NANYANG TECHNOLOGICAL UNIVERSITY

SEMESTER 2 EXAMINATION 2015-2016

CE1006/CZ1006 – COMPUTER ORGANIZATION AND ARCHITECTURE

Apr/May 2016 Time Allowed: 2 hours

INSTRUCTIONS

1. This paper contains 4 questions and comprises 8 pages.

2. Answer **ALL** questions.

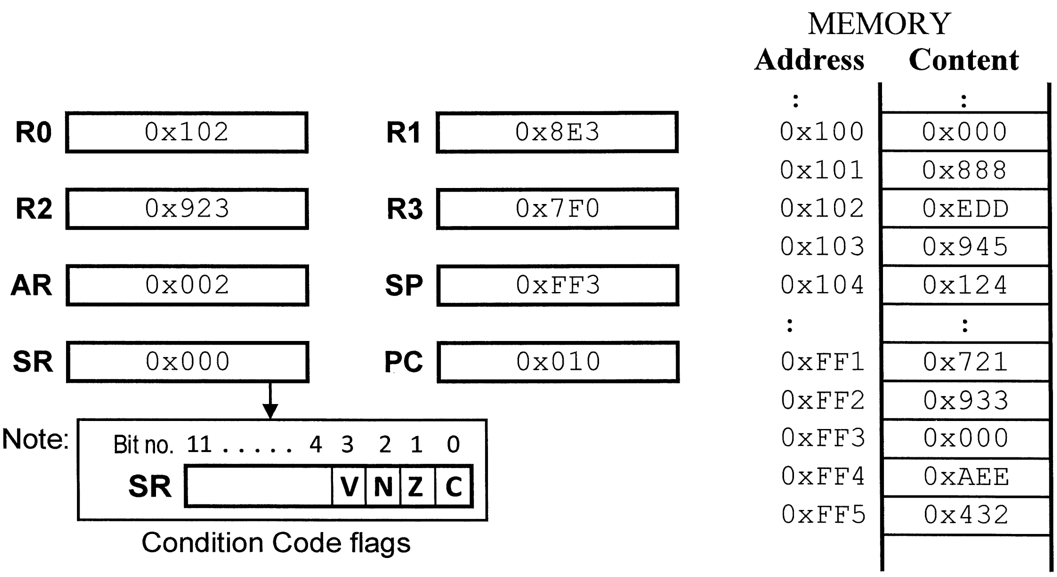
3. This is a closed-book examination.

4. All questions carry equal marks.

5. The VIP Instruction Set Summary Chart is provided in Appendix 1 on page 8.

---

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



R0	0x102	R1	0x8E3
R2	0x923	R3	0x7F0
AR	0x002	SP	0xFF3
SR	0x000	PC	0x010

MEMORY

Address	Content
:	:
0x100	0x000
0x101	0x888
0x102	0xEDD
0x103	0x945
0x104	0x124
:	:
0xFF1	0x721
0xFF2	0x933
0xFF3	0x000
0xFF4	0xAEE
0xFF5	0x432

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,R3**

(ii) **MOV R1,[R0]**

(iii) **OR R1,#0x104**

(iv) **ADD R1,[SP+0xFFF]**

(v) **RAR R1**

(10 marks)

(b) A VIP assembly language program is given in Figure Q1b. Using the initial conditions shown in Figure Q1a and relevant information in Appendix 1, give (*in hexadecimal*) the 12-bit contents of the **seven** registers **R0**, **R1**, **R2**, **R3**, **PC**, **SP** and **SR after** the execution of the last instruction at the label **FINISH**. Assume the execution of the given series of consecutive instructions begins at the label **START**.

(11 marks)

```
START       MOV   R3,#2
            MOV   R1,#0x103
            MOV   R2,#2
            MOV   R0,[PC]
LOOP        ADD   R2,R3
            SUB   R3,#1
            JPL   LOOP
            MOV   [0x103],R3
FINISH      RET
```

Figure Q1b

(c) Re-write the code segment given in Figure Q1b so that the program will be optimized. You must ensure that the result produced by the optimized code is identical to that of the original version. (Marks will be awarded based on how well you have optimized the given code segment for both execution speed and code size.)

(4 marks)

2. (a) Describe clearly the difference between passing parameter by reference and by value. State how the parameter **VarX** is passed by the instruction (**I1**) in the VIP assembly program in Figure Q2a.

(5 marks)

(b) Give the VIP mnemonics at (**I2**) and (**I3**) in Figure Q2a to complete the missing instructions based on their associated comments.

(4 marks)

```
Main    :      :
        PSH    #0x100  ; Pass parameter to subroutine SubX          (I1)
        CALL   SubX    ; Call subroutine SubX
        ?              ; Remove stack parameter                     (I2)
        :
        :
SubX    PSHM   15      ; save registers
        :          ⎫   ; to be completed as
        :          ⎭   ; required in question Q2(c)
        ?              ; Return to calling program                  (I3)
```

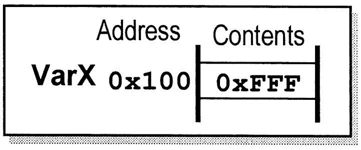


Figure Q2a

(c) The VIP subroutine **SubX** computes the absolute value of a 12-bit signed parameter passed to it. For example, the negative value **0xFFF** in memory variable **VarA** will be converted to the value **0x001** on returning from **SubX**. Complete the partially written subroutine **SubX** shown in Figure Q2a based on the functionality described and the manner in which the parameter has been passed.

(10 marks)

(d) The C statement in Figure Q2b sets the memory variable **VarA** to **TRUE** (i.e. value of 1) if its current value is a *negative odd number*, otherwise it clears **VarA** to **FALSE** (i.e. value of 0). Using **branchless logic**, implement the equivalent VIP assembly code. You may assume the absolute address of variable **VarA** is given by the label **VarA** and you may use any number of registers in your implementation.

(6 marks)

```
if ((VarA < 0) && ((VarA % 2) == 1)
    VarA = TRUE;
else
    VarA = FALSE;
```

Note: The modulus operator "%" produces the remainder after an integer division

Figure Q2b

3. (a) Determine the most suitable memory to be used for each of the requirements listed in Table Q3a below. 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.

Table Q3a

Requirement	Memory Type
Cache Memory to allow very fast access of recent data by CPU	
System memory in the computer system and needs to be able to update and erase every byte individually	
Main data storage in Data Center for cloud storage services	

(6 marks)

(b) With reference to the structure of a DRAM, explain clearly why there is a need to perform refresh operations when using DRAM to store data.

(4 marks)

(c) Give a brief description of the two factors that limit the speed in which a parallel bus can operate.

(4 marks)

(d) Figure Q3 shows two processor systems connected via a UART link.

- System B transmits sensor data continuously over the UART to System A. One UART packet is transmitted with each call to the UART transmit routine in CPU-B.

- Each time System A's UART peripheral receives a packet of UART data, it will send an interrupt to CPU-A, which will then proceed to read the data in an interrupt service routine (UART_RX_ ISR).

Table Q3b shows the various timings and UART parameters of the two systems. Instruction cycle time refers to the time needed to execute one instruction.

Note: Question No. 3 continues on Page 5

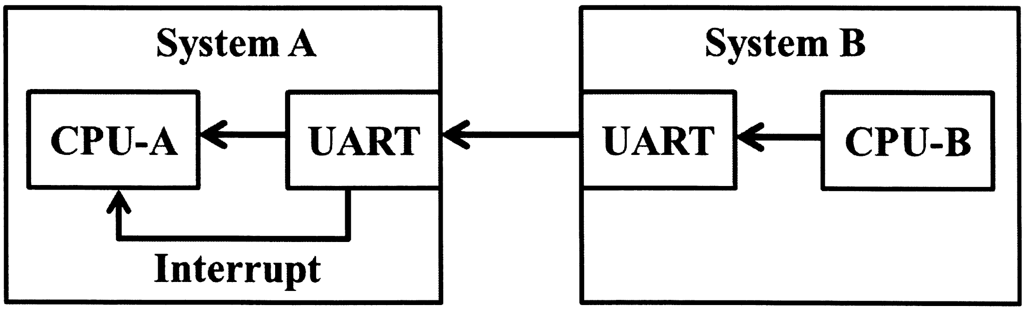


Figure Q3

Table Q3b

System A	System B
CPU Interrupt Latency = 10 instruction cycles	CPU Interrupt Latency = 5 instruction cycles
Number of instructions in UART_RX_ISR = 40	Number of instructions in UART transmit routine = 20
Instruction cycle time = 10µs	Instruction cycle time = 20µs
UART Configuration: 1 Start Bit, 7 data bits, Odd Parity, 1 STOP bit	
Allowable UART baud rate: 4800, 9600, 14400, 19200, 28800, 38400, 57600, 115200	

With reference to Figure Q3 and Table Q3b, answer the following questions:

(i) In order to achieve the maximum transfer rate, the UART packets are being transferred one after another without any delay between the packets. Calculate and explain whether the system in Figure Q3 will be able to support such transfers at a baud rate of 19200.

(6 marks)

(ii) Draw the timing diagram of the UART transmission when the data is an ASCII character of value 0x21. Label clearly the Start, Stop, Parity bits and the value of each data bits (‘1’or ‘0’).

(3 marks)

(iii) If delays between each UART packet transmission are allowed, what would be the maximum baud rate that can be used? Choose from the allowable UART baud rate in Table Q3b and explain clearly the reason for choosing the specified baud rate.

(2 marks)

4. (a) List one advantage and one disadvantage for the following

- Magnetic Hard Disk Drive
- Solid State Drive

(4 marks)

(b) Consider a computer system with the characteristics given below, what is the effective access time of this hierarchical memory system?

- Main memory access time = 50 ns
- Cache and main memory access do not overlap
- Cache parameters
  - o Cache access time = 5 ns
  - o Average cache hit rate = 90%

(4 marks)

(c) One version of the VIP code for multiplication is shown in Figure Q4a.

(i) Comment on the potential issue in using this code to multiply two 12-bit operands.

(2 marks)

(ii) Modify the code in Figure Q4a to perform multiplication of two 12-bit operands correctly. State all assumptions made.

(6 marks)

```
           MOV   AR, #12      ; I1
loop       ROR   R2           ; I2
           JNC   skip         ; I3
           ADD   R0, R3       ; I4
skip       BCSR  1            ; I5
           RLC   R3           ; I6
           JDAR  loop         ; I7
```

Figure Q4a

Note: Question No. 4 continues on Page 7

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

- Branch target address is calculated at the execute stage.
- Instruction length for every instruction is one word long.
- Each pipeline stage take 1 cycle to complete.
- Figure Q4b shows a code segment which performs accumulation of 10 sets of data.
- **NOP** refers to "No Operation" instruction.
- Note that this processor is <u>NOT</u> the VIP processor.

(i) How many cycles does it take to execute the code in Figure Q4b? Explain clearly how the answer is derived.

(4 marks)

(ii) Assume that the processor supports delayed branching, modify the code in Figure Q4b to fill up the delay slot(s) with proper instructions.

(2 marks)

(iii) Explain clearly how the code in Figure Q4b results in less efficient code and how the changes made to the code in Q4(d)(ii) help to improve the efficiency.

(3 marks)

```
      MOV   AR, #10     ; I1
      MOV   R0, #200    ; I2
      MOV   R1, #0      ; I3
      NOP               ; I4
loop  ADD   R1, [R0]    ; I5
      INC   R0          ; I6
      JDAR  loop        ; I7
      MOV   R2, R0      ; I8
      MOV   R3, R1      ; I9
```

<u>Figure Q4b</u>

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.