

UNIVERSITY OF SOUTHAMPTON

COMP2323

Computer Systems II Logbook

Semester 2, 2026

Christian Gleitzman



February 6, 2026

Contents

1	Week X: Example Topic	2
1.1	Task Outline	2
1.2	Implementation	2
1.3	Outcomes	3
A	Exam Quick Reference	4
A.1	C Operator Precedence	4
A.2	Bitwise Cheat Sheet	4
A.3	Common Register Maps	4

Week X: Example Topic

1.1 Task Outline

Objective: [e.g., Sampling the potentiometer at 10kHz.]

Design Decision

[Explain your choice here. e.g.:]

- **Selected Mode:** Circular Buffer DMA.
- **Justification:** Polling would consume 100% CPU at this sample rate. DMA allows the CPU to sleep, reducing power.

1.2 Implementation

[Brief note on register setup, e.g., "ADC1 is triggered by TIM3 TRGO event."]

```
1 // src/week-01/example.c
2 #include <stdio.h>
3
4 int main(void) {
5     // Placeholder for Task 1
6     return 0;
7 }
```

Listing 1.1: Register-Level Driver

1.3 Outcomes



Figure 1.1: Logic analyzer trace showing 100us sample interval.

Note for Exam

- **Bug Found:** The ADC value drifted when WiFi was enabled.
- **Fix:** Added a 0.1uF decoupling capacitor to the VREF pin.
- **Constraint:** Maximum sampling frequency is 2.4 MSPS on this bus.

Exam Quick Reference

A.1 C Operator Precedence

↑ higher precedence									
()	[]	.	->						
!	+	-	~	++	--	*	&	(type)	sizeof
*	/	%							
+	-								
<<	>>								
<	>	<=	>=						
==	!=								
&									
^									
&&									
?	:								
=	&=	=	^=	<<=	>>=	+=	-=	*=	/=
%=									
,									
↓ lower precedence									

Figure A.1: C Operator Precedence and Associativity Table. (Note: Unary operators (right-to-left) bind tighter than arithmetic.)

Exam Reminder: Associativity

Remember that assignment operators ($=$, $+=$, $-=$) associate **Right-to-Left**.

- $a = b = c = 0$; is valid and executes as $a = (b = (c = 0))$;
- $a / b / c$ executes as $(a / b) / c$ (Left-to-Right).

A.2 Bitwise Cheat Sheet

- **Set Bit 3:** `REG |= (1 << 3);`
- **Clear Bit 3:** `REG &= ~(1 << 3);`
- **Toggle Bit 3:** `REG ^= (1 << 3);`
- **Check Bit 3:** `if (REG & (1 << 3))`

A.3 Common Register Maps

```
1 #define GPIOA_MODER  (*((volatile uint32_t *) 0x48000000))
2 #define RCC_AHB1ENR  (*((volatile uint32_t *) 0x40023830))
```
