Southern University of Science and Technology

2017-2018 Academic Year
Fall Semester
Final Exam Paper

Course Name：CS301   Dept.：CSE   Exam Duration：2 Hours

Question No.	1	2	3	4	5			
Score	20	25	25	20	10			

This exam paper contains 5 questions and the score is 100 in total. (Please hand in your exam paper, answer sheet, and your scrap paper to the proctor when the exam ends.)

INSTRUCTIONS TO CANDIDATES

1、 The number in the column on the right indicates the approximate marks for each section.

2、 Answer should be written in the answer booklet(s) provided.

3、 Answer all questions.

4、 It is not allowed to use calculators in the exam.

Southern University of Science and Technology

Q1

a) What is cache memory? 5

b) If cache access time t_c is 10 nanoseconds, main memory access time t_m is 70 nanoseconds, the extra delay caused by cache control and routing circuits is 5 nanoseconds, and HIT RATIO h = 0.90, please calculate mean access time t_{ave}. 5

c) The operation of a cache relies on two features of microprocessor programs, that is, temporal locality and spatial locality. Briefly describe these two features. 5

Y=32

d) If for an ARM7 microcontroller with the associative cache, the number of bytes in a block is 32, how many bits does the block number have and what is the cache line length? 5

X=32， a=5

Q2

a) List and briefly describe three addressing modes in an ARM7 microcontroller. **5**

b) Registers r1, r3 and r15 hold the values 0x5E8B6A89, 0x14F6EC43, and 0x00008000 respectively, What values are held in r1, r3, r5, r8, r9, r10, r11, and r15 after the execution of the following instructions? **8**

```
ADD r5, r3, r1,
ADD r8, r1, r3,
SUB r9, r1, r3
AND r10, r1, r3
ORR r11, r1, #0xFF0
```

c) What values are held in r5, r9 and r10 and how is memory modified after the execution of the following instructions (assume big endian)? **7**

```
MOV r5, #0x8000
LDR r9, [r5], #4
STR r9, [r5], #4
LDR r10, [r5, #-4]!
```

Memory Address	Contents
0x00008000	0xDC
0x00008001	0x86
0x00008002	0x54
0x00008003	0x97

d) For an ARM7 microcontroller, write an assembly language program to add twenty 32-bit signed integers stored consecutively in memory starting from address 0x00004000, and store the result back to the memory at address 0x00005000. **5**

Q3

a) Briefly describe one type of Static RAM (SRAM) and one type of Dynamic RAM (DRAM); please briefly state, at least, one difference between SRAM and DRAM. **5**

b) For a memory mapped input/output port in an ARM7 microcontroller, brief describe how to input/output data from the port. **5**

c) Draw a memory map for the following memory system using a 32 bit address bus: a 512 KB Non-volatile memory at the lowest possible address, a 32 KB RAM memory starting from address 0x4000 0000, and a 256 MB ROM at the highest possible address. **8**

d) The lowest N address lines are used to connect to the 32 KB memory. For an 8-bit data bus, what is the number N and what are the other address lines used for? **7**

Q4

a) A simple sequential multiplier uses adders, double length registers, and a shifter. It performs one step of the process at a time according to the following procedure: **10**

1) Set a total to zero.
2) Examine the least significant bit (LSB) of the multiplier, if it is 1 add the multiplicand to the total otherwise do nothing.
3) Shift the multiplicand one place left moving in 0 as LSB.
4) Shift the multiplier one place right discarding the old LSB and moving in zero at the left side.
5) Repeat from step 2) until all bits of the multiplier are tested (alternatively end when multiplier is zero).

By tabulating each cycle, show the multiplication calculation process of two numbers 0xA5 and 0x64 according to the procedure presented above. The multiplier 0x64 and the multiplicand 0xA5 are 8-bit number and the results will be 16-bit number.

b) The ARM multiplier employs a modified Booth's algorithm to produce the 2-bit product. This allows all four values of the 2-bit multiplier to be implemented by a simple shift and add or subtract. The control settings for the Nth cycle of the multiplication are shown in Table 1 below **10**

Carry-in	**Multiplier**	**Shift**	**ALU**	**Carry-out**
0	x 0	LSL #2N	A + 0	0
	x 1	LSL #2N	A + B	0
	x 2	LSL #(2N + 1)	A – B	1
	x 3	LSL #2N	A – B	1
1	x 0	LSL #2N	A + B	0
	x 1	LSL #(2N + 1)	A + B	0
	x 2	LSL #2N	A – B	1
	x 3	LSL #2N	A + 0	1

Table 1

By tabulating each cycle, show the multiplication calculation process of two numbers 0xA5 and 0x64 according to the process presented in Table 1. The multiplier 0x64 and the multiplicand 0xA5 are 8-bit number and the results will be 16-bit number.

Q5

According to the Data processing instruction binary encoding shown in Figure 1, what is machine code of the instruction ADDNE r5, r4, #14? (Please refer to the Appendix B for the opcodes). **10**

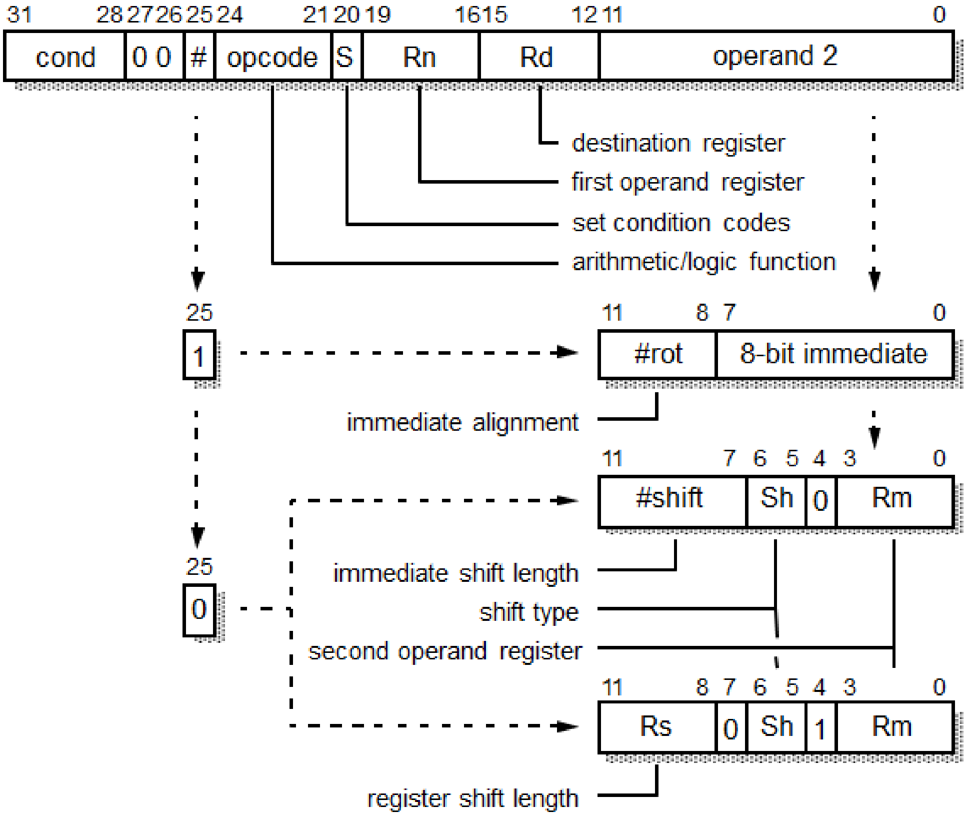


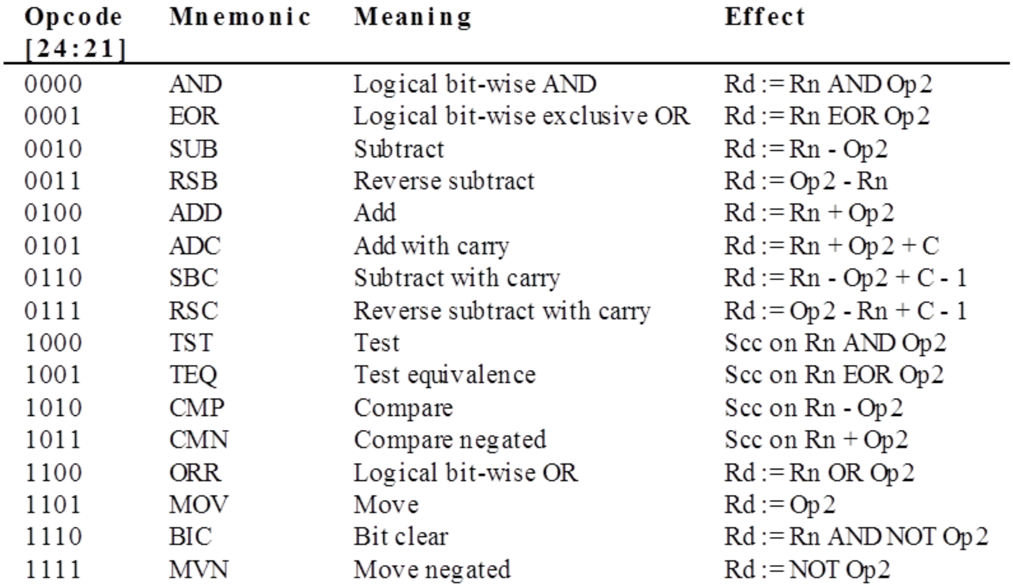
Figure 1. Data processing instruction binary encoding

Southern University of Science and Technology

Appendix A ARM condition codes

Opcode [31:28]	Mnemonic extension	Interpretation	Status flag state for execution
0000	EQ	Equal / equals zero	Z set
0001	NE	Not equal	Z clear
0010	CS/HS	Carry set / unsigned higher or same	C set
0011	CC/LO	Carry clear / unsigned lower	C clear
0100	MI	Minus / negative	N set
0101	PL	Plus / positive or zero	N clear
0110	VS	Overflow	V set
0111	VC	No overflow	V clear
1000	HI	Unsigned higher	C set and Z clear
1001	LS	Unsigned lower or same	C clear or Z set
1010	GE	Signed greater than or equal	N equals V
1011	LT	Signed less than	N is not equal to V
1100	GT	Signed greater than	Z clear and N equals V
1101	LE	Signed less than or equal	Z set or N is not equal to V
1110	AL	Always	any
1111	NV	Never (do not use!)	none

Appendix B ARM data processing instructions



Opcode [24:21]	Mnemonic	Meaning	Effect
0000	AND	Logical bit-wise AND	Rd := Rn AND Op2
0001	EOR	Logical bit-wise exclusive OR	Rd := Rn EOR Op2
0010	SUB	Subtract	Rd := Rn - Op2
0011	RSB	Reverse subtract	Rd := Op2 - Rn
0100	ADD	Add	Rd := Rn + Op2
0101	ADC	Add with carry	Rd := Rn + Op2 + C
0110	SBC	Subtract with carry	Rd := Rn - Op2 + C - 1
0111	RSC	Reverse subtract with carry	Rd := Op2 - Rn + C - 1
1000	TST	Test	Scc on Rn AND Op2
1001	TEQ	Test equivalence	Scc on Rn EOR Op2
1010	CMP	Compare	Scc on Rn - Op2
1011	CMN	Compare negated	Scc on Rn + Op2
1100	ORR	Logical bit-wise OR	Rd := Rn OR Op2
1101	MOV	Move	Rd := Op2
1110	BIC	Bit clear	Rd := Rn AND NOT Op2
1111	MVN	Move negated	Rd := NOT Op2