123gh6pm.h or tm4c1294ncpdt.h in your answers without showing the define statements. E.g., these apply to all software

```
#include "tm4c1294ncpdt.h"
#include "tm4c123gh6pm.h"
```

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>2N2222 (I_c=500mA)</th>
<th>PN2907 (I_c=150mA)</th>
<th>2N2907 (I_c=500mA)</th>
<th>TIP120 (I_c=3A)</th>
<th>TIP125 (I_c=3A)</th>
</tr>
</thead>
<tbody>
<tr>
<td>hfe</td>
<td>100</td>
<td>40</td>
<td>1000</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>V_BE_sat</td>
<td>0.6</td>
<td>2</td>
<td>2.5 V</td>
<td></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} = 8mA, I_{OH} = 8mA, I_{IL} = 2μA, I_{IH} = 2μA,
V_{OL} = 0.4V, V_{OH} = 2.4V, V_{IL} = 1.3V, V_{IH} = 2.0 V

<table>
<thead>
<tr>
<th>Address</th>
<th>31-24</th>
<th>23-17</th>
<th>16</th>
<th>15-3</th>
<th>2</th>
<th>1</th>
<th>0</th>
<th>Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>$E000E010</td>
<td>0</td>
<td>0</td>
<td>COUNT</td>
<td>0</td>
<td>CLK_SRC</td>
<td>INTEN</td>
<td>ENABLE</td>
<td>NVIC_ST_CTRL_R</td>
</tr>
<tr>
<td>$E000E014</td>
<td>0</td>
<td>24-bit RELOAD value</td>
<td>NVIC_ST_RELOAD_R</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$E000E018</td>
<td>0</td>
<td>24-bit CURRENT value of SysTick counter</td>
<td>NVIC_ST_CURRENT_R</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Address</th>
<th>31-29</th>
<th>28-24</th>
<th>23-21</th>
<th>20-8</th>
<th>7-5</th>
<th>4-0</th>
<th>Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>$E000ED20</td>
<td>TICK</td>
<td>0</td>
<td>PENDSV</td>
<td>0</td>
<td>DEBUG</td>
<td>0</td>
<td>NVIC_SYS_PRI3_R</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Name</th>
<th>31-3</th>
<th>2-0</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>GPTMCFG</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Name</th>
<th>31-4</th>
<th>3</th>
<th>2</th>
<th>1-0</th>
</tr>
</thead>
<tbody>
<tr>
<td>TAAMS</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TACMR</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TAMR</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TIMER0_TAMR_R</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Name</th>
<th>14</th>
<th>13</th>
<th>11-10</th>
<th>8</th>
<th>6</th>
<th>5</th>
<th>3-2</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>TBPWML</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TBOTE</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TBEVENT</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TBEN</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TAPWML</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TAOTE</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TAEVENT</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TAEN</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TIMER0_CTL_R</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Name</th>
<th>31-11</th>
<th>10</th>
<th>9</th>
<th>8</th>
<th>7-4</th>
<th>2</th>
<th>1</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>CBEMIS</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CBMMIS</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TBTOMIS</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CAEMIS</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CAMMIS</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TATOMIS</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TIMER0_MIS_R</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Name</th>
<th>31-11</th>
<th>10</th>
<th>9</th>
<th>8</th>
<th>7-4</th>
<th>2</th>
<th>1</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>CBECINT</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CBMCINT</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TBTOCINT</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CAECINT</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CAMCINT</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TATOCINT</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TIMER0_ICR_R</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Name</th>
<th>31-16</th>
<th>15-0</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>TAILRH</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TAILRL</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TIMER0_TAILR_R</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Name</th>
<th>31-16</th>
<th>15-0</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>TAMRH</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TAMRL</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TAMATCHR_R</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Name</th>
<th>31-8</th>
<th>7-0</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>TAPSR</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TIMER0_TAPR_R</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Name</th>
<th>31-8</th>
<th>7-0</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>TAPSMR</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TIMER0_TAPMR_R</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Name</th>
<th>31-16</th>
<th>15-0</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>TARH</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TARL</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TIMER0_TAR_R</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Address</td>
<td>31 – 29</td>
<td>23 – 21</td>
<td>15 – 13</td>
</tr>
<tr>
<td>---------</td>
<td>---------</td>
<td>---------</td>
<td>---------</td>
</tr>
<tr>
<td>0xE000E400</td>
<td>GPIO Port D</td>
<td>GPIO Port C</td>
<td>GPIO Port B</td>
</tr>
<tr>
<td>0xE000E404</td>
<td>SS10, Rx T</td>
<td>UART1, Rx T</td>
<td>UART0, Rx T</td>
</tr>
<tr>
<td>0xE000E408</td>
<td>PWM Gen 1</td>
<td>PWM Gen 0</td>
<td>PWM Fault</td>
</tr>
<tr>
<td>0xE000E410</td>
<td>ADC Seq 1</td>
<td>ADC Seq 0</td>
<td>Quad Encoder</td>
</tr>
<tr>
<td>0xE000E414</td>
<td>Timer 2A</td>
<td>Timer 1B</td>
<td>Timer 1A</td>
</tr>
<tr>
<td>0xE000E418</td>
<td>Comp 2</td>
<td>Comp 1</td>
<td>Comp 0</td>
</tr>
<tr>
<td>0xE000ED20</td>
<td>SysTick</td>
<td>PendSV</td>
<td>--</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>UART2</td>
<td>H</td>
<td>NVIC_EN1_R</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>