NANYANG TECHNOLOGICAL UNIVERSITY

SEMESTER 1 EXAMINATION 2017-2018

CE1006/CZ1006 – COMPUTER ORGANIZATION AND ARCHITECTURE

Nov/Dec 2017 Time Allowed: 2 hours

INSTRUCTIONS

1. This paper contains 4 questions and comprises 7 pages.

2. Answer **ALL** questions.

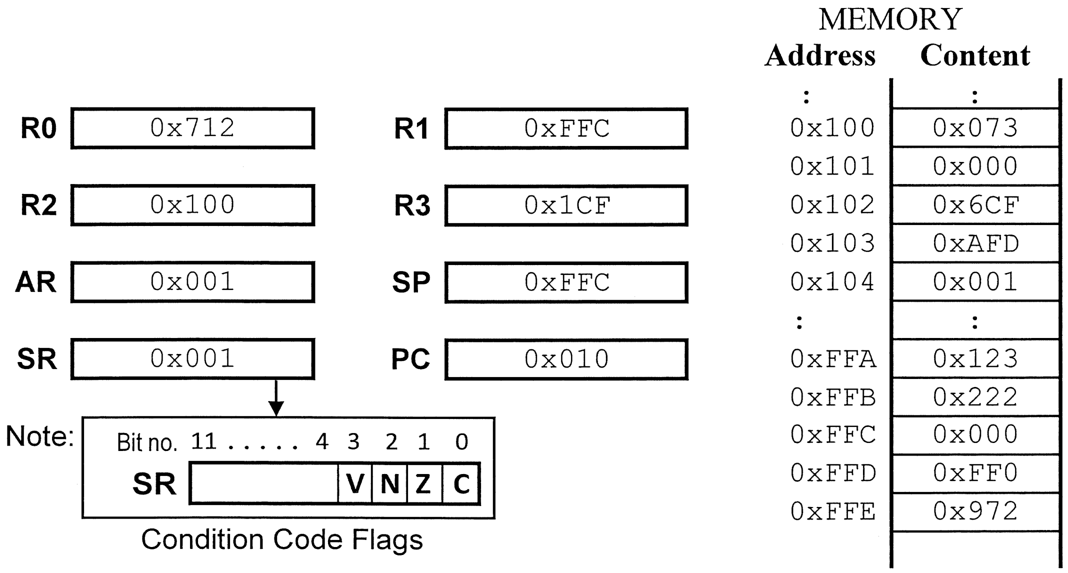
3. This is a closed-book examination.

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.



R0	0x712	R1	0xFFC
R2	0x100	R3	0x1CF
AR	0x001	SP	0xFFC
SR	0x001	PC	0x010

MEMORY

Address	Content
:	:
0x100	0x073
0x101	0x000
0x102	0x6CF
0x103	0xAFD
0x104	0x001
:	:
0xFFA	0x123
0xFFB	0x222
0xFFC	0x000
0xFFD	0xFF0
0xFFE	0x972

Figure Q1a

Note: Question No. 1 continues on Page 2

(a) Give (*in hexadecimal*) the 12-bit contents in the two registers **R0** and **SR**, immediately after the execution of each instruction given below.

Note: Instructions (i) to (v) 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    R0,#0xFFB

(ii)   MOV    R0,[R1]

(iii)  AND    R0,[R2+2]

(iv)   ADD    R0,R3

(v)    RRC    R0
```

(10 marks)

(b) A series of VIP assembly instructions, their associated start addresses in code memory and the contents of three memory variables in data memory are given in Figure Q1b. Give (*in hexadecimal*) the 12-bit contents in registers **R0**, **R2**, **SR**, **PC** and memory location at address **0x052** immediately **after** the execution of the instruction at the label **Exit**. Assume execution of the instructions begins at the label **Start.**

(7 marks)

```
Address
0x000   Start  MOV   R2,#0x050     ;
0x002          MOV   R0,#1         ;
0x004          ADD   R0,[R2+0]     ;
0x006          CMP   R0,[0x051]    ;
0x008          JGE   Exit          ;
0x009          MOV   PC,#0x004     ;
0x00B   Exit   MOV   [R2+2],R0     ;
                 :
```

Address	Contents
0x050	0x003
0x051	0x009
0x052	0x000

Contents in Data Memory

Figure Q1b

(c) Re-write the program segment in Figure Q1b so that it will execute in a **position independent** manner. You must ensure your modifications do not change the functionality of the program with respect to the data memory variables but you are free to use whichever registers you so choose. Assume that the code segment starting at address **0x000** and the data segment starting at address **0x050** constitute a single contingious program entity that can be relocated to another location in memory.

(8 marks)

2. (a) An incomplete VIP assembly language program is given in Figure Q2. Give the mnemonics of (**I1**) to (**I6**) that will complete the missing instructions based on their associated comments. Assume memory variable **RESULT** is in address **0x200**.

(7 marks)

```
Address
0x000   Begin  MOV  SP,#0xFFF     ; Initialise the stack pointer
               MOV  R0,#5         ; Move value of X which is 5 into register R0
               PSH  R0            ; Push value of X in R0 to the stack
               MOV  [0x200],#0    ; Clear memory variable  RESULT
               PSH  #0x200        ; Push address of memory variable RESULT to the stack
               CALL  SubA         ; Call subroutine SubA
        Done          :           ; Instruction immediate after CALL instruction
                      :
        SubA          ?           ; Save to stack all registers used in subroutine      (I1)
                      ?           ; Retrieve value of X from stack into R1              (I2)
                      ?           ; Retrieve address of RESULT from stack into R2       (I3)
               MOV  R0,R1         ;
               AND  R0,#0x001     ;
               JEQ  Skip1         ;
               ADD  [R2+0],R1     ;
        Skip1  SUB  R1,#1         ;
               JEQ  Skip2         ;
               PSH  R1            ;
               PSH  R2            ;
               CALL  SubA         ;
                      ?           ; Remove parameters on the stack                      (I4)
        Skip2         ?           ; Restore all used registers                          (I5)
                      ?           ; Return to calling program                           (I6)
```

Figure Q2

(b) With reference to Figure Q2, give (*in hexadecimal*) the 12-bit contents in registers **R0**, **PC**, **SP** and memory variable **RESULT** just **before** the execution of the instruction at the label **Done**. Assume execution of the instructions begins at the address label **Begin** and the first instruction is stored starting at address **0x000**. Based on your analysis, briefly describe the mathematical operation implemented by the subroutine **SubA** to compute the final value of **RESULT** based on the value of **X** passed into the subroutine.

(8 marks)

(c) Re-write the entire program given in Figure Q2 to produce the most optimized code in terms of execution speed. Your modifications must retain the functionality of the program and produce the same final outcome in memory variable **RESULT**. More marks will be given for a more optimized solution.

(10 marks)

3. (a) Determine the most suitable memory to be used for each requirement listed in Table Q3a below and give explanation for your choice. The memory types available are: SRAM, DRAM, EEPROM, NAND Flash, NOR Flash and Magnetic HDD. Note that each memory type may be used more than once if deemed most suitable.

(6 marks)

Table Q3a

Requirements	Memory Type and Reason for Choice
Main Storage Memory for a Commercial Company's Data Center	
External System Memory for a processor that does not have any internal (on-chip) memory. Product powers on/off weekly.	
Very High Speed internal memory of a high end processor.	

(b) Explain how the design of a Floating Gate Transistor enables storage of logic '1' and logic '0'. Illustrate the explanation using the I-V graph of a floating gate transistor and the memory read operation.

(6 marks)

(c) Explain in detail why the maximum SPI bus data transfer rate is always much higher than the maximum UART bus transfer rate. For example, in a processor, the maximum transfer rate via SPI is 50Mbps while the maximum UART transfer rate is only 1Mbps.

(4 marks)

(d) Give one reason why the statement below may be valid and one reason why the same statement may not be valid. Higher marks will be given to more specific answers.

"The INTEL CPU in my laptop is running at 2.2GHz while the INTEL CPU in your laptop is running at 1.8GHz. Since both laptops have 16GBtye system memory and have identical HDD, my laptop's performance is slightly better than yours."

(4 marks)

Note: Question No. 3 continues on Page 5

(e) Two computer systems, COMP-A and COMP-B are connected via a UART link. Table Q3b shows the UART configuration for COMP-A and COMP-B.

Table Q3b

COMP-A UART	COMP-B UART
1 Start, 8 Data, 1 Stop Bit	1 Start, 8 Data, 1 Stop Bit
No Parity Scheme	No Parity Scheme
Baud rate = 115200 bps	Baud rate = Unknown
Allowable baud rate: 115200, 57600, 38400, 28800, 19200, 14400, 9600, 4800, 2400.	

COMP-B transmitted a data of value 0x0D to COMP-A. But the data value received by COMP-A is 0x1C. With the aid of a UART timing diagram, compute the UART baud rate of COMP-B.

(5 marks)

4. (a) Explain why the cache memory controller always transfers one cache block consisting of multiple bytes instead of just the single byte that the CPU requires.

(4 marks)

(b) Consider a system with the following characteristics.

- Main Memory Size = 1 MByte
- Direct Mapped Cache
- Cache Size = 256 Bytes
- Cache Block Size = 32 bytes

Which Cache block would the Main Memory address 0x00999 be mapped to and what is the corresponding TAG Value?

(8 marks)

(c) Compare the advantages and disadvantages of the fixed point and floating point number representation system with respect to Range and Precision. There should be at least one advantage and one disadvantage for each of the number systems.

(4 marks)

Note: Question No. 4 continues on Page 6

(d) Consider a processor with 4 pipeline stages: Fetch Instruction (F), Decode (D), Execute (E) and Store (S). Assume that

- Branch target address is calculated at the execute stage
- Instruction length for every instruction is one word long
- Each pipeline stage takes 1 cycle to complete
- No resource conflicts
- Delayed Branching disabled.
- This processor is not the VIP processor

(i) List all possible data dependencies that the code in Figure Q4 may encounter.

(3 marks)

(ii) What are the modification(s) to the code in Figure Q4 that can be done to best resolve the pipeline conflicts due to the data dependencies listed in Q4(d)(i) above?

(3 marks)

(iii) A 5-stage pipeline processor is created by splitting the Store stage of the original 4-stage pipeline into two stages: S1 and S2, where register content are updated in S1 while memory content are updated in S2. The new processor has 5 pipeline stages: F-D-E-S1-S2. Will the data dependencies listed in Q4(d)(i) still exist? Illustrate your answer with a pipeline diagram.

(3 marks)

```
      MOV   AR, #10     ; I1
      MOV   R0, #0x200  ; I2
      MOV   R1, #0x100  ; I3
Loop  SUB   [R0],[R1]   ; I4
      INC   R1          ; I5
      JDAR  Loop        ; I6
      ADD   R2, [R0]    ; I7
      MOV   [R1], R2    ; I8
```

Figure Q4

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.