

PAPER CODE	EXAMINER	DEPARTMENT	TEL
CPT210	Jianjun Chen	Computer Science and Software Engineering	81889137

**2nd SEMESTER 2021/22 RESIT EXAMINATION**

**Undergraduate – Year 3**

**Microprocessor Systems**

**TIME ALLOWED: 2 Hours**

**SUBMISSION TIME ALLOWED: 30 Minutes**

---

**INSTRUCTIONS TO CANDIDATES**

- 1、 This is an open-book examination.**
- 2、 Total marks available are 100.**
- 3、 Answer all questions.**
- 4、 Only English solutions are accepted.**
- 5、 All answers should be submitted to the LearningMall at the end of the exam**

**Question A (20 Marks)**

1. Convert the following decimal numbers into their IEEE-754 single-precision (32-bit) representations. Give your answers in hexadecimal form. All intermediate steps must be clearly shown. A single result will not receive marks. (4 marks each)
  - a. 655.34375
  - b. -1.875
  - c. 0.734375
2. Convert the following IEEE 754 single-precision numbers in hexadecimal into their decimal values accurate to 7 significant figures. All intermediate steps must be clearly shown. A single result will not receive marks. (4 marks each)
  - a. 0x3d4c0000
  - b. 0x40bc0000

**Question B (20 Marks)**

1. Computer memory can be organized as either 'big endian' or 'little endian'. Explain the difference between these two options. (4 marks)
2. The following ARM instructions would store the contents of register r9 in a memory location with address given by the value in the base register, r5, plus any offset.

```
MOV r9, #0x1A2B3C4D      ; let's assume that R9 has this value (doesn't work)
MOV r5, #0x08000000
STR r9, [r5]              ; store full word, no update of r5
STRH r9, [r5, #4]         ; store half word, no update of r5
STRB r9, [r5, #6]         ; store byte, no update of r5
```

Determine the contents of the memory locations with addresses from 0x08000000 to 0x08000006 after the code above has been executed for both big endian and little endian. (12 marks)

3. Explain why there are many options (big/little endian, word/half word/byte, pre/post indexing) provided for load and store instructions? (4 marks)

## Question C (30 Marks)

1. Write an ARM assembly language program to modify the ASCII characters stored in “my\_chars”, so that all lower case letters are changed into upper case and all digits are increased by one. If a digit is 9, it should be changed to 0 instead. Assume that 0x00 marks the end of “my\_chars”. In the example below, the characters are “CPT210isHard” and the expected result should be “CPT321ISHARD”

my\_chars      DCD      0x43505432, 0x31306973, 0x48617264, 0x21000000

### Decimal - Binary - Octal - Hex – ASCII Conversion Chart

Decimal	Binary	Octal	Hex	ASCII	Decimal	Binary	Octal	Hex	ASCII	Decimal	Binary	Octal	Hex	ASCII
32	00100000	040	20	SP	64	01000000	100	40	@	96	01100000	140	60	`
33	00100001	041	21	!	65	01000001	101	41	A	97	01100001	141	61	a
34	00100010	042	22	"	66	01000010	102	42	B	98	01100010	142	62	b
35	00100011	043	23	#	67	01000011	103	43	C	99	01100011	143	63	c
36	00100100	044	24	\$	68	01000100	104	44	D	100	01100100	144	64	d
37	00100101	045	25	%	69	01000101	105	45	E	101	01100101	145	65	e
38	00100110	046	26	&	70	01000110	106	46	F	102	01100110	146	66	f
39	00100111	047	27	'	71	01000111	107	47	G	103	01100111	147	67	g
40	00101000	050	28	(	72	01001000	110	48	H	104	01101000	150	68	h
41	00101001	051	29	)	73	01001001	111	49	I	105	01101001	151	69	i
42	00101010	052	2A	*	74	01001010	112	4A	J	106	01101010	152	6A	j
43	00101011	053	2B	+	75	01001011	113	4B	K	107	01101011	153	6B	k
44	00101100	054	2C	,	76	01001100	114	4C	L	108	01101100	154	6C	l
45	00101101	055	2D	-	77	01001101	115	4D	M	109	01101101	155	6D	m
46	00101110	056	2E	.	78	01001110	116	4E	N	110	01101110	156	6E	n
47	00101111	057	2F	/	79	01001111	117	4F	O	111	01101111	157	6F	o
48	00110000	060	30	0	80	01010000	120	50	P	112	01110000	160	70	p
49	00110001	061	31	1	81	01010001	121	51	Q	113	01110001	161	71	q
50	00110010	062	32	2	82	01010010	122	52	R	114	01110010	162	72	r
51	00110011	063	33	3	83	01010011	123	53	S	115	01110011	163	73	s
52	00110100	064	34	4	84	01010100	124	54	T	116	01110100	164	74	t
53	00110101	065	35	5	85	01010101	125	55	U	117	01110101	165	75	u
54	00110110	066	36	6	86	01010110	126	56	V	118	01110110	166	76	v
55	00110111	067	37	7	87	01010111	127	57	W	119	01110111	167	77	w
56	00111000	070	38	8	88	01011000	130	58	X	120	01111000	170	78	x
57	00111001	071	39	9	89	01011001	131	59	Y	121	01111001	171	79	y
58	00111010	072	3A	:	90	01011010	132	5A	Z	122	01111010	172	7A	z
59	00111011	073	3B	;	91	01011011	133	5B	[	123	01111011	173	7B	{
60	00111100	074	3C	<	92	01011100	134	5C	\	124	01111100	174	7C	
61	00111101	075	3D	=	93	01011101	135	5D	]	125	01111101	175	7D	}
62	00111110	076	3E	>	94	01011110	136	5E	^	126	01111110	176	7E	~
63	00111111	077	3F	?	95	01011111	137	5F	_	127	01111111	177	7F	DEL

License: To view a copy of this license, visit <http://creativecommons.org/licenses/by-sa/3.0/>

ASCII Conversion Chart.doc Copyright © 2008, 2012 Donald Weiman 22 March 2012

**Question D (30 marks)**

1. Given a piece of ARM assembly code shown below and a three-stage pipeline. Describe the running of this program by drawing the pipeline diagram of the whole process. Please clearly indicate the status of each clock circle. (16 marks)

	mov	r1, #0
	mov	r2, #2
L1	add	r1, r1, #1
	add	r3, r3, r1
	cmp	r1, r2
	blt	L1
	mov	r8, r3

2. In ARM7, explain with the aid of a diagram what happens in the instruction pipeline when either a register load or a register store instruction is executed. (4 marks)
3. How are bus conflicts between instruction fetches and data load/store eliminated in the ARM9 microprocessor and what effects does this have on the overall performance of the ARM9 compared to the ARM7? (10 marks)

END OF EXAM PAPER