NANYANG TECHNOLOGICAL UNIVERSITY

SEMESTER 1 EXAMINATION 2018-2019

CE1006/CZ1006 – COMPUTER ORGANIZATION AND ARCHITECTURE

Nov/Dec 2018 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.

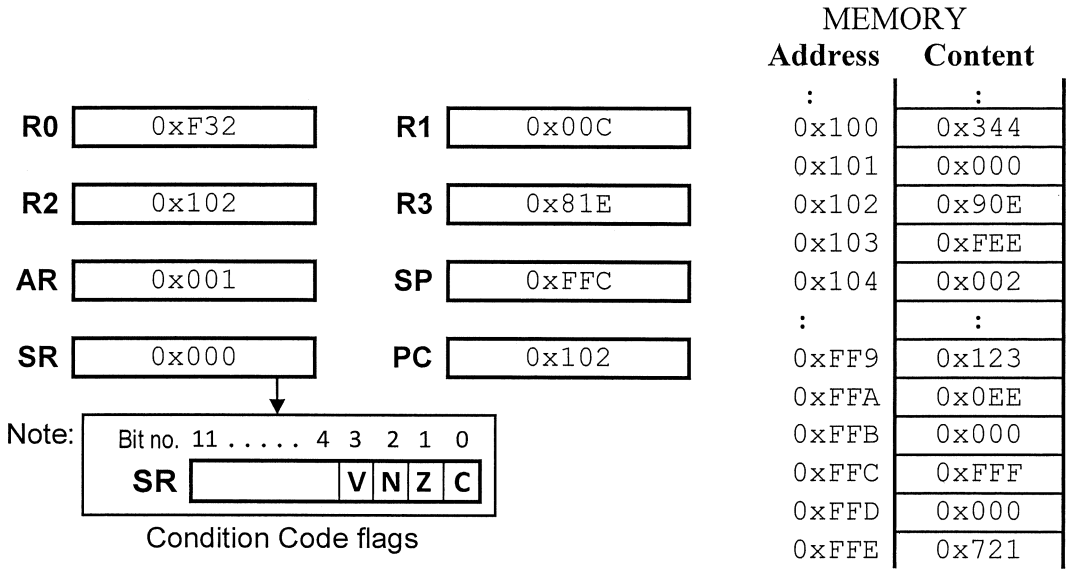

Figure Q1a

Note: Question No. 1 continues on Page 2

(a) Give (*in hexadecimal*) the 12-bit contents in the two registers **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,#0x999`

(ii) `MOV R0,[R2+0xFFF]`

(iii) `EOR R0,R1`

(iv) `ADD R0,R3`

(v) `ROL R0`

(10 marks)

(b) A series of VIP instructions and their start addresses in memory are given in Figure Q1b. Give the 12-bit hexadecimal contents in registers `R0`, `R1`, `R2`, `R3` and `SR` immediately **after** the execution of the instruction at the label `1st_Check` **and also** immediately **after** the execution of the instruction at the label `2nd_Check`. Assume instruction execution begins at the label `Start`. Answer the question using the initial register values shown in Figure Q1a in page 1 and relevant information in Appendix 1 in page 7.

(12 marks)

```
Address
0x000   Start       PSHM   15
0x001               MOV    R0,#0
0x003   Loop        ROR    R2
0x004               JC     Next
0x005               ADD    R0,#1
0x007   Next        SUB    R1,#1
0x009               JNE    Loop
0x00A   1st_Check   POP    R3
0x00B               POPM   12
0x00C               MOV    [0x011],#0x41C
0x00F               MOV    R0,[0x003]
0x011               MOV    R1,R3
0x012               MOVS   R2,#15
0x013   2nd_Check   POP    R3
0x014                :      :
```

Figure Q1b

(c) Identify and replace **three** VIP instructions in Figure Q1b with their respective equivalent optimized mnemonics.

(3 marks)

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

(a) Give the mnemonics of (**I1**) to (**I8**) that will complete the missing instructions based on their associated comments.

(10 marks)

(b) Using the initial values in the three memory variables **Vx**, **Vy** and **Vz** shown, give their new values after the execution of the given program. Assume program execution begins at **Strt** and stops at **Done**.

(6 marks)

(c) Give the equivalent mathematical expression implemented by the given program in terms of the variables **Vx**, **Vy** and **Vz**.

(5 marks)

```
Strt   MOV   SP,#0xFFF    ; Initial the stack pointer
       TST   [0x101]      ; Set or clear N or Z flag based of the value of Vy   (C1)
       JPL   Next         ;                                                     (C2)
       NEG   [0x101]      ;                                                     (C3)
Next     ?                ; Push the value of Vx to the stack                   (I1)
         ?                ; Push the value of Vy to the stack                   (I2)
         ?                ; Push address of memory variable Vz to the stack     (I3)
       CALL  SubZ
Done   ADD   SP,#3
         :
         :
         :
SubZ   PSHM  7
         ?                ; Retrieve value of Vx from the stack into R0         (I4)
         ?                ; Retrieve value of Vy from the stack into R1         (I5)
         ?                ; Retrieve address of Vz from the stack into R2       (I6)
       ADD   R0,R0
         ?                ; Save R0 into Vz using stack parameter passed in     (I7)
       DEC   R1
       JEQ   Skip
       PSH   R0
       PSH   R1
       PSH   R2
       CALL  SubZ
       ADD   SP,#3
Skip   POPM  7
         ?                ; Return from subroutine                              (I8)
```

Address	Contents
Vx 0x100	0x005
Vy 0x101	0xFFD
Vz 0x102	0x000

Memory Variables

Figure Q2

(d) Write the equivalent C high-level language construct that represents the code segment given by the three VIP instructions (**C1**) to (**C3**). Assume the memory variable **Vy** at address **0x101** is represented by the integer variable **Vy** in the C language construct.

(4 marks)

3. (a) You are tasked to design the next generation Orange Smartphone with the following specifications

- Ultra-thin thickness of 6mm
- Uses the 2.4GHz quad-core OZ1 processor with 8MByte Cache and 4MByte Internal RAM
- Requires 16GByte System Memory and 512GByte Storage Memory

(i) Choose ONE memory type that you will use as the storage memory and justify your choice.

(2 marks)

(ii) The OZ1 processor uses 32 registers for internal operations. Explain clearly which memory type the registers use.

(2 marks)

(iii) Give one difference between Cache and Internal RAM.

(2 marks)

(b) You are tasked to design the memory interface in a high performance graphics card with a data transfer rate of 256GByte/s. The memory type used is GDDR5, which is a high speed DRAM variant.

(i) In the initial design, PCB traces of the data bus between the Graphic Processor and the GDDR5 Memory were found to be of unequal length. What potential issue would occur and how can the issue be mitigated?

(4 marks)

(ii) Each GDDR5 chip has 32 data lines and each line has a maximum data rate of 8 Gbit/s. How can the data transfer rate of 256GByte/s be achieved? Over-clocking the GDDR5 memory is not allowed. State any assumptions made.

(3 marks)

(c) An Arduino Board is connected to a Display panel via UART. The Arduino sends the letters 'S' and 'C' to the display without any delay between each letter. Draw the complete UART timing diagram of the transmission and label all information/data bits clearly on the diagram. The ASCII values for the letter 'S' and 'C' are 0x53 and 0x43, respectively. UART configuration is 1 Start bit, 7 Data bits, 1 Stop bit and Even Parity.

(6 marks)

Note: Question No. 3 continues on Page 5

(d) A HDD with the following parameters has just gone through a defragmentation process. Calculate the time needed to transfer 262KBytes of data from the HDD. State all assumptions made in deriving the answer.

- 8 surfaces
- 1024 cylinders
- 256 sectors per track
- 512 bytes per sector
- 12000 RPM
- Average seek time = 5 ms

(6 marks)

4. (a) Describe the write-allocate and write-no-allocate cache write policies. Under what condition (write-hit or write-miss) is these policies used? Explain clearly which policy you would use for a system that performs writes to many random addresses.

(5 marks)

(b) A 2-level cache system is a system with two caches: level 1 (L1) and level 2 (L2). When the CPU requires data, the order of visit will be the L1 cache, followed by L2 cache and lastly the main memory. Consider a system with the following characteristics.

- 2-level Direct Mapped cache
- L1 cache size = 64 KByte, hit rate = 0.8, access time = 5ns
- L2 cache size = 2 MByte, hit rate = 0.9, access time = 10ns
- Main memory size = 1 GByte, access time = 100ns
- L1 cache, L2 cache and main memory accesses do not overlap

Calculate the effective access time of this system.

(6 marks)

(c) Give two examples of arithmetic operations that will result in truncation of the lower order data bit(s) and the corresponding solution to mitigate the loss in accuracy.

(6 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 one cycle to complete
- No Resource Conflicts
- Delayed Branching is enabled
- This processor is not the VIP processor

Identify and describe ALL pipeline conflicts the code in Figure Q4 has when it is executed in the pipeline processor above. Suggest one workaround for each pipeline conflict identified.

```
      MOV   R0, #200    ; I1
      MOV   AR, #10     ; I2
Loop  JDAR  Loop        ; I3
      ADD   R1, [R0]    ; I4
      INC   R0          ; I5
      MOV   R2, R0      ; I6
      MOV   R3, R1      ; I7
```

Figure Q4

(8 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 & 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.