

**NANYANG TECHNOLOGICAL UNIVERSITY****SEMESTER 1 EXAMINATION 2019-2020****CE1006/CZ1006 – COMPUTER ORGANIZATION AND ARCHITECTURE**

Nov/Dec 2019

Time Allowed: 2 hours

**INSTRUCTIONS**

1. This paper contains 4 questions and comprises 10 pages.
2. Answer **ALL** questions.
3. This is a closed-book examination.
4. Questions carry unequal marks.
5. The VIP Instruction Set Summary Chart is provided in Appendix 1 on page 10.

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

MEMORY

Address	Content
:	:
0x100	0x50C
0x101	0x001
0x102	0x0B0
0x103	0x800
0x104	0x000
:	:
0xFF9	0x123
0xFFA	0x0EE
0xFFB	0x7F9
0xFFC	0x111
0xFFD	0x888
0FFE	0x721

**R0** 0x917

**R1** 0x703

**R2** 0x102

**R3** 0xFFFF

**AR** 0x002

**SP** 0xFFB

**SR** 0x000

**PC** 0x000

Note: Bit no. 11 . . . . . 4 3 2 1 0

SR	V	N	Z	C
----	---	---	---	---

Condition Code flags

**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. The 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 , [0x103]**
- (ii) **MOV R0 , [R2+0x002]**
- (iii) **AND R0 , [SP]**
- (iv) **RAR R0**
- (v) **ADD R0 , #0xB12**

(10 marks)

- (b) With reference to the VIP assembly language code segment given in Figure Q1b, answer the questions below.

<b>Start</b>	<b>MOV R1 , #16</b>
	<b>MOV R2 , #16</b>
<b>Loop</b>	<b>CMP R1 , [0x100]</b>
	<b>JGT Done</b>
	<b>ADD R1 , #8</b>
	<b>SUB R2 , #1</b>
	<b>MOV [0x101] , R1</b>
	<b>JMP Loop</b>
<b>Done</b>	<b>PSH R2</b>

**Figure Q1b**

- (i) Give the 12-bit hexadecimal contents in registers **R1**, **R2** and **SR** immediately **after** the execution of the instruction at label **Done**. Assume execution begins at the label **Start** and the initial value in the memory address **0x100** is **0x028**.  
(4 marks)
- (ii) How many memory access cycles in total would be incurred for the code execution in part (i).  
(4 marks)
- (iii) Rewrite the code given in Figure Q1b to optimize its execution speed. More marks are given for a more optimized code. You can use additional registers provided registers **R1**, **R2** and memory address **0x101** have the same content at the end of execution at label **Done** for any initial value in address **0x100**.  
(7 marks)

2. Answer this question based on the incomplete VIP assembly language program in Figure Q2, the contents of the two memory variables given and any relevant information from Appendix 1 in page 10.
- (a) Give the mnemonics of (I1) to (I9) that will complete the missing instructions based on their associated comments. (10 marks)
- (b) Describe briefly what the subroutine **ChkN** is computing from the parameter passed in by the **Main** program. Based on the value of **N1** shown in Figure Q2, give the value that is returned by subroutine **ChkN** in register **R0**. (7 marks)
- (c) Write the equivalent C high-level language construct that will represent the logic of the code segment given by the four VIP instructions (C1) to (C4). Assume register contents **R0** and **R3** are given by the C variables **Var0** and **Var3** respectively. (4 marks)
- (d) Based on the values in the two memory addresses shown in Figure Q2, describe what you think will happen if the value of the **PC** was set to **0x100** and program execution begins at address **0x100**. (4 marks)

Main	?	; Push the value of N1 to the stack	(I1)
	CALL ChkN		
	?	; Copy returned result in R0 into memory variable A1	(I2)
	?	; Remove parameter passed from the stack	(I3)
	:		
ChkN	?	; Save registers to the stack	(I4)
	?	; Retrieve value of N1 from the stack into R1	(I5)
	MOV R2,#12	;	
	MOV R0,#0	;	
	MOV R3,#0	;	
Loop	ROR R1	;	
	JC SetC	;	
	CMP R3,#0	;	
	JEQ Skip	;	
	ADD R0,#1	;	
	MOV R3,#0	;	
	JMP Skip	;	
SetC	CMP R3,#1	; (C1)	
	JEQ Skip	; (C2)	
	ADD R0,#1	; (C3)	
	MOV R3,#1	; (C4)	
Skip	?	; Decrease the value in register R2 by 1	(I6)
	?	; Jump back to Loop if value in R2 is still more than zero	(I7)
Exit	?	; Restore saved registers from the stack	(I8)
	?	; Return from subroutine	(I9)

Address	Contents
N1 0x100	0x031
A1 0x101	0xBFE

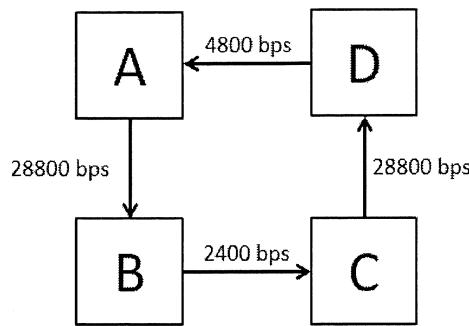
Memory Variables

**Figure Q2**

3. (a) You are tasked to design a Smart Home Controller with the following specifications
- 10-inch tablet computer with very slim 7mm profile
  - Long Run-time Operation Battery Life of 7 days
  - 16GByte of DRAM as System Memory to run the Android OS and Smart Home Applications
  - Android and Smart Home Application occupies 10GByte of the system memory.
  - Main application processor with 1MByte cache, 2GByte on-chip NOR flash memory and 32MByte on-chip SRAM.
- (i) With such a huge system memory, why do we still need storage memory for this system? (2 marks)
- (ii) Explain clearly what type of memory we should choose for the storage memory. (2 marks)
- (iii) Suggest a reason why NOR instead of NAND flash is used for the processor on-chip flash. (2 marks)
- (iv) Explain why the processor still needs the 1 MByte cache when it has a much larger 32MByte on-chip SRAM. Both of these memories are fast internal volatile memory. (2 marks)
- (b) Figure Q3a shows 4 processors (A, B, C and D) connected via UART link.
- All the UART links have the same configuration of 1 Start Bit, 7 Data bits, 1 Stop bit and Even Parity.
  - Baud rate of the UART link between each processor pair is different and is shown in Figure Q3a.
  - Each processor has two UART peripherals for independent connection to two processors. The baud rate of the UART is configured according to the baud rate of the link it is connected to.
  - Processor A sends a text string “12345678” to Processor B, the text string gets transferred from B to C to D and back to A via their UART connection.
  - Assume that there is a small input buffer in each processor’s UART receive program to store incoming UART data.

Note: Question No. 3 continues on Page 5

- (i) What possible text string will processor A receive? Would processor A receive only valid ASCII characters or would it receive some non-ASCII character data as well? (3 marks)
- (ii) Explain clearly how the answers in (i) above is derived and what happen in each processor during the transmission of the text string. (5 marks)



**Figure Q3a**

- (c) Consider a system with the following characteristics.
- Direct mapped cache of 16 cache blocks and block size 16 bytes
  - Cache uses Virtual Address for address mapping
  - Virtual Memory page size 1024 bytes
  - Virtual Memory size 1Mbyte. Physical Memory size 64 KByte
  - Extracts of Page Table (valid entries)
    - Virtual Page 0 → Physical Frame 2
    - Virtual Page 1 → Physical Frame 5
    - Virtual Page 2 → Physical Frame 8
    - Virtual Page 9 → Physical Frame 3

From which cache block would the CPU retrieve the data for a virtual address 0x00119? What is the corresponding tag value of the cache block if it is a cache hit?

(5 marks)

Note: Question No. 3 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 is enabled
  - This processor is not the VIP processor
- (i) Identify and describe the pipeline conflicts when executing the code in Figure Q3b (3 marks)
- (ii) Suggest how the conflicts in (i) can be resolved. (3 marks)
- (iii) For the code in Figure Q3b, how many cycles are saved by using the delayed branching feature? (3 marks)

```

    MOV  AR, #10      ; I1
    MOV  R1, #200     ; I2
    MOV  R0, #100     ; I3
Loop ADD  R2, [R0]   ; I4
        SUB  R2, [R1]   ; I5
        JDAR Loop      ; I6
        INC  R1          ; I7
        INC  R0          ; I8
    MOV  [R0], R2      ; I9

```

**Figure Q3b**

4. Please answer all 10 Multiple Choice Questions below. Each question carries 2 marks.
- (a) Which of the following statement below is FALSE?
- A. Difference in electrical signal level of two connected devices may result in safety issues.
  - B. For two devices to communicate, both VOH and VOL of transmitter has to be higher than VIH and VIL of receiver respectively.
  - C. Differential signals have higher noise tolerance because external interference has similar influence on the V+ ad V- line.
  - D. VOH of a device is always higher than VIH of the same device.
  - E. None of the above
- (b) Which of the following scenario will not increase the amount of cross talk between the wires?
- A. Longer wire length
  - B. Higher electrical current in wires
  - C. Smaller ground plane below the wires
  - D. Transmission done in single-ended rather than differential mode.
  - E. None of the above
- (c) Processor X receives an interrupt request from Device A at time  $t = 1\text{s}$ , X proceeds to do some register saving and looks up the interrupt service routine (ISR) address at  $t = 5\text{s}$ . The ISR executes at  $t = 7\text{s}$ , completes at  $t = 12\text{s}$  and program returns to main routine at  $t = 16\text{s}$ . What is the interrupt latency of processor X?
- A. 4s
  - B. 6s
  - C. 7s
  - D. 11s
  - E. None of the above
- (d) Which of the following will not affect the DMA data transfer rate?
- A. DMA mode of transfer
  - B. System bus utilization rate of CPU
  - C. DMA Destination device's maximum transfer rate
  - D. DMA Source device's address range
  - E. None of the above

Note: Question No. 4 continues on Page 8

- (e) NOR flash supports random byte programming but can only erase at block level. A NOR flash memory contains 0x55, 0xAA and 0x34 in address location 0x1, 0x2 and 0x3 respectively, all within the same block. Is it possible to program just the location 0x2 with a value of 0x1A?
- A. Yes. NOR flash is able to support random byte programming so there is no need to erase the block before programming location 0x2.
  - B. Yes. But user need to save location 0x1 and 0x3 to somewhere else first, erase the whole block and re-write location 0x1, 0x2 and 0x3 with the new values.
  - C. No. Flash programming can only change a memory bit from ‘1’ to ‘0’ and not ‘0’ to ‘1’ so it is not possible to change 0xAA to 0x1A in location 0x2
  - D. No. NOR flash does not support random byte programming. It is NAND flash that supports random byte programming.
  - E. None of the above
- (f) Given the same cache size and cache block size, what is the difference between direct mapped and set associative cache?
- A. Direct mapped cache has more cache blocks
  - B. Set associative cache has more cache blocks
  - C. Direct mapped cache has a shorter search time
  - D. Set associative cache has a shorter search time
  - E. None of the above
- (g) In a system with 1 KByte page size, if the virtual page 0, 1, 2 is mapped to physical frame 3, 2, 1 respectively, what physical address will virtual address 0x0200 be translated to?
- A. 0x0100
  - B. 0x0300
  - C. 0x0E00
  - D. 0x3200
  - E. None of the above

Note: Question No. 4 continues on Page 9

- (h) If the Mean time between failures (MTBF) of a product is 1 million hours, what is the probability that the product will still be functioning after 2 years?
- A. 0.983
  - B. 0.991
  - C. 0.999
  - D. 1.0
  - E. None of the above
- (i) Give that the format of IEEE754 number is  $(-1)^S * (1.F) * 2^{E-\text{Bias}}$ . What is the smallest positive number that can be represented for a single precision IEEE754 number where S, F and E are allocated 1, 23 and 8 bits respectively? Bias = 127 for single precision.
- A. 0
  - B.  $2^{-126}$
  - C.  $2^{-127}$
  - D.  $2^{-255}$
  - E. None of the above
- (j) Which of the following parameters does not directly affect the computational performance of a computer system?
- A. CPU frequency
  - B. System bus bandwidth
  - C. Presence of graphics accelerator
  - D. Battery Capacity
  - E. More than one correct answers

(20 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	$d \leftarrow d + (\text{NOT. } s) + 1$	VNZC
7	CMP	d	s	$d + (\text{NOT. } s) + 1$	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	$d \leftarrow \text{NOT. } d$	NZ
9	NEG	d	$d \leftarrow (\text{NOT. } d) + 1$	NZC
A	DADD	s	$AR \leftarrow AR + s + \text{carry}$ (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	$SR(\text{bits } 3-0) \leftarrow SR . AND. (\text{NOT. } n)$	explicit
F	BSSR	n	$SR(\text{bits } 3-0) \leftarrow SR . OR. n$	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	$SP \leftarrow SP-1; (SP) \leftarrow \text{Return Address}$ $PC \leftarrow \text{Effective address}$	
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



**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.