CSC111 Fa2025 Final Exam CR

The final exam is on Canvas using the Lockdown browser.

1 Multiple Choice 1 point

For unsigned subtraction, when is the Carry flag (C) set to 0?

- When no borrow occurs
- When a borrow occurs
- When the result is negative
- When overflow occurs

2 Multiple Choice 1 point

What ARM condition code suffix is used for a branch if the Zero flag is set after a comparison?

- NE
- GT
- EQ
- AL

3 Multiple Choice 1 point

The TST instruction performs which operation?

- R1 + R2
- R1 - R2
- R1 & R2 (bitwise AND)
- R1 ^ R2 (bitwise XOR)

4 Multiple Choice 1 point

What does the ARM instruction TST r1, #1 check?

- If the least significant bit of r1 is set
- If the least significant bit of r1 is cleared
- If the most significant bit of r1 is set
- If the most significant bit of r1 is cleared

5 Multiple Choice 1 point

Which instruction is used to return control to the caller from a subroutine?

- BL LR
- BX LR
- BL PC
- BX PC

6 Multiple Choice 1 point

When the BL (Branch and Link) instruction is executed, what two operations occur?

- PC is set to target address; LR is incremented by 1
- LR holds the target address; PC is incremented by 2
- LR = PC + 4 (return address); PC = target address
- SP is decremented; PC is set to target address

7 Multiple Choice 1 point

In a for loop implementation using 'SUBS r1, r1, #1' followed by 'BNE loop', what does this accomplish?

- Increments counter and loops if zero
- Decrements counter and loops if not zero
- Sets flags without changing register
- Unconditional branch

8 Multiple Choice 1 point

According to the ARM EABI, how is a 64-bit argument (such as a long int) passed to a subroutine?

- In a single 64-bit register
- Split across R0 and R2
- In two consecutive 32-bit registers (e.g., R0:R1 or R2:R3)
- Always passed on the stack

9 Multiple Choice 1 point

Where is a 32-bit return value placed by a subroutine?

- In register R0
- In register R7
- On the stack
- In the link register

10 Multiple Choice 1 point

When you execute PUSH {R2, R1, R3}, in what memory address order are the registers stored in memory, from lowest to highest address?

- R2, R1, R3
- R1, R2, R3
- R3, R2, R1
- No specific order; implementation-dependent

11 Multiple Choice 1 point

On a PUSH operation in Cortex-M (full descending stack), when is SP decremented?

- After storing each register
- Only once at the end
- Before storing the first register
- SP is incremented, not decremented

12 Multiple Choice 1 point

A timer runs from a 72 MHz clock. PSC = 71 and ARR = 999. In PWM Mode 1, CCR = 250. What is the duty cycle?

- 10%
- 25%
- 50%
- 75%

13 Multiple Choice 1 point

What is the effect of the following instruction ADD r4, r4, r4, LSL #2?

- Multiply R4 by 2
- Multiply R4 by 3
- Multiply R4 by 4
- Multiply R4 by 5

14 Multiple Choice 1 point

What is the effect of the following instruction RSB r4, r4, r4, LSL #2?

- Multiply R4 by 2
- Multiply R4 by 3
- Multiply R4 by 4
- Multiply R4 by 5

15 Multiple Choice 1 point

What is the effect of the following instruction: SUB r4, r4, r4?

- Multiply R4 by 2
- Multiply R4 by 3
- Set R4 to 0
- Set R4 to 1

16 Multiple Choice 1 point

What is the effect of the following instruction: ADD r4, r4, r4?

- Multiply R4 by 2
- Multiply R4 by 3
- Set R4 to 0
- Set R4 to 1

17 Multiple Choice 1 point

What is r2's value after executing the following instructions sequentially?

```
MOV r2, #0
MOVW r2, #0x1234
MOVT r2, #0xABCD
```

- 0xABCD1234
- 0x1234ABCD
- 0xABCD0000
- 0x00001234

18 Multiple Choice 1 point

What is r2's value after executing the following instructions sequentially?

```
MOV r2, #0
MOVT r2, #0xABCD
MOVW r2, #0x1234
```

- 0xABCD1234
- 0x1234ABCD
- 0xABCD0000
- 0x00001234

19 Multiple Choice 1 point

In PWM Mode 1 (Up-counting), if CNT < CCR, the output is:

- ◯ Active (High)
- ◯ Inactive (Low)
- ◯ Toggled
- ◯ Undefined

20 Multiple Choice 1 point

To increase the Duty Cycle in PWM Mode 1, you should:

- ◯ Decrease the CCR value.
- ◯ Increase the CCR value.
- ◯ Decrease the Prescaler (PSC).
- ◯ Increase the Prescaler (PSC).

21 Fill in the Blank 5 points

Find the register values (in hex) after finishing execution of the following program (sequentially, not individually).

```
MOV r0, 0x11
MOV r1, r0, LSL#1
MOV r2, r1, LSL#1
MOV r3, r1, LSR #1
ADD r4, r1, r1, LSL#2
```

Output:

r0 = 0x

r1 = 0x

r2 = 0x

r3 = 0x

r4 = 0x

22 Fill in the Blank 5 points

Find the register values after finishing execution of the following program (sequentially, not individually). (Tip: CMP R0, R1 sets the Carry flag based on R0 - R1. Encode a negative number by taking two's complement of the corresponding positive number.)

```
MOV  R0, #0x1234
MOV  R1, #0x1111
ADD  R2, R0, R1
SUB  R3, R0, R1
RSB  R4, R0, R1
CMP  R0, R1
ADC  R5, R0, R1
ADC  R6, R0, #0x11
```

Àfter execution:

R2 = 0x

R3 = 0x

R4 = 0x

R5 = 0x

R6 = 0x

23 Fill in the Blank 5 points

Given R0 = 0x12345678, write the destination register value after executing each instruction (individually, not sequentially). (Tip: ROR #n rotates the value in a register right by an immediate value n)

1) MOV R1, R0, ROR #8

R1 = 0x

2) MOV R2, R0, ROR #16

R2 = 0x

3) MOV R3, R0, ROR #24

R3 = 0x

4) MOV R7, R0, ASR #24

R7 = 0x

5) MOV R8, R0, ASR #31

R8 = 0x

24 Fill in the Blank 10 points

Assume a 4-bit system. For each of the following operations, compute the result and NZCV flags, based on the first row that has been given to you. (If the result is incorrect, write the correct result in parenthesis like (true = 9).)

Operation	Result in binary	Equivalent unsigned arithmetic in decimal	Equivalent signed arithmetic in decimal	NZCV
0100 + 0101	1001	4 + 5 = 9	4 + 5 = –7 (true = 9)	1001
0011 – 0101				
1111 + 1111				
0000 - 1111				
0100 – 1001				
1110 + 0011				

25 Fill in the Blank 6 points

Write a program to swap the contents of 2 registers R0 and R1 using EOR (Exclusive OR) operations. Hint: do it in three steps:
Step1: Compute R0⊕R1
Step 2: Compute (R0⊕R1)⊕R1 = R0
Step 3: Compute (R0⊕R1)⊕R0 = R1

Line1:

Line2:

Line3:

26 Fill in the Blank 15 points

Consider the following C program. Fill in the blanks for the corresponding assembly programs, implementing two versions of the short-circuit behavior of ||, i.e., if the first operand evaluates to true, the second operand is not evaluated at all because the overall result is already determined to be true.

C Program	Assembly Program 1 (using branch statement)	Assembly Program 2 (using conditional execution)
if ((r1 == r3) || (r5 == r6)) r7 = r7 + 10	CMP r1, r3 ; Compare r1 and r3 [ ] ; if r1 == r3, skip checking second condition [ ] ; Compare r5 and r6 [ ] ; if r5 != r6, both tests failed → skip ADDIT: ADD r7, r7, #10 SKIP:	CMP r1, r3 ; Compare r1 and r3 [ ] ; Only compare r5 and r6 if the first compare failed. [ ] ; Perform the add if either the first check returned Equal (and the second was skipped), or the second check ran and returned Equal.

27 Fill in the Blank 6 points

Consider the following C program that swaps the values of two chars. Fill in the blanks for the corresponding assembler program (recall that a string is terminated with a null (#0)).

C Program	Assembly Program

void strcpy (char *dst, char *src) {
  unsigned i;
  for (i = 0; src[i] != '\0'; ++i)
  dst[i] = src[i];
  dst[i] = '\0';
  return dst;
}
	strcpy PROC
	EXPORT strcpy
	
	loop
	  LDRB r2, [r1] ; load byte from src
	  STRB r2, [r0] ; store byte to dst
	[ ]
	[ ]
	[ ]
	  BNE loop
	  BX LR
	ENDP

28 Fill in the Blank 4 points

Suppose a 16-bit timer has the following settings: CPU clock frequency = 360 MHz; Prescaler PSC = 35999; Counting direction: up-counting; Desired timer frequency = 1 KHz.

Calculate Timer clock frequency = _____ KHz and Auto-Reload Register (ARR) = _____.

29 Fill in the Blank 6 points

Suppose a 16-bit timer has the following settings:

1. CPU clock frequency = 400 MHz.
2. The prescaler PSC = 399
3. ARR = 999
4. CCR = 700
5. Counting direction: down-counting
6. Output is set as PWM mode 1

Calculate Timer clock frequency = _____ MHz; Timer Frequency = _____ KHz; PWM duty cycle = _____.

30 Fill in the Blank 18 points

Show all updates to registers as the assembly code shown below runs, assuming **little-endian** ordering. Fill in the tables of final register values and memory contents. Show the NZCV flags after execution, assuming NZCV=0000 initially. (Tips: the MVN instruction inverts all the bits of Operand2; LSLS is the only instruction that sets flags in this program.)

```
LDR  R1, =0x10000010
LDR  R2, [R1, #4]
MVN  R3, R2
MOV  R4, #0x00F0
AND  R5, R3, R4
EOR  R6, R5, R2
STR  R6, [R1, #8]
LSLS R7, R6, #28
```

Initial memory content at 0x10000010 contains the following bytes in increasing memory address order:

0x10000010 FF EF CD AB 00 00 CD AB AA BB CC DD EE FF

Final register values:

R1 = 0x

R2 = 0x

R3 = 0x

R4 = 0x

R5 = 0x

R6 = 0x

R7 = 0x

Final memory contents:

0x10000010 FF EF CD AB 00 00 CD AB EE FF

Final NZCV =