

NAME: _____

SANTA CLARA UNIVERSITY
Department of Computer Engineering

COEN 020

Final Exam (Part 2)

Spring 2017

12. [5 pts] Give a sequence of no more than two instructions that multiplies the value in R0 by 12 without using a multiply instruction of any kind:

ADD R0,R0,R0,LSL 1
LSL R0,R0,2

13. [5 pts] Consider the code shown below that implements $x=y/k$, where k is a constant. What is the value of k?

LDR R0,y LDR R1,=1<<28 SMULL R0,R1,R0,R1 STR R1,x	(a) 4	(b) 16	(c) 28
--	-------	---------------	--------

14. [5 pts] Circle that set of floating-point registers that the APCS says functions must preserve:

S0 through S15	S4 through S8	S16 through S31
----------------	---------------	------------------------

15. [5 pts] Circle the instruction sequence that is written correctly:

VLDR S0,temp CMP S0,0 IT LT MOVLT R0,1 MOVGE R0,0 STR R0,freezing	VLDR S0,temp VCMP.F32 S0,#0.0 IT LT MOVLT R0,1 MOVGE R0,0 STR R0,freezing	VLDR S0,temp VCMP.F32 S0,#0.0 VMRS APSR_nzcv,FPSCR IT LT MOVLT R0,1 MOVGE R0,0 STR R0,freezing
--	--	---

16. [5 pts] Circle the instruction sequence that is written correctly:

LDR R0,sum LDR R1,count VMOV S0,R0 VMOV S1,R1 VCVT.F32.S32 S0,S0 VCVT.F32.S32 S1,S1 VDIV.F32 S0,S0,S1 VSTR S0,average	LDR R0,sum LDR R1,count VCVT.F32.S32 S0,R0 VCVT.F32.S32 S1,R1 VDIV.F32 S0,S0,S1 ADR R0,average VSTR S0,[R0]	LDR R0,sum LDR R1,count VMOV S0,R0 VMOV S1,R1 VCVT.F32.S32 S0,S0 VCVT.F32.S32 S1,S1 VDIV.F32 S0,S0,S1 ADR R0,average VSTR S0,[R0]
--	---	--

NAME: _____

SANTA CLARA UNIVERSITY
Department of Computer Engineering

COEN 020

Final Exam (Part 2)

Spring 2017

17. [5 pts] Circle the correct Q4.4 representation of -1.25:

1111.0100 ₂	1110.1100 ₂	1001.0100 ₂
------------------------	------------------------	------------------------

18. [5 pts] Circle the arithmetic operation that is written correctly where a, b and c are all Q32 fixed-point reals:

a += 1 ; // increment a	a = b * c ; // a = b × c	a = b << 4 ; // a = 16xb
-------------------------	--------------------------	--------------------------

19. [5 pts] Circle the value that would be printed by the code shown below:

<pre>typedef int64_t Q32 ; Q32 pi = 3.14159 ; printf("%d", pi) ;</pre>	<p>a. 0</p> <p>b. 3</p> <p>c. 3.14159</p>
--	---

20. [2 pts ea] Consider the following assembler template. For each of the output operands a, b and c, place an "X" in one column to indicate the appropriate constraint string:

```
"ADD %[b], %[a], 1 \n\t"  
"MOV %[c], 1000 \n\t"  
"MUL %[a], %[a], %[c] "
```

	"=r"	"+r"	"=&r"
[a]		X	
[b]			X
[c]			X

21. [4 pts] Circle the I/O programming technique that is the simplest to program:

(a) Blocking I/O	(c) Interrupt Driven
(b) Programmed Waiting Loop	(d) Direct Memory Access

NAME: _____

SANTA CLARA UNIVERSITY
Department of Computer Engineering

COEN 020

Final Exam (Part 2)

Spring 2017

22. [3 pts] Circle the I/O programming technique that has the fastest data rate:

(a) Blocking I/O	(c) Interrupt Driven
(b) Programmed Waiting Loop	(d) Direct Memory Access

23. [5 Pts] Determine the amount of representation error that occurs when storing the value $1/3^{\text{rd}}$ using only four fractional bits.

$$1/3 \rightarrow .0101_2 = 5/16$$

$$\text{Error} = \left| \frac{1}{3} - \frac{5}{16} \right| = \left| \frac{16-5x3}{3x16} \right| = \left| \frac{1}{48} \right| = \frac{1}{48}$$

24. [5 pts] The instruction "UMULL R0,R1,R2,R3" produces the unsigned double-length product of R2 and R3. Without using SMULL, give a sequence of ARM instructions that will convert the unsigned product that UMULL leaves in R1.R0 into a signed double-length product.

```
UMULL    R0,R1,R2,R3
CMP      R2,0
IT       LT
SUBLT    R1,R1,R3
CMP      R3,0
IT       LT
SUBLT    R1,R1,R2
// R1.R0 = signed product
```

25. [4 pts] Complete the table at right that defines how binary subtraction works, when computing the difference, $D = X - Y$. Note: X_i , Y_i and D_i each represent a single bit in X , Y and D respectively, B_i is the borrow *into* bit position i and B_{i+1} is the borrow *out* of that bit position.

B_i	X_i	Y_i	B_{i+1}	D_i
0	0	0	0	0
0	0	1	1	1
0	1	0	0	1
0	1	1	0	0
1	0	0	1	1
1	0	1	1	0
1	1	0	0	0
1	1	1	1	1