

### CE-1921 – Dr. Durant – Quiz 3

#### Spring 2020, Week 4

Open: book, notes, assembler/disassembler, previous quizzes, Internet. But, do not discuss the quiz with anyone except the professor until after everybody's work is submitted and the due date has passed. Please submit in Teams Assignments.

I will accept your answers in any well-organized format, but it will probably be easiest if you either:

- Print this quiz, complete it by hand, and scan your result.
- Convert this quiz to PDF and use a tablet-based PDF editor to draw your answers.

1. (5 points) Assemble the following instruction to ARMv4 machine code: **orr1t r9,r2,#0x500**

- Label and box in each field above the boxes below. "cond" has been done for you.
- Box in and fill each field in the next row with the value for that field.
- Convert values to binary in the following row.
- Convert values to hexadecimal in the final row.

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
a	cond				001 opcode				5 Rn				Rd				rot				Imm8											
b	LT= B				DP-Imm 001				1000 0				2				9				12				5							
c	1	0	1	1	0	0	1	1	1	0	0	0	0	0	1	0	0	0	1	1	1	0	0	0	0	0	0	0	0	1	0	1
d	B				3				8				2				9				C				0				5			

$$0x500 = 0x5 \text{ rot } 8 = 0x5 \text{ rot } 24 \therefore \text{rot} = \frac{24}{2} = 12$$

2. (5 points) Show how to disassemble the machine code instruction from last week's quiz:

**0xE586\_5040.**

- Write the machine code in both hex and binary in the given spaces.
- Complete the instruction format template (use the 2-page PDF reference card) based on bits 27...25. Do not put values here, just field names, such as Rn and Rd.
- Draw vertical lines separating the fields through rows b and c. Fill in the values of each field in row c; use a mix of binary, decimal, and hex as appropriate
- Interpret any fields (e.g., PUBWL) that are not obvious (e.g., register numbers are obvious)
- Write the resulting assembly language instruction.

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
a	E				5				8				6				5				0				4				0				
	1	1	1	0	0	1	0	1	1	0	0	0	0	1	1	0	0	1	0	1	0	0	0	0	0	0	1	0	0	0	0	0	0
b	cond				0	1	I	PUBWL				Rn				Rd				Imm12													
c	always/E					I	M	11000				6				5				0x040													

$\bar{I}=0$

① LB=00 → STR  
 PW=10 → offset ([base, offset] asm syntax)  
 U=1 → add offset to base (imm in asm is +, not -)  
 $\bar{I}=0 \therefore$  immediate offset (not reg. offset)

=64 in decimal

② str r5, [r6, #0x40]