

**NANYANG TECHNOLOGICAL UNIVERSITY****SEMESTER 2 EXAMINATION 2013-2014****CE1006/CZ1006 – COMPUTER ORGANIZATION AND ARCHITECTURE**

April/May 2014

Time Allowed: 2 hours

**INSTRUCTIONS**

1. This paper contains 4 questions and comprises 7 pages.
2. Answer **ALL** questions.
3. This is a closed-book examination.
4. All questions carry equal marks.
5. The VIP Instruction Set Summary Chart is provided in Appendix 1 on page 7.

1. Figure Q1a shows the hexadecimal contents of several registers in the VIP processor and a section of its memory.

		MEMORY	
Address	Content		
R0	0xFFB	R1	0xFEF
R2	0x101	R3	0x000
AR	0x002	SP	0xFFD
SR	0x000	PC	0x000
Note:		Bit no. 11 . . . . . 4 3 2 1 0 SR [ ] V N Z C Condition Code flags	
		:	:
		0x100	0x000
		0x101	0x777
		0x102	0xE2E
		0x103	0x123
		0x104	0x000
		:	:
		0xFFB	0x81E
		0xFFC	0xFF1
		0xFFD	0x022
		0FFE	0xFFFF
		0FFF	0x000

**Figure Q1a**

Note: Question No. 1 continues on Page 2

- (a) Give (*in hexadecimal*) the 12-bit contents in the two registers **R1** and **SR**, *immediately after* the execution of each instruction given below.

**Note:** Instructions (i) to (vi) are **not consecutive instructions**. You must use the initial conditions shown in Figure Q1a to derive your answer for each of the instructions given below.

- (i) **MOV R1 , [0x104]**
- (ii) **MOV R1 , [R2+2]**
- (iii) **CMP R1 , [SP+0xFFFF]**
- (iv) **ADD R1 , #0x021**
- (v) **RAR R1**
- (vi) **AND R1 , [R0]**

(12 marks)

- (b) With reference to the VIP assembly language program in Figure Q1b, give (*in hexadecimal*) the 12-bit contents of the **seven** registers **R0**, **R1**, **R2**, **R3**, **PC**, **SP** and **SR** *after* the execution of the instruction at the address label **DONE**. Assume that the execution of the given series of consecutive instructions begins at the label **BEGIN**.

<b>BEGIN</b>	<b>MOV</b>	<b>SP , #0xFF2</b>
	<b>MOV</b>	<b>SR , #0</b>
	<b>MOV</b>	<b>R0 , #0</b>
	<b>MOV</b>	<b>R1 , #2</b>
	<b>MOV</b>	<b>R2 , R0</b>
	<b>PSH</b>	<b>R1</b>
	<b>SUB</b>	<b>R2 , R1</b>
	<b>CMP</b>	<b>R0 , R1</b>
	<b>JGT</b>	<b>NEXT</b>
	<b>DEC</b>	<b>R0</b>
<b>NEXT</b>	<b>MOV</b>	<b>R3 , #0xFFE</b>
	<b>ROL</b>	<b>R3</b>
	<b>JPL</b>	<b>DONE</b>
	<b>SUB</b>	<b>PC , #4</b>
<b>DONE</b>	<b>RET</b>	

**Figure Q1b**

(13 marks)

2. A VIP assembly language program and the contents of several memory variables are given in Figure Q2a.

- (a) Complete the mnemonics for instructions **(a1)** and **(a2)** based on their associated comments and the manner in which the parameters are passed within the calling program **Main**.

(6 marks)

<b>Main</b>	MOV SP, #0xFFFF ;	
	PSH [0x100] ;	
	PSH #0x101 ;	
	CALL SubX ;	
	:	
<b>SubX</b>	PSHM 7	;
	MOV R0, ?	; retrieve value of Var1 on the stack into R0 <b>(a1)</b>
	MOV R1, ?	; retrieve the address of Var2 on the stack into R1 <b>(a2)</b>
<b>Loop</b>	MOV R2, [R1]	;
	RRC R2	;
	JC ToDo	; <b>(b1)</b>
	JMP Done	; <b>(b2)</b>
<b>ToDo</b>	BCSR 1	; <b>(b3)</b>
	RLC R2	; <b>(b4)</b>
	MOV [R1], R2	; <b>(b5)</b>
	ADD R1, #1	; <b>(b6)</b>
	SUB R0, #1	; <b>(b7)</b>
	JNE Loop	;
<b>Done</b>	BSSR 1	;
	RLC R2	;
	MOV [R1], R2	;
	? ; <b>(c1)</b>	
	? ; <b>(c2)</b>	

Address	Contents
Var1 0x100	0x005
Var2 0x101	0xFFFF
Var3 0x102	0x111
Var4 0x103	0xCC
Var5 0x104	0x777

Figure Q2a

- (b) Re-write the code segment given by instructions **(b1)** to **(b7)** so that the loop body will be optimized. Marks will be awarded based on how well you have optimized the given code segment for both execution speed and code size.

(8 marks)

- (c) Give the mnemonics for the last two instructions **(c1)** and **(c2)** so that the subroutine **SubX** is completed correctly.

(4 marks)

- (d) Based on the memory variable values shown in Figure Q2a, give the 12-bit hexadecimal values of **Var1**, **Var2**, **Var3**, **Var4**, and **Var5** immediately after returning from the subroutine **SubX**. You may assume that the program execution begins at the label **Main**.

(7 marks)

3. (a) Considering only Redundant Array of Independent Disks (RAID) configurations 0, 1, 5 and 6, state for each part, (i) to (iii) shown below, which configuration provides the feature described. For each stated configuration, explain its basic working principle.
- (i) The RAID configuration has 0% disk redundancy. (4 marks)
  - (ii) The RAID configuration has 100% disk redundancy. (4 marks)
  - (iii) The RAID configuration can recover from more than one hard disk failure. (4 marks)
- (b) Figure Q3a shows a basic Sketch program used on the Arduino UNO R3 board to enable UART communication between the board and its interfacing computer.

```

void setup() {
    //Arduino serial configuration
    //19200 baud rate
    //1 start bit, 8 data bits
    //1 even parity bit, 1 stop bit
    Serial.begin(19200,SERIAL_8E1);

    //Print title with ending line break
    Serial.println("ASCII Table ~ Character Map");
}

//Initialize variable to ASCII character 'U'
int thisByte = 'U';

void loop() {
    Serial.write(thisByte);
    delay(1000);
}

```

**Figure Q3a**

Note: Question No. 3 continues on Page 5

- (i) The program is compiled and loaded onto the Arduino board. On the PC, the Arduino serial monitor program is started. Assume the software configurations are correct, describe clearly what are expected to be observed on the serial monitor program. (3 marks)
- (ii) The board is configured to send an 8-bit data packet with even parity. Determine the maximum data transfer rate in characters per second. (3 marks)
- (iii) The board may also be configured to send the same data packet with no parity. Using this configuration with no parity, determine the minimum baud rate setting needed to transfer at least 2950 characters per second. The supported baud rates are 19200, 28800, 38400 and 57600. Justify your answer. (5 marks)
- (iv) Describe what would be observed if the program shown in Figure Q3a is compiled and loaded, but with the whole **setup** routine omitted. (2 marks)
4. (a) A microprocessor system uses an embedded DMA controller to handle its data transfer with an I/O device. Given that the bus cycle takes 100ns, the transfer of the bus control between the CPU and the DMA controller takes 50 ns and the I/O device has a data transfer rate of 125 Kbyte/sec.
- (i) If burst-mode is used by the DMA controller, how long will it take to transfer a block of 10 Kbyte data? (5 marks)
- (ii) Repeat the calculation described in part (i) if the DMA controller is using the cycle-stealing mode. Assume that one byte is transferred in each cycle. (5 marks)

Note: Question No. 4 continues on Page 6

- (b) A VIP assembly language program is shown in Figure Q4a. Consider a VIP processor organization with 4 pipeline stages: Fetch Instruction (F), Decode (D), Execute (E) and Store (S).

<b>ADD</b>	<b>R2 ,R1</b>	<b>; R2 = R2+R1 (I1)</b>
<b>RAR</b>	<b>R2</b>	<b>; R2 = R2÷ 2 (I2)</b>
<b>ADD</b>	<b>R3 ,R1</b>	<b>; R3 = R3+R1 (I3)</b>
<b>RAR</b>	<b>R3</b>	<b>; R3 = R3÷ 2 (I4)</b>
<b>ADD</b>	<b>R2 ,R3</b>	<b>; R2 = R2+R3 (I5)</b>

**Figure Q4a**

- (i) Data dependency between instructions can lead to pipeline conflict. With the help of a pipeline diagram, identify one (or more) dependency between the instructions shown in Figure Q4a. Explain the cause of each identified data dependency.

(6 marks)

- (ii) Show how the instructions can be reordered to mitigate pipeline conflict (by writing down the improved program in the answer booklet). With the help of a pipeline diagram, illustrate how the pipeline conflict is mitigated.

(6 marks)

- (c) The Booth algorithm is to be used to multiply two numbers, namely 0x155 and 0x037. State your choice of multiplicand and multiplier that will result in the least number of operations. Give reasons for your choice. **There is no need to show the solution for the multiplication.**

(3 marks)

## VIP Instruction Encoding – Opcode Formats

11	10	9	8	7	6	5	4	3	2	1	0
0-7	Dual operand			d				s			
8	Short Move			d				n			
9-A	Unary/Control			op-code				operand = s, d or n			
B-F	JMP			2's complement -128 to +127 relative							

## Group 1 – Dual-Operand Instructions (Opcode: 000 to 8FF)

Bits 8-11	Name	Bits 4-7	Bits 0-3	Operation	Flags
0	MOV	d	s	$d \leftarrow s$	NZ
1	AND	d	s	$d \leftarrow d .AND. s$	NZ
2	OR	d	s	$d \leftarrow d .OR. s$	NZ
3	EOR	d	s	$d \leftarrow d .EOR. s$	NZ
4	ADD	d	s	$d \leftarrow d + s$	VNZC
5	ADDC	d	s	$d \leftarrow d + s + \text{carry}$	VNZC
6	SUB	d	s	$d \leftarrow d + (.NOT. s) + 1$	VNZC
7	CMP	d	s	$d + (.NOT. s) + 1$	VNZC
8	MOVS	d	n	$d \leftarrow n$	

## Group 2 – Unary and Control Instructions (Opcode: 900 to 9FF)

Bits 4-7	Name	Bit 0-3	Operation	Flags
0	INC	d	$d \leftarrow d + 1$	C
1	DEC	d	$d \leftarrow d + 0xFF$	NZC
2	ROR	d	Rotate d right : msb $\leftarrow$ lsb; and C $\leftarrow$ lsb	NZC
3	ROL	d	Rotate d left : lsb $\leftarrow$ msb; and C $\leftarrow$ msb	NZC
4	RRC	d	Rotate d right including carry	NZC
5	RLC	d	Rotate d left including carry	NZC
6	RAR	d	Rotate d 'arithmetic' right preserving msb	NZC
7	PRSG	d	Left shift lsb from EOR (bits 11,5,3,0)	NZC
8	INV	d	$d \leftarrow .NOT. d$	NZ
9	NEG	d	$d \leftarrow (.NOT. d) + 1$	NZC
A	DADD	s	AR $\leftarrow$ AR + s + carry (as 3 BCD digits)	ZC
B	UMUL	s	R1:R0 $\leftarrow$ unsigned R0 times unsigned s	Z
C	TST	s	$s + 0$	NZ
D	EXEC	s	Execute s as an instruction	implied
E	BCSR	n	SR (bits 3-0) $\leftarrow$ SR .AND. (.NOT. n)	explicit
F	BSSR	n	SR (bits 3-0) $\leftarrow$ SR .OR. n	explicit

## Group 3 – Unary and Control Instructions (Opcode: A00 to AFF)

Bits 4-7	Name	Bits 0-3	Operation	Flags
0	PSH	s	$SP \leftarrow SP-1; (SP) \leftarrow s$	
1	POP	d	$d \leftarrow (SP); SP \leftarrow SP+1$	explicit if d=SR
2	PSHM	3:2:1:0	Push R3:2:1:0 to stack, R3 first	
3	POPM	3:2:1:0	Pop R3:2:1:0 from stack, R3 last	
4	CALL	s	$SP \leftarrow SP-1; (SP) \leftarrow \text{Return Address}$ PC $\leftarrow$ Effective address	
5	RET	n	$PC \leftarrow (SP) + n; SP \leftarrow SP+1$	
6			See subgroup 3a	
7	RCN	n	Count for next rotate instruction. if n=0 use bits 3:2:1:0 of AR	
8	JDAR	$\pm n$	$AR \leftarrow AR-1$ , if AR != 0, PC $\leftarrow PC \pm n$	
9	JPE	$\pm n$	If parity of AR is even, PC $\leftarrow PC \pm n$	
A	JPL	$\pm n$	If N = 0, PC $\leftarrow PC \pm n$	
B	JVC	$\pm n$	If V = 0, PC $\leftarrow PC \pm n$	
C	JGE	$\pm n$	If N = V, PC $\leftarrow PC \pm n$	
D	JLT	$\pm n$	If N != V, PC $\leftarrow PC \pm n$	
E	JGT	$\pm n$	If Z = 0 and N = V, PC $\leftarrow PC \pm n$	
F	JLE	$\pm n$	If Z = 1 or N != V, PC $\leftarrow PC \pm n$	

## Appendix 1

## VIP Instruction Set Summary Chart

## Group 1 – Jump Instructions (8-bit Range) (Opcode: B00 to FFF)

Bits 8-11	Name	n = Bits 0 to 7	Operation
B	JMP = BRA	-128 to +127	$PC \leftarrow PC \pm n$
C	JEQ = JZ	-128 to +127	If Z=1, $PC \leftarrow PC \pm n$
D	JNE = JNZ	-128 to +127	If Z=0, $PC \leftarrow PC \pm n$
E	JHS = JC	-128 to +127	If C=1, $PC \leftarrow PC \pm n$
F	JLO = JNC	-128 to +127	If C=0, $PC \leftarrow PC \pm n$

## Group 3a – Control Instructions (Opcode: A60 to A6F)

Bits 4-7	Name	Bits 0-3	Operation
6	RETI	0	$SR \leftarrow (SP); SP \leftarrow SP+1;$ $PC \leftarrow (SP); SP \leftarrow SP+1$
6	SWI	1	$SP \leftarrow SP-1; (SP) \leftarrow PC;$ $SP \leftarrow SP-1; (SP) \leftarrow SR; PC \leftarrow (0x009)$
6	WAIT	2	IE $\leftarrow 1$ ; Execution resumes after interrupt signal
6	HALT	3	Stop execution. Non-maskable interrupt or hardware reset to exit.
6	STOP	4	Stop execution. Reset to exit.
6	SYNC	8	Pulse SYNC output pin high for 1 clock cycle
6	NOP	9	No operation
6	LOCK	A	Block interrupts and bus sharing
6	UNLK	B	Allow interrupts and bus sharing
6	MSS	C to F	Memory Space Select override

## Addressing Modes

Hex	Symbol	Location of Data	Availability
0	R0	Register R0	Both d and s
1	R1	Register R1	Both d and s
2	R2	Register R2	Both d and s
3	R3	Register R3	Both d and s
4	[R0]	Register R0 indirect	Both d and s
5	[R1]	Register R1 indirect	Both d and s
6	[R2+n]	Register R2 with offset indirect	Both d and s
7	[R3+n]	Register R3 with offset indirect	Both d and s
8	AR	Data is in Auxiliary Register	Both d and s
9	SR	Status Register	Both d and s
A	SP	Stack Pointer	Both d and s
B	PC	Program Counter	Both d and s
C	#n	Immediate, (or just n for CALL)	s only
D	[n]	Absolute (code space for CALL)	Both d and s
E	[SP+n]	SP with offset indirect	Both d and s
F	[PC+n]	PC with offset indirect (CALL is relative with PC+n)	Both d and s

Notation: d = destination; s = source

## Description of bits in Status Register

SR	F	R	Description
11-8	*	*	Reserved
7-4	*	*	Defined but not described here
3	V	0	Set if 2's complement sign is incorrect
2	N	0	Is most significant bit of result
1	Z	0	1 if result is zero, otherwise 0
0	C	0	1 if carry out, otherwise 0

Notation: SR=Bits in register; F=Name of flag; R=Value after reset

END OF PAPER

**CE1006 COMPUTER ORGANISATION AND ARCHITECTURE**  
**CZ1006 COMPUTER ORGANISATION AND ARCHITECTURE**

Please read the following instructions carefully:

- 1. Please do not turn over the question paper until you are told to do so. Disciplinary action may be taken against you if you do so.**
2. You are not allowed to leave the examination hall unless accompanied by an invigilator. You may raise your hand if you need to communicate with the invigilator.
3. Please write your Matriculation Number on the front of the answer book.
4. Please indicate clearly in the answer book (at the appropriate place) if you are continuing the answer to a question elsewhere in the book.