NANYANG TECHNOLOGICAL UNIVERSITY

SEMESTER 2 EXAMINATION 2018-2019

CE1006/CZ1006 – COMPUTER ORGANIZATION AND ARCHITECTURE

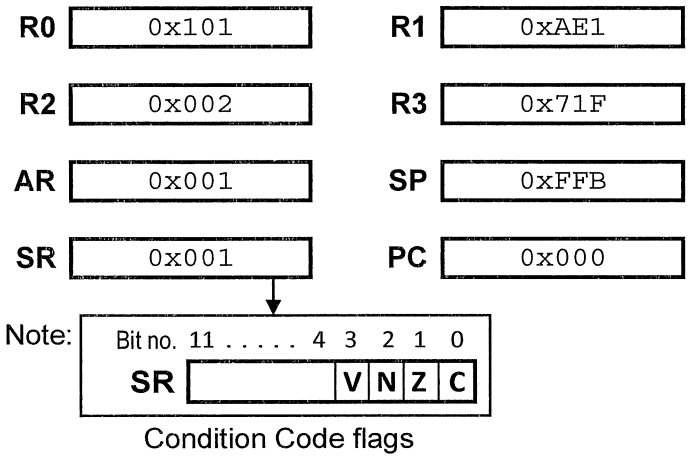
Apr/May 2019 Time Allowed: 2 hours

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
:	:
0x100	0x40C
0x101	0x002
0x102	0x0B0
0x103	0x801
0x104	0x100
:	:
0xFF9	0x123
0xFFA	0x0EE
0xFFB	0xFFF
0xFFC	0x111
0xFFD	0x888
0xFFE	0x721

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 (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 R1,#0x000`

(ii) `MOV R1,[R0]`

(iii) `OR R1,[SP+2]`

(iv) `RRC R1`

(v) `SUB R1,R3`

(10 marks)

(b) With reference to the VIP assembly language code segment given in Figure Q1b, answer the questions below using the **initial conditions** shown in Figure Q1a and relevant information in Appendix 1.

```
Start     MOVS  R1,#0
Back      ROL   R3
          JNC   Next
          ADD   R1,[R0]
Next      INC   R0
          CMP   R0,#0x104
          JLO   Back
Check     POP   R2
Proceed   MOV   R0,PC
          MOV   PC,[0x104]
Loop      NEG   R2
          JMP   Loop
```

Figure Q1b

(i) Give the 12-bit hexadecimal contents in registers `R0`, `R1`, `R2`, `R3`, `SP` and `SR` immediately **after** the execution of the instruction at the label `Check`. Assume instruction execution begins at the label `Start`.

(8 marks)

(ii) Describe what you expect to observe if code execution **continues** from the label `Proceed`. You must describe clearly which registers are expected to change and how they will change during execution.

(7 marks)

2. An incomplete VIP assembly language program and the contents of three memory variables are given in Figure Q2.

(a) Write the equivalent C high-level language construct that represents the four VIP instructions (**A1**) to (**A4**). Assume that registers **R0, R1** and **R2** are represented by the C integer variables **R0, R1** and **R2**.

(4 marks)

(b) Give the mnemonics of (**I1**) to (**I10**) that will complete the missing instructions based on their associated comments.

(10 marks)

(c) Give a concise equivalent mathematical expression implemented by the subroutine SubA in terms of the values of memory variables **Nx**, **Ny** and **Nz**. Based on the values in variables **Nx** and **Ny**, give the value in **Nz** after returning from SubA. Assume execution begins at the label Main.

(5 marks)

(d) Re-write the code segment given by instructions (**C1**) to (**C6**) to produce the most optimised code in terms of execution speed. Your solution only needs to ensure that an identical result is obtained in **R0** at the end of the equivalent code segment. More marks will be given for a more optimised solution.

(6 marks)

```
Main    ?                ; Push address of memory variable Nz to the stack   (I1)
        ?                ; Push the value of Nx to the stack                 (I2)
        ?                ; Push the value of Ny to the stack                 (I3)
        CALL SubA
          :
SubA    ?                ; Saved all used registers to the stack             (I4)
        ?                ; Retrieve value of Nx from the stack into R1       (I5)
        ?                ; Retrieve value of Ny from the stack into R2       (I6)
        ?                ; Retrieve address of Nz from the stack into R3     (I7)
        CMP  R2,#0       ;  (A1)
        JGE  Skip        ;  (A2)
        NEG  R2          ;  (A3)
Skip    MOV  R0,R1       ;  (A4)
Loop    SUB  R2,#1       ;  (C1)
        JPL  Next        ;  (C2)
        JMP  Done        ;  (C3)
Next    ADD  R0,R1       ;  (C4)
        ADD  R1,R1       ;  (C5)
        JMP  Loop        ;  (C6)
Done    ?                ; Save R0 into Nz using stack parameter passed in   (I8)
Exit    ?                ; Restore all used registers from the stack         (I9)
        ?                ; Return from subroutine                            (I10)
```

	Address	Contents
Nx	0x200	0x012
Ny	0x201	0x005
Nz	0x202	0x000

Memory Variables

Figure Q2

3. (a) You are tasked to design the next generation Durian Notebook PC with the following specifications

- Able to sustain drop from 1.2 meter above ground.
- Uses the 2.8GHz Quad-core DU2 processor with 256KByte Cache, 1MByte of Internal SRAM and 1MByte of Internal Flash.
- Requires 16GBytes System Memory and 512GBytes Storage Memory, these memories are external to the processor.

(i) Explain whether the Storage Memory should be volatile or non-volatile in nature.

(2 marks)

(ii) The DU2 processor uses the Internal Flash to store the bootloader code and execute the code directly from the Internal Flash upon power up. The bootloader code transfers the Operating System code from the Storage Memory to the System Memory. Explain which flash type (NAND or NOR) should the Internal Flash use.

(2 marks)

(iii) Can the bootloader code be stored in Internal SRAM instead? Explain.

(2 marks)

(iv) Explain why a HDD is not a suitable candidate for the storage memory of this product.

(2 marks)

(b) Give two reasons for the increased popularity of the serial interface bus standard over the parallel bus standard in modern day electronics. Give an example of a parallel bus interface standard that is replaced by a serial bus standard.

(6 marks)

(c) A processor board is connected to a display panel via an RS232 interface. The processor sends the letters 'E' and 'X' to the display without any delay between each letter. Draw the complete RS232 timing diagram of the transmission, label all information/data bits clearly on the diagram. ASCII value for the letters 'E'=0x45 and 'X'=0x58. Configuration used is 1 Start, 7 Data, 1 Stop and Odd Parity. State all assumptions used.

(5 marks)

Note: Question No. 3 continues on Page 5

(d) Given that the typical erasure edurance of modern day NAND flash can be as low as 3000 cycles, how is it that manufacturers of SSD can still claim Mean-Time-Before-Failure (MTBF) of 1 million hours? Note that 1 million hours MTBF is similar to that of a HDD, a media that can sustain almost infinite erasure.

(4 marks)

(e) With reference to Q3(d) above, give one reason why you should execute the "Eject Drive" option in the Windows OS File Explorer before attempting to unplug a portable SSD from your PC.

(2 marks)

4. (a) What is a Translational Lookaside Buffer (TLB)? What does it imply when mapping information of a particular virtual page can be found in the TLB entry? Given the following virtual memory configuration, explain why this is a bad design.

- Paging scheme
- Virtual memory size = 1MByte
- Physical memory size = 64KBytes
- Virtual page size = 2KBytes
- TLB size = 32 entries

(5 marks)

(b) Given a fully associative cache with 4 cache blocks and a 16-Byte cache block size.

(i) What is the difference between First-In-First-Out (FIFO) and Least-Recently-Used (LRU) cache replacement policies?

(2 marks)

(ii) Explain clearly why the user is not able to differentiate whether the cache is using a FIFO or LRU cache replacement policy from the address access sequence below. Suggest a simple modification to the address sequence to enable FIFO and LRU to be differentiated.

- Sequence: 0xA1, 0x34, 0x22, 0x60, 0x28, 0x71, 0x42

(6 marks)

Note: Question No. 4 continues on Page 6

(c) Given a 32-bit fixed point number and a 32-bit floating point number, both numbers are limited by having only 2^{32} possible number representations. Explain clearly why the 32-bit floating number is able to have a much larger range compared to the 32-bit fixed point number. What attributes are traded off?

(4 marks)

(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
- Dynamic Branch Prediction is enabled, the prediction algorithm used is as follows
  - Assume first branch is TRUE
  - Prediction of current branch = results of previous branch, i.e. IF branch results of iteration #1 is TRUE, then algorithm will predict that branch result of iteration #2 is TRUE and vice versa.
- This processor is not the VIP processor

(i) What is the purpose of including a NOP instruction in the code shown in Figure Q4?

(2 marks)

(ii) Calculate the number of cycles required to execute the entire program in Figure Q4.

(6 marks)

```
      MOV   AR, #10     ; I1
      MOV   R0, #200    ; I2
      NOP               ; I3
Loop  ADD   R1, [R0]    ; I4
      INC   R0          ; I5
      JDAR  Loop        ; I6
      MOV   R2, R1      ; I7
      MOV   R3, R0      ; 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 & ARCHITECTURE
CZ1006 COMPUTER ORGANISATION & 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.