

EE445M Midterm Study Guide (Spring 2017) (updated February 25, 2017):

Instructions:

- Open book and open notes.
- No calculators or any electronic devices (turn cell phones off).
- Please be sure that your answers to all questions (and all supporting work that is required) are contained in the space (boxes) provided.
- *Anything outside the boxes will be ignored in grading.*
- For all questions, unless otherwise stated, find the most efficient (time, resources) solution.

Lecture notes Lectures 1 through 5 (up through lecture 5 slide 63)

Book Chapters 1-4

1. Computer Architecture 1

1.1. Introduction to Real-Time Operating Systems	1
1.1.1. Real-time operating systems	1
1.1.2. Embedded Systems	3
1.2. Computer Architecture	4
1.2.1. Computers, processors, and microcontrollers	4
1.2.2. Memory	7
1.3. Cortex-M Processor Architecture	9
1.3.1. Registers	9
1.3.2. Stack	11
1.3.3. Operating modes	13
1.3.4. Reset	14
1.3.5. Clock system	14
1.4. Texas Instruments Cortex-M Microcontrollers	15
1.4.1. Introduction to I/O	15
1.4.2. Texas Instruments TM4C123 LaunchPad I/O pins	17
1.4.3. Texas Instruments TM4C1294 Connected LaunchPad I/O pins	21
1.4.4. Texas Instruments MSP432 LaunchPad I/O pins	26
1.4.5. Interfacing to a LaunchPad	29
1.5. ARM Cortex-M Assembly Language	30
1.5.1. Syntax	30
1.5.2. Addressing modes and operands	32
1.5.3. List of twelve instructions	32
1.5.4. Accessing memory	34
1.5.5. Functions	37
1.5.6. ARM Cortex Microcontroller Software Interface Standard	39
1.5.7. Conditional execution	40
1.5.8. Stack usage	42
1.5.9. Floating point math	43
1.5.10. Keil assembler directives	47
1.6. Pointers in C	48
1.6.1. Pointers	48
1.6.2. Arrays	50

1.6.3. Linked lists	52
1.7. Memory Management	54
1.7.1. Use of the heap	54
1.7.2. Simple fixed-size heap	56
1.7.3. Memory manager: malloc and free	57
1.8. Introduction to debugging	61
2. Microcontroller Input/Output	67
2.1. Parallel I/O	67
2.1.1. TM4C I/O programming	68
2.1.2. MSP432 I/O programming	71
2.2. Interrupts	76
2.2.1. NVIC	77
2.2.2. SysTick periodic interrupts	80
2.2.3. Periodic timer interrupts	83
2.2.4. Critical sections	86
2.2.5. Executing periodic tasks	90
2.2.6. Software interrupts	91
2.3. First in First Out (FIFO) Queues	92
2.4. Edge-triggered Interrupts	94
2.4.1. Edge-triggered interrupts on the TM4C123	94
2.4.2. Edge-triggered Interrupts on the MSP432	98
2.5. UART Interface	101
2.5.1. Transmitting in asynchronous mode	103
2.5.2. Receiving in asynchronous mode	104
2.5.3. Interrupt-driven UART on the TM4C123	105
2.5.4. Interrupt-driven UART on the MSP432	110
2.6. Synchronous Transmission and Receiving using the SSI	112
2.7. Input Capture or Input Edge Time Mode	114
2.7.1. Basic principles	114
2.7.2. Period measurement on the TM4C123	116
2.7.3. Period measurement on the MSP432	120
2.7.4. Pulse width measurement	125
2.7.5. Ultrasonic distance measurement	125
2.8. Pulse Width Modulation	125
2.8.1. Pulse width modulation on the TM4C123	126
2.8.2. Pulse width modulation on the MSP432	127
2.9. Analog Output	130
2.10. Analog Input	134
2.10.1. ADC Parameters	134
2.10.2. Internal ADC on TM4C	135
2.10.3. Internal ADC on MSP432	139
2.10.4. IR distance measurement	143
2.11. OS Considerations for I/O Devices	144
2.11.1 Board Support Package	144
2.11.2 Path Expression	146

2.12. Debugging	148
2.12.1. Functional Debugging	148
2.12.2. Performance Debugging (FFT analysis)	150
2.12.3. Debugging heartbeat	153
2.12.4. Profiling	154
3. Thread Management	159
3.1. Introduction to RTOS	159
3.1.1. Motivation	159
3.1.2. Parallel, distributed and concurrent programming	160
3.1.3. Introduction to threads	162
3.1.4. States of a main thread	163
3.1.5. Real-time systems	164
3.1.6. Producer/Consumer problem using a mailbox	168
3.1.7. Scheduler	169
3.2. Function pointers	172
3.3. Thread Management	174
3.3.1. Two types of threads	174
3.3.2. Thread Control Block (TCB)	175
3.3.3. Creation of threads	177
3.3.4. Launching the OS	178
3.3.5. Switching threads	179
3.3.6. Profiling the OS	181
3.3.7. Linking assembly to C	182
3.3.8. Periodic tasks	183
3.4. Semaphores	185
3.5. Thread Synchronization	188
3.5.1. Resource sharing, nonreentrant code or mutual exclusion	188
3.5.2. Condition variable	189
3.5.3. Thread communication between two threads using a mailbox	190
3.6. Process Management	192
3.7. Dynamic loading and linking	193
4. Time Management	197
4.1. Cooperation	197
4.1.1. Spin-lock semaphore implementation with cooperation	197
4.1.2. Cooperative Scheduler	199
4.2. Blocking semaphores	200
4.2.1. The need for blocking	200
4.2.2. The blocked state	201
4.2.3. Implementation	203
4.2.4. Thread rendezvous	205
4.3. First In First Out Queue	206
4.3.1. Producer/Consumer problem using a FIFO	206
4.3.2. Little's Theorem	207
4.3.3. FIFO implementation	208

4.3.4. Three-semaphore FIFO implementation 209
 4.3.5. Two-semaphore FIFO implementation 212
 4.3.6. One-semaphore FIFO implementation 214
~~4.3.7. Kahn Process Networks 215~~
 4.4. Thread sleeping 217
 4.5. Deadlocks 218
 4.6. Monitors 220
 4.7. Fixed Scheduling 222

Week	Notes	Topics
1 1/18	Lecture 1	Introduction - ARM architecture, instruction set, stack, µVision4 compiler, GPIO, timer, UART, device drivers
2 1/25	Lecture 2	Software design - Modular programming, call & data flow graphs, flowcharts, I/O synchronization Debugging - Lab environment, intrusiveness, monitor, output to scope, simulator
3 2/1	Lecture 3	RTOS - Multi-threading/-tasking, OS architecture OS kernel - Interrupt servicing, operating modes, context switching
4 2/8	Lecture 4	Threads - TCB, cooperative & preemptive multitasking, round-robin scheduler Thread communication & synchronization - Critical sections, reentrance, FIFO, mailbox
5 2/15	Lecture 5	Semaphores - Spinlock & blocking semaphores, monitors, deadlock Debugging - Testing, performance measures (response time, jitter, throughput)

Lab Important Topics

Lab 1: Interrupts, Cortex M4 architecture, FIFO queues, UART, ADC

Lab 2: Real time OS, semaphores, critical sections, synchronization, communication

Lab 3: Debugging, blocking semaphores

Architecture (Chapters 1, 2 are a review of EE445L)

Registers, buses, ports, stack,
 Interrupts NVIC, tail chain, late arriving,
 SysTick, PendSV, edge triggered interrupts,
 arm, enable, latency, jitter, hardware FIFO,
 ARM assembly code,
 subroutine linkage (AAPCS),
 parameter passing (registers and stack),
 local variables (registers and stack),
 interrupt linkage.
 Data flow graph, call graph, flowcharts.
 HAL, device driver.

Debugging

- intrusiveness,
- profile,
- dump,
- control, observability,
- coverage
- white box, black box

Data structures

- FIFO,
- statically allocated linked lists,
- dynamically allocated linked lists,

OS stuff

- latency, real- time, interrupt priority,
- reentrancy, critical sections, race condition,
- sleeping,
- scheduling

 - preemptive vs nonpreemptive=cooperative,

 - round robin,

 - ~~priority,~~

 - ~~rate monotonic,~~

 - ~~earliest deadline first,~~

 - ~~least slack time first),~~

- semaphore implementation

 - spinlock,

 - cooperative spinlock,

 - blocking,

- semaphore applications

- OS Concepts

 - kernel,

 - bounded waiting,

 - ~~priority inversion, priority inheritance, aging, starvation,~~

 - mutual exclusion,

 - certification,

 - hooks,

 - slack time,

 - lateness,

 - ~~rate monotonic scheduler, rate monotonic theorem,~~

 - protection,

 - CPU utilization,

 - semaphore application (study the book examples),**

 - deadlocks

 - necessary conditions,

 - detection,

 - prevention,

avoidance,
resource allocation graph, monitors.

See questions at the end of Chapters 3 and 4).

Real time OS, semaphores, critical sections, synchronization, communication

Spring 2001, Quiz2, Question 2, Sleep primitive

Fall 2001, Quiz2, Question 4, Priority scheduler, deadlock

Spring 2002, Quiz1, Question 3, Dynamic thread allocation, thread Kill

Fall 2002, Quiz2, Question 2, application of semaphores

Fall 2002, Final, Question 4, use of semaphores

Fall 2002, Final, Bonus questions 1,2,6, assembly language used in OS programming

Fall 2003, Quiz1, Question 2, use of semaphores

Fall 2003, Quiz1, Question 3, changing the TCB

Fall 2003, Quiz1, Question 4, definition of time jitter

Fall 2003, Quiz1, Question 5, implementation of OS_Wait

Fall 2003, Final, Question 14, definitions of OS concepts/terms

Fall 2004, Quiz2, Question 2, Three thread rendezvous

Fall 2004, Quiz2, Question 3, Binary semaphore

Fall 2004, Final, Question 9, Path expression

Fall 2005, Quiz2, Question 4, Reader/writer problem

Fall 2005, Quiz2, Question 5, Cooperative thread scheduler

Fall 2006, Quiz2, Question 9, Fork

Fall 2006, Quiz2, Question 5, Resource allocation graph

Fall 2006, Final, Question 5, Exponential Queue or multi-level feedback queue scheduling

Spring 2008, Quiz2, Question 4, use of semaphores

Spring 2008, Final, Question 2, Effect of OS on time-jitter while sampling an ADC

Spring 2008, Final, Question 5, Critical section, design new instruction

Spring 2009, Quiz 2, Question 4, Critical section

Spring 2009, Quiz 2, Question 5, Fork and join

Spring 2009, Final, Question 5, kill threads that finish executing

Spring 2010, Quiz 1, Question 2, word bank

Spring 2010, Quiz 1, Question 4, alternate words for signal and wait

Spring 2010, Quiz 1, Question 5, what happens if an ISR calls OS_Wait

Spring 2010, Quiz 1, Question 6, implementing mutual exclusion

Spring 2010, Quiz 1, Question 7, application of semaphores

Spring 2011, Quiz 1, Question 4, definitions

Spring 2011, Quiz 1, Question 5, application of semaphores

Spring 2011, Quiz 1, Question 6, new implementation of semaphores

~~Spring 2011, Quiz 1, Question 7, priority scheduler (the 2011 class did horrible on this question because they parroted their lab solution without reading the question)~~

Spring 2010 Final, Question 5, definitions d, i, j

Spring 2011 Final, Question 8, bounded waiting

Spring 2011 Final, Question 9, real time OS, minimizing latency

Spring 2011 Final, Question 11, FIFO with semaphores

Spring 2011 Final, Question 12, implementing semaphores in a Dual core processor

Spring 2011 Final, Question 16, implementing a thread scheduler on a 16-core processor

Spring 2012 Quiz 1, Question 4, Two SPs.
Spring 2012 Quiz 1, Question 5, OS definitions.
Spring 2012 Quiz 1, Question 7, Monitor and deadlocks.
Spring 2012 Quiz 1, Question 8, OS_AddThread and OS_Kill.
Spring 2012 Quiz 1, Question 9, Use OS to debounce a switch.
~~Spring 2013 Quiz 1, Question 1, Priority.~~
Spring 2013 Quiz 1, Question 3, OS definitions.
Spring 2013 Quiz 1, Question 5, using semaphores.
Spring 2013 Quiz 1, Question 6, Assembly language thread switch.

General questions

Fall 2004, Quiz2, Question 4, Time-jitter
Fall 2004, Quiz2, Question 5, Definitions and a word bank
Fall 2005, Quiz2, Question 6, Time-jitter
Fall 2006, Final, Question 4, Critical section
Spring 2009, Quiz 2, Question 3, FIFO implementation
Spring 2011, Quiz 1, Question 1, time jitter
Spring 2011, Quiz 1, Question 2, reentrant, parameter passing, LR
Spring 2011, Quiz 1, Question 3, bit-banded I/O eliminates critical section, which registers are pushed on the stack during an interrupt context switch, what is LR during an ISR
Spring 2010 Final, Question 1, Cortex M3 interrupt context switch (answer for TM4C123)
Spring 2011 Final, Question 2, Cortex M3 interrupt context switch
Spring 2012 Quiz 1, Question 3, Harvard architecture.
Spring 2012 Quiz 1, Question 6, Reentrancy.
Spring 2013 Quiz 1, Question 2, Control and observability.
Spring 2013 Quiz 1, Question 4, Critical section