NANYANG TECHNOLOGICAL UNIVERSITY

SEMESTER 2 EXAMINATION 2014-2015

CE1006/CZ1006 – COMPUTER ORGANIZATION AND ARCHITECTURE

Apr/May 2015 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.

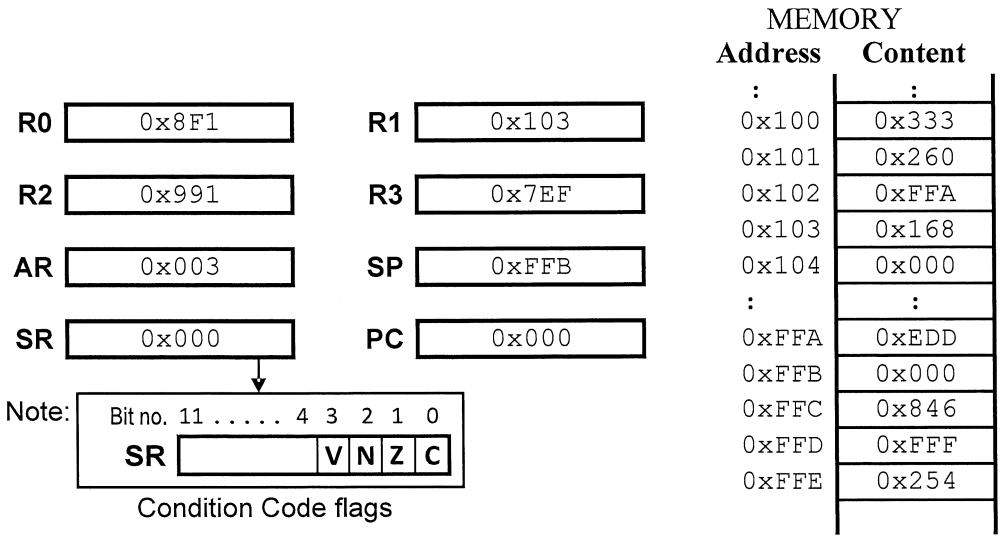

Figure Q1a

Note: Question No. 1 continues on Page 2

(a) Give (*in hexadecimal*) the 12-bit contents in the two registers **R2** 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 R2,#0xABC`

(ii) `MOV R2,[SP]`

(iii) `OR R2,[R1]`

(iv) `PSH R2`

(v) `ADD R2,R0`

(vi) `RRC R2`

(12 marks)

(b) Write the equivalent C high-level language program that is represented by the VIP assembly language program given in Figure Q1b. You may assume that your C integer variables **Var0** and **Var1** are represented by the memory addresses given by labels **Var0** and **Var1** respectively in Figure Q1b.

(8 marks)

```
START       MOVS  [Var0],#0
            MOV   [Var1],#10
Label_1     JEQ   Label_3
            CMP   [Var1],#5
            JGE   Label_2
            INC   [Var0]
Label_2     DEC   [Var1]
            JMP   Label_1
Label_3     ADD   [Var0],[Var0]
```

Figure Q1b

(c) Using the initial conditions given in Figure Q1a and the information given in Appendix 1, give (*in hexadecimal*) the 12-bit contents in the two registers **R3** and **SR** immediately after the execution of the VIP instruction `ADD R3,[PC+0xFFE]`.

(5 marks)

2. (a) An incomplete VIP assembly language program and the contents of several registers and memory variables are given in Figure Q2. Give the five mnemonics of (**I1**) to (**I5**) in the calling program and the six mnemonics of (**S1**) to (**S7**) in the subroutine that will complete the instructions based on their associated comments.

(14 marks)

```
Address
0x000  Main  MOV   SP,#0xFFF      ; Initialize stack pointer
0x002          ?                  ; Push decimal value 4 (i.e. array size) to stack     (I1)
0x004          ?                  ; Push start address of memory array KA to stack      (I2)
0x006          ?                  ; Call subroutine SubZ                                (I3)
0x008          ?                  ; Copy result passed out to memory variable Ans       (I4)
0x00A          ?                  ; Remove stack parameters                             (I5)
               :
0x050  SubZ  PSHM  0x00A          ; Save away used registers
             MOV   R3,SP          ; Setup frame pointer R3
             SUB   SP,#2          ; Create 2 local variables V1 and V2 on stack frame
             MOVS  [R3+0xFFF],?   ; Clear local variable V1                             (S1)
             MOV   [R3+0xFFE],?   ; Move array size on stack into local variable V2     (S2)
             MOV   R1,?           ; Move start address of array on stack into R1        (S3)
       Loop  ADD   [R3+0xFFF],?   ; Add next array element to local variable V1         (S4)
              ?     ?             ; Increment R1                                        (S5)
              ?     ?             ; Decrement local variable V2                         (S6)
              ?     ?             ; Loop back until all array elements are added        (S7)
             MOV   R0,[R3+0xFFF]  ; Move result of summation into R0 for passing out
               :                  ; to be completed
               :                  ; as specified in
               :                  ; question Q2(c)
```

R0	0xAAA	R1	0xBBB
R2	0xCCC	R3	0xDDD

Contents in Registers

Address	Contents
Ans 0x100	0x000
KA 0x101	0x002
0x102	0x036
0x103	0x023
0x104	0x024

Contents in Memory

Figure Q2

(b) Draw a labeled memory map to show all the known contents on the system stack immediately after the execution of the instruction at (**S2**). Your diagram must include (*in hexadecimal*) the stack addresses and their respective contents. Assume the program given in Figure Q2 begins execution at the address label **Main**.

(7 marks)

(c) With reference to Figure Q2, give the sequence of VIP assembly language instructions that will allow subroutine **SubZ** to return to the calling program in a correct manner.

(4 marks)

3. (a) For each listed device in Table Q3, state if the memory is volatile or non-volatile. Also, categorize the device as either system or storage memory. Provide your answers using the table format given in Table Q3.

(6 marks)

Table Q3

Device	Volatile/Non-volatile	System/Storage Memory
NAND Flash		
Magnetic Hard Disk		
SRAM		

(b) Describe what Execute-in-Place (XIP) means and explain why the NOR flash memory supports XIP.

(3 marks)

(c) Which type of flash memory would you choose when designing high capacity USB Flash Drives for the consumer market? Give one reason for your choice.

(3 marks)

(d) Figure Q3 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() {
    // 19200 baud rate
    // 1 start bit, 7 data bits
    // 1 even parity bit, 1 stop bit
    Serial.begin(19200,SERIAL_7E1);
}

//Initialize variable to 'y' (Ascii value = 0x79)
int thisByte = 'y';

void loop() {
    Serial.write(thisByte);
    delay(1000); // 1 Second Delay
}
```

Figure Q3

Note: Question No. 3 continues on Page 5

With reference to Figure Q3, answer the following questions:

(i) What is the maximum number of characters that can be transmitted per second?

(3 marks)

(ii) Draw the waveform of the UART signal transmitted by the Arduino Board running the given code. Label the START, STOP and PARITY bits.

(5 marks)

(iii) A UART receiver is configured to receive the signal transmitted by the Arduino Board running the given code. Instead of data '0x79', the receiver repeatedly received a data of '0x06' at an interval of about 1 second. No error is detected by the UART receiver. Determine the UART configuration used in the receiver and give clear explanations for your answer.

(5 marks)

4. (a) State one advantage and one disadvantage for each of the following cache mapping scheme:

- Direct mapped cache.
- N-Way set associative cache.

(6 marks)

(b) Explain why the code shown in Figure Q4a allows a cache to have a high hit rate.

(6 marks)

```
for (i=0; i<10; i++)
{
    a[i] = b[i]*c[i];
}
```

Figure Q4a

Note: Question No. 4 continues on Page 6

(c) An array consisting of the length of 8 wires is given by L[0], L[1], … , L[7]. Describe a scheme to compute the average length of the 8 wires that will yield a result with the highest precision based on the following specifications:

- 12-bit registers are used for storing data and result.
- Only single-precision, fixed-point arithmetic is used.
- Maximum possible length of each wire is 0x3FF.

Give your answer in the form of a mathematical expression and justify your answer.

Note: you do not need to write any code.

(7 marks)

(d) Consider a processor with 4 pipeline stages: Fetch Instruction (F), Decode (D), Execute (E) and Store (S). Figure Q4b shows the initial 12-bit hexadecimal contents of several registers in the processor and a program that is subsequently executed.

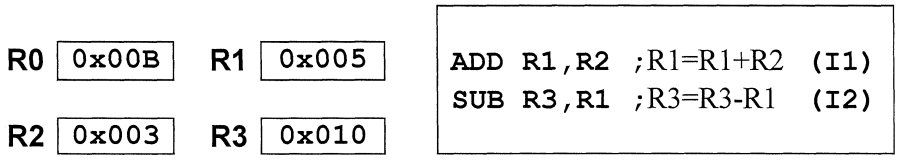

Figure Q4b

(i) What is the hexadecimal content of register **R3** after instructions `I1` and `I2` are sequentially executed starting with instruction `I1`?

(2 marks)

(ii) Identify the pipeline conflict observed in part Q4(d)(i) and explain the cause of the conflict. Suggest one method to resolve this conflict.

(4 marks)

Appendix 1

VIP Instruction Set Summary Chart

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 ← s	NZ
1	AND	d	s	d ← d .AND. s	NZ
2	OR	d	s	d ← d .OR. s	NZ
3	EOR	d	s	d ← d .EOR. s	NZ
4	ADD	d	s	d ← d + s	VNZC
5	ADDC	d	s	d ← d + s + carry	VNZC
6	SUB	d	s	d ← d + (.NOT. s) + 1	VNZC
7	CMP	d	s	d + (.NOT. s) + 1	VNZC
8	MOVS	d	n	d ← n	

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

Bits 4-7	Name	Bit 0-3	Operation	Flags
0	INC	d	d ← d + 1	C
1	DEC	d	d ← d + 0xFFF	NZC
2	ROR	d	Rotate d right : msb ← lsb; and C ← lsb	NZC
3	ROL	d	Rotate d left : lsb ← msb; and C ← 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 ← .NOT. d	NZ
9	NEG	d	d ← (.NOT. d) + 1	NZC
A	DADD	s	AR ← AR + s + carry (as 3 BCD digits)	ZC
B	UMUL	s	R1:R0 ← 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) ← SR .AND. (.NOT. n)	explicit
F	BSSR	n	SR (bits 3-0) ← 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 ← SP-1; (SP) ← s	
1	POP	d	d ← (SP); SP ← 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 ← SP-1; (SP) ← Return Address PC ← Effective address	
5	RET	n	PC ← (SP) + n; SP ← 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	±n	AR ← AR-1, if AR != 0, PC ← PC ± n	
9	JPE	±n	If parity of AR is even, PC ← PC ± n	
A	JPL	±n	If N = 0, PC ← PC ± n	
B	JVC	±n	If V = 0, PC ← PC ± n	
C	JGE	±n	If N = V, PC ← PC ± n	
D	JLT	±n	If N != V, PC ← PC ± n	
E	JGT	±n	If Z = 0 and N = V, PC ← PC ± n	
F	JLE	±n	If Z = 1 or N != V, PC ← PC ± n	

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 ← PC ± n
C	JEQ = JZ	-128 to +127	If Z=1, PC ← PC ± n
D	JNE = JNZ	-128 to +127	If Z=0, PC ← PC ± n
E	JHS = JC	-128 to +127	If C=1, PC ← PC ± n
F	JLO = JNC	-128 to +127	If C=0, PC ← PC ± n

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

Bits 4-7	Name	Bits 0-3	Operation
6	RETI	0	SR ← (SP); SP ← SP+1; PC ← (SP); SP ← SP+1
6	SWI	1	SP ← SP-1; (SP) ← PC; SP ← SP-1; (SP) ← SR; PC ← (0x009)
6	WAIT	2	IE ← 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.