

ECE 461/561, Spring 2024: Test 2 Solutions

This quiz is closed-computer, closed-book, closed-notes. You may use two 8.5" x 11" sheets of paper with anything you want written or printed on its two sides. Calculators are allowed. Unless otherwise stated, assume the code is built using MDK-ARM (AC6 compiler, armlink linker, all settings for maximum optimization for time) for the Kinetis KL25Z128 MCU used on the FRDM-KL25Z board and the core clock frequency is fixed at 48 MHz. ECE 461 students may attempt questions marked **ECE 561 Only** for extra credit. **Please read and sign this statement: I have not received assistance from anyone nor assisted others while taking this test. I have also notified the test proctor of any violations of the above conditions.**

Signature _____

Question	561 Only?	Max. Score	Score
1		3	
2		2	
3		1	
4		1	
5		2	
6		1	
7		1	
8		1	
9		1	
10		1	
11		1	
12		1	
13		1	
14	yes	1	
15		4.5	
16		2	

Question	561 Only?	Max. Score	Score
17		2	
18	yes	1	
19		2	
20		2	
21		2	
22		2	
23		1	
24		1	
25		2	
26		1	
27		1.5	
28		1	
29		2	
30		2	
31	yes	2	

Possibly Useful Reference Information

Condition Flag	Meaning if 0	Meaning if 1
Z Zero	Result not zero 0x0000	Result was zero
V Overflow	Result did not overflow	Result overflowed
N Negative	Not negative, MS bit of result is 0	Negative, MS bit of result is 1
C Carry	No carry out or borrow in	Carry out or borrow in occurred

Mnemonic Extension	Meaning	Condition Flags	Mnemonic Extension	Meaning	Condition Flags
EQ	Equal	Z == 1	VC	No overflow	V == 0
NE	Not equal	Z == 0	HI	Unsigned higher	C == 1 and Z == 0
CS	Carry set	C == 1	LS	Unsigned lower or same	C == 0 or Z == 1
CC	Carry clear	C == 0	GE	Signed greater than or equal	N == V
MI	Minus, negative	N == 1	LT	Signed less than	N != V
PL	Plus, positive or zero	N == 0	GT	Signed greater than	Z == 0 and N == V
VS	Overflow	V == 1	LE	Signed less than or equal	Z == 1 or N != V

Power and Energy

Modeling Current vs. Frequency

The datasheet shows the current consumption I_{DD} of the STM32L010C6 (a newer MCU based on the Cortex-M0+ CPU) under various conditions and with supply voltage $V_{DD} = 3.0$ V.

1. Create a linear model of CPU current consumption for Range 2 ($V_{CORE} = 1.5$ V) based on CPU frequency: $I_{DD} = m * f_{HCLK} + b$. The slope m must be in $\mu A/MHz$, and the y-intercept b in μA .

$$m = (2.1 \text{ mA} - 0.56 \text{ mA}) / (16 \text{ MHz} - 4 \text{ MHz}) = 128.33 \text{ uA/MHz}$$

$$b = 46.67 \text{ uA}$$

$$I(f_{HCLK}) = f_{HCLK} * 128.33 \text{ uA/MHz} + 46.67 \text{ uA}$$

Symbol		f_{HCLK} (MHz)	Typ	Max	Unit
I_{DD} (Run from Flash memory)	Range 3, $V_{CORE} = 1.2$ V $VOS[1:0] = 11$	1	140	200	μA
		2	245	310	
		4	460	540	
	Range 2, $V_{CORE} = 1.5$ V $VOS[1:0] = 10$	4	0.56	0.63	mA
		8	1.1	1.2	
		16	2.1	2.3	
	Range 1, $V_{CORE} = 1.8$ V $VOS[1:0] = 01$	8	1.25	1.4	
		16	2.5	2.7	
		32	5	5.6	

Assume that the MCU clock frequency f_{HCLK} is 10 MHz and the supply voltage $V_{DD} = 3.0$ V.

2. What is this MCU's current consumption in μA ?

$$I(f) = f_{HCLK} * 128.33 \text{ uA/MHz} + 46.67 \text{ uA} = 10 \text{ MHz} * 128.33 \text{ uA/MHz} + 46.67 \text{ uA} = 1283.3 \text{ uA} + 46.67 \text{ uA} = 1330 \text{ uA}$$

3. What is this MCU's power consumption in mW?

$$1330 \text{ uA} * 3.0 \text{ V} / (1000 \text{ uW/mW}) = 3.99 \text{ mW}$$

4. What is this MCU's energy use per clock cycle in pJ?

$$3.99 \text{ mW} / 10 \text{ MHz} = 0.399 \text{ nJ}$$

$$0.399 \text{ nJ} * 1000 \text{ pJ/nJ} = 399 \text{ pJ}$$

Frequency Scaling

The KL25Z128 MCU draws $I_{DD} = f_{CLK} * 82 \text{ uA/MHz} + 1856 \text{ uA}$ when supply voltage $V_{DD} = 3.0$ V.

Assume the KL25Z128 MCU clock frequency f_{CLK} is 10 MHz and the supply voltage $V_{DD} = 3.0$ V.

5. What is this MCU's current consumption in μA ?

$$I(f) = f_{CLK} * 82 \text{ uA/MHz} + 1856 \text{ uA} = 10 \text{ MHz} * 82 \text{ uA/MHz} + 1856 \text{ uA} = 820 \text{ uA} + 1856 \text{ uA} = 2676 \text{ uA}$$

6. What is this MCU's power consumption in mW?

$$P = I * V = 2676 \text{ uA} * 3 \text{ V} = 8028 \text{ uW}$$

$$8028 \text{ uW} / (1000 \text{ uW/mW}) = 8.028 \text{ mW}$$

7. What is this MCU's energy use per clock cycle in pJ?

$$8.028 \text{ mW} / 10 \text{ MHz} = 0.8028 \text{ nJ}$$

$$0.8028 \text{ nJ} * 1000 \text{ pJ/nJ} = 802.8 \text{ pJ}$$

Using Sleep Mode

Consider an MCU which runs at 50 MHz with a 3.0 V supply, drawing 1.5 mA. It needs to perform 10,000 cycles of computation every 30 ms. It is normally sleeping, drawing 200 uA. Every 30 ms it wakes up, performs the computations, and goes back to sleep. Assume it takes 250 us to wake up, and draws 1 mA while waking up.

Duty Cycles

8. What is the fraction of time that the MCU is performing computations?

$$(10,000 \text{ cycles}/50 \text{ MHz})/(30 \text{ ms}) = 200 \text{ us}/30 \text{ ms} = 0.00667$$

9. What is the fraction of time that the MCU is sleeping?

$$1 - 0.00667 - 0.00833 = 0.985 \text{ (see \#10 below)}$$

10. What is the fraction of time that the MCU is waking up?

$$250 \text{ us} / 30 \text{ ms} = 0.00833$$

Average Power Consumption

11. What is the average power consumption (in mW) due to the MCU performing computations?

$$0.00667 * 1.5 \text{ mA} * 3 \text{ V} = 0.030 \text{ mW}$$

12. What is the average power consumption (in mW) due to the MCU sleeping?

$$0.985 * 0.200 \text{ mA} * 3 \text{ V} = 0.591 \text{ mW}$$

13. What is the average power consumption (in mW) due to the MCU waking up?

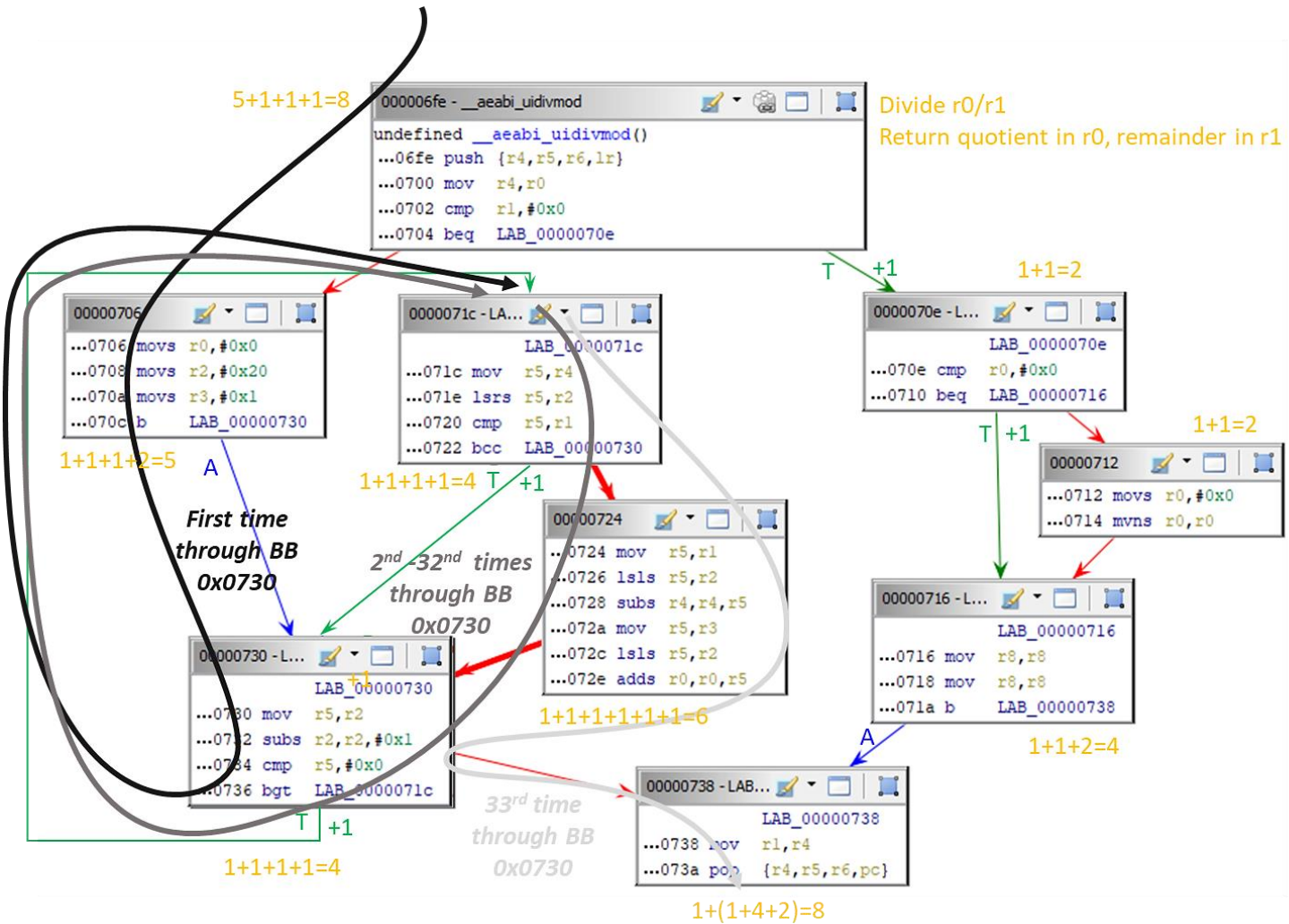
$$0.00833 * 1.0 \text{ mA} * 3 \text{ V} = 0.025 \text{ mW}$$

14. **ECE 561 Only:** What is the total average power consumption (in mW) for the MCU?

$$0.030 \text{ mW} + 0.591 \text{ mW} + 0.025 \text{ mW} = 0.646 \text{ mW}$$

Responsiveness

Static Timing Analysis



Consider the CFG above. Assume the following cycle counts (note the **changes**):

- CPU cycles required for instruction execution:
 - Push: 1 + # of registers pushed
 - Pop: 1 + # of registers popped + 2 (if to PC)
 - Load: 2
 - Store: 1
 - Branch (b): 2
 - **Branch and link (bl): 3**
 - Conditional branch: 2 if taken (T), 1 if not
 - All other instructions: 1

15. Complete this table with the minimum and maximum number of CPU clock cycles to execute each basic block **once**. Hint: a basic block which don't end in a conditional branch will have the same minimum and maximum number of cycles.

Basic Block Start Address	# of CPU Cycles for Executing BB Once	
	Minimum	Maximum
0x06fe	8	9
0x0706	5	5
0x070e	2	3
0x0712	2	2
0x0716	4	4
0x071c	4	5
0x0724	6	6
0x0730	4	5
0x0738	8	8

Consider the execution path which results in the **minimum** number of CPU clock cycles to execute the entire function.

16. List the basic blocks on that path in the order which execute. Identify each basic block by its starting address. If a sequence of blocks X, Y, Z repeats N times, write N*(X, Y, Z).

0x06fe, 0x070e, 0x0716, 0x0738

17. How many CPU clock cycles does that path take?

$8+1+2+1+4+8=24$ cycles

18. **ECE 561 Only:** Consider the loop containing basic blocks 0x071c, 0x0724 and 0x0730. Block 0x0730 uses r2 to decide whether to repeat the loop or not. How many times will the branch from the end of block 0x0730 to the beginning of block 0x071c be taken? Explain why for full credit.

In BB 0x0706: `movs r2,#0x20` puts the value 32 decimal into r2.

In BB 0x0730: `mov r5, r2` will copy r2 (initially 32) to r5

`subs r2,r2,#0x1` decrements r2 by 1

`cmp r5,#0x0` compares r5 (the old value of r2, not the just decremented version) with zero.

The conditional branch `bgt` will be taken if r5 is greater than 0

In BB 0x071c: r2 is used not changed.

Assume `bcc` is not taken, leading to worst-case execution path.

In BB 0x0724: r2 is used but not changed

The first time BB 0x0730 executes, r5 gets 32 from r2, r2 goes from 32 to 31, and r5 is 32. $32 > 0$, so the `bgt` branch to 0x071c will be taken.

The second time, r5 gets 31, r2 goes to 30, and r5 is 31. $31 > 0$, so the branch is taken.

...

The 32nd time, r5 gets 1, r2 goes to 0, and r5 is 1. $1 > 0$, the branch is taken..

The 33rd time, r5 gets 0, r2 goes to 0xffffffff, and r5 is 0. $0 !> 0$, so the branch is not taken and the loop falls through to BB 0x0738.

The `bgt` branch is executed 33 times but will be taken only the first 32 times.

Consider the execution path which results in the **maximum** number of CPU clock cycles to execute the entire function.

19. List the basic blocks on that path in the order which execute. Identify each basic block by its starting address. If a sequence of blocks X, Y, Z repeats N times, write $N*(X, Y, Z)$. If you did not attempt question 18, assume that branch will be taken 10 times.

0x06fe, 0x0706, $32*(0x0730, 0x071c, 0x0724), 0x0730, 0x0738$

If assuming bgt taken 10 times:

0x06fe, 0x0706, $10*(0x0730, 0x071c, 0x0724), 0x0730, 0x0738$

20. How many CPU clock cycles does that path take?

0x06fe,	0x0706,	$32*(0x0730,$	$0x071c,$	$0x0724),$	$0x0730,$	$0x0738$	
8 +	5 +	$32*(4+1+$	$4 +$	$6) +$	$4 +$	$+8$	$= 505$

If bgt taken 10 times:

0x06fe,	0x0706,	$10*(0x0730,$	$0x071c,$	$0x0724),$	$0x0730,$	$0x0738$	
8 +	5 +	$10*(4+1+$	$4 +$	$6) +$	$4 +$	$+8$	$= 175$

Timing Analysis Methods

21. Does static timing analysis (based on correct assumptions) provide a safe or unsafe timing bound? Explain why.

Assuming all the assumptions made are correct, static timing analysis will provide a safe timing bound. There may be some cases where it can't rule out possible paths or limit loop iteration counts, so it must assume the worst-case (maximum) values there. This will be safe, but an overestimate.

22. Does experimental timing analysis provide a safe or unsafe timing bound? Explain why.

Experimental timing analysis usually provides an unsafe timing bound because you cannot guarantee that the test data triggered the worst-case behavior. In trivially simple cases, it may provide a safe timing bound.

23. How can you improve the accuracy of static timing analysis?

Use better analysis which can rule out more possible code paths, loop counts, input data, timing of events, etc.

24. How can you improve the accuracy of experimental timing analysis?

Run better tests which trigger the execution closer to the worst case.

Response Time and Schedulability

Consider a real-time system consisting of the following set of independent periodic tasks and the RTX5 RTOS configured for preemptive scheduling. Each task has the range of possible execution times and periods shown.

Task	Task Execution Time C (ms)		Task Period T (ms)	
	Minimum	Maximum	Minimum	Maximum
Fee	0.45	0.9	4	4
Fi	3.5	7	21	23.1
Fo	0.16	0.2	1.2	2.2
Fum	3	4	100	110

25. Complete the table below by selecting the values of C and T which will lead to the **worst-case** response time .

Task	Task Execution Time C (ms)	Task Period T (ms)
Fee	0.9	4
Fi	7	21
Fo	0.2	1.2
Fum	4	100

26. Calculate the **maximum** CPU utilization of the task set with these times and periods.

$$U = 0.9/4 + 7/21 + 0.2/1.2 + 4/100 = 0.765 = 76.5\%$$

27. Assume that each task has a deadline equal to its worst-case period. Does the utilization bound test show that this task set is **always schedulable** using **Earliest-Deadline First dynamic-priority preemptive** scheduling? Why or why not?

Yes, the UB test shows this task is always schedulable with EDF DPP, because the task set's utilization of 76.6% is less than the EDF DPP utilization bound of 100%.

28. Use the rate-monotonic approach to assign a fixed priority (high (A) to low (D)) to each task based on the worst-case conditions determined above.

Task	Fee	Fi	Fo	Fum
Priority	B	C	A	D

29. Assume that each task has a deadline equal to its worst-case period. Does the utilization bound test show that this task set is **always schedulable** using **fixed-priority preemptive** scheduling with these priorities? Why or why not?

The RM FPP utilization bound for a task set with four tasks is $U_{\max}(4) = 4(2^{1/4} - 1) = 0.7568 = 75.68\%$.

No, the test does not show the task set is always schedulable with RM FPP, because the task set's utilization of 76.6% is less than the RM FPP utilization bound of 75.68%.

30. Find the worst-case response time of the **lowest** priority task when using **preemptive** fixed-priority scheduling and these priorities.

$$R_i^0 = C_i + \sum_{j \in hp(i)} C_j = C_{Fum} + C_{Fo} + C_{Fee} + C_{Fi} = 4 + 0.2 + 0.9 + 7 = 12.1 \text{ ms}$$

$$R_i^1 = C_i + \sum_{j \in hp(i)} \left\lceil \frac{R_i^0}{T_j} \right\rceil C_j = 4 + \left\lceil \frac{12.1}{1.2} \right\rceil * 0.2 + \left\lceil \frac{12.1}{4} \right\rceil * 0.9 + \left\lceil \frac{12.1}{21} \right\rceil * 7 = 4 + 11 * 0.2 + 4 * 0.9 + 1 * 7 = 16.8 \text{ ms}$$

$$R_i^2 = C_i + \sum_{j \in hp(i)} \left\lceil \frac{R_i^1}{T_j} \right\rceil C_j = 4 + \left\lceil \frac{16.8}{1.2} \right\rceil * 0.2 + \left\lceil \frac{16.8}{4} \right\rceil * 0.9 + \left\lceil \frac{16.8}{21} \right\rceil * 7 = 4 + 14 * 0.2 + 5 * 0.9 + 1 * 7 = 18.3 \text{ ms}$$

$$R_i^3 = C_i + \sum_{j \in hp(i)} \left\lceil \frac{R_i^2}{T_j} \right\rceil C_j = 4 + \left\lceil \frac{18.3}{1.2} \right\rceil * 0.2 + \left\lceil \frac{18.3}{4} \right\rceil * 0.9 + \left\lceil \frac{18.3}{21} \right\rceil * 7 = 4 + 16 * 0.2 + 5 * 0.9 + 1 * 7 = 18.7 \text{ ms}$$

$$R_i^4 = C_i + \sum_{j \in hp(i)} \left\lceil \frac{R_i^3}{T_j} \right\rceil C_j = 4 + \left\lceil \frac{18.7}{1.2} \right\rceil * 0.2 + \left\lceil \frac{18.7}{4} \right\rceil * 0.9 + \left\lceil \frac{18.7}{21} \right\rceil * 7 = 4 + 16 * 0.2 + 5 * 0.9 + 1 * 7 = 18.7 \text{ ms}$$

The worst-case response time for Fum is 18.7 ms.

31. **ECE 561 Only:** Find the worst-case response time of the **second-lowest** priority task when using **preemptive** fixed-priority scheduling and these priorities.

Second-lowest priority task is Fi. Fo and Fee are higher priority.

$$R_i^0 = C_i + \sum_{j \in hp(i)} C_j = C_{Fi} + C_{Fo} + C_{Fee} = 7 + 0.2 + 0.9 = 8.1 \text{ ms}$$

$$R_i^1 = C_i + \sum_{j \in hp(i)} \left\lceil \frac{R_i^0}{T_j} \right\rceil C_j = 7 + \left\lceil \frac{8.1}{1.2} \right\rceil * 0.2 + \left\lceil \frac{8.1}{4} \right\rceil * 0.9 = 7 + 7 * 0.2 + 3 * 0.9 = 11.1 \text{ ms}$$

$$R_i^2 = C_i + \sum_{j \in hp(i)} \left\lceil \frac{R_i^1}{T_j} \right\rceil C_j = 7 + \left\lceil \frac{11.1}{1.2} \right\rceil * 0.2 + \left\lceil \frac{11.1}{4} \right\rceil * 0.9 = 7 + 10 * 0.2 + 3 * 0.9 = 11.7 \text{ ms}$$

$$R_i^3 = C_i + \sum_{j \in hp(i)} \left\lceil \frac{R_i^2}{T_j} \right\rceil C_j = 7 + \left\lceil \frac{11.7}{1.2} \right\rceil * 0.2 + \left\lceil \frac{11.7}{4} \right\rceil * 0.9 = 7 + 10 * 0.2 + 3 * 0.9 = 11.7 \text{ ms}$$

The worst-case response time for Fi is 11.7 ms.