NANYANG TECHNOLOGICAL UNIVERSITY

SEMESTER 1 EXAMINATION 2016-2017

CE1006/CZ1006 – COMPUTER ORGANIZATION AND ARCHITECTURE

Nov/Dec 2016 Time Allowed: 2 hours

INSTRUCTIONS

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

2. Answer **ALL** questions.

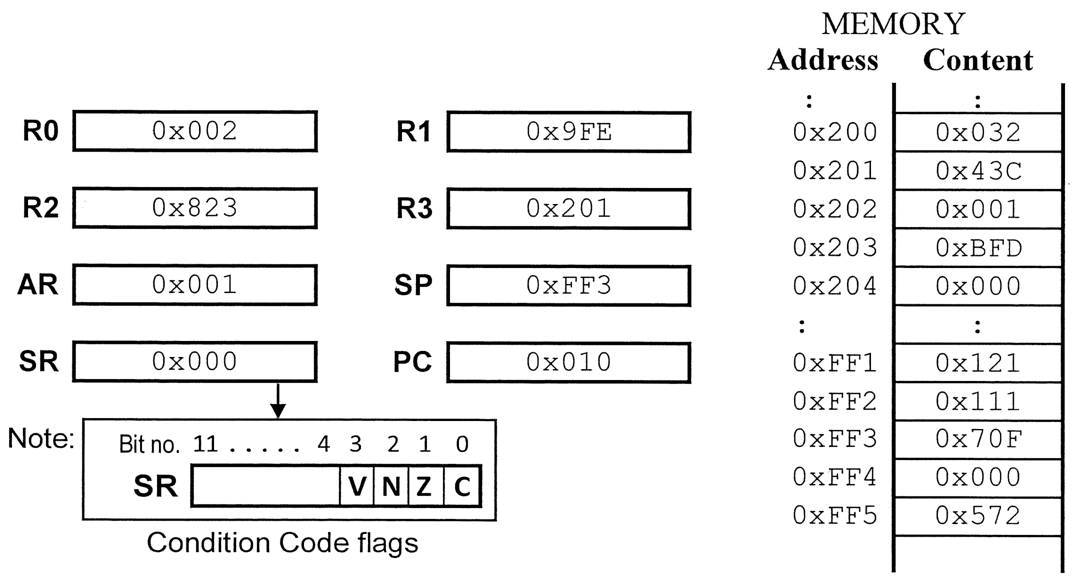
3. This is a closed-book examination.

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.



Address	Content
:	:
0x200	0x032
0x201	0x43C
0x202	0x001
0x203	0xBFD
0x204	0x000
:	:
0xFF1	0x121
0xFF2	0x111
0xFF3	0x70F
0xFF4	0x000
0xFF5	0x572

Figure Q1a

Note: Question No. 1 continues on Page 2

(a) Give (*in hexadecimal*) the 12-bit contents in the two registers **R2** 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 R2,#0xFF1**

(ii) **MOV R2,[R3+3]**

(iii) **EOR R2,[0xFF2]**

(iv) **SUB R2,R1**

(v) **POPM 5**

(10 marks)

(b) A VIP assembly language program is given in Figure Q1b. Answer the questions below using the initial conditions shown in Figure Q1a and relevant information in Appendix 1.

```
START      POP   R2
           BSSR  1
LOOP       RLC   R1
           ADD   R0,#0xFFF
           JNE   LOOP
MIDWAY     ADD   R3,R2
NEXT       MOV   PC,#0x201
           PSH   R1
           SUB   R3,R3
FINISH     RET
```

Figure Q1b

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

(8 marks)

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

(7 marks)

2. (a) An incomplete VIP assembly language program is given in Figure Q2a. Give the mnemonics of (`I1`) to (`I6`) that will complete the missing instructions based on their associated comments. Assume memory variables **N** and **ANS** are in addresses `0x100` and `0x101` respectively.

(9 marks)

```
Main     :
       MOV  [0x101],#0     ; Clear memory variable  Ans for cumulation
         ?                 ; Push value in memory variable N to stack         (I1)
         ?                 ; Push address of memory variable Ans to stack     (I2)
       CALL  SumN          ; Call subroutine SumN
         ?                 ; Remove parameters on the stack                   (I3)
         :
         :
SumN   PSHM  0x003         ; Save away used registers
         ?                 ; Retrieve the value of N from stack into R0       (I4)
         ?                 ; Retrieve the address of Ans from stack into R1   (I5)
         :              ⎫  ;
       CALL   SumN      ⎬  ; to be completed as required in question Q2(b)
         :              ⎭  ;
         ?                 ; Return to calling program                        (I6)
```

Figure Q2a

(b) Subroutine `SumN` sums the positive numbers from 1 to **N**, where **N** is a value passed to the subroutine. The result is stored in the memory variable **ANS**, whose address is also passed to the subroutine as shown in Figure Q2a. For example, if the value of **N** is 3, then the value 6 given by the summation (3+2+1) is placed into the memory variable **ANS**. Complete the incomplete subroutine `SumN` shown in Figure Q2a using a **recursive** technique to provide the described requirement.

(10 marks)

(c) Write the equivalent VIP assembly language code that represents the C high-level language code segment given in Figure Q2b. You may assume the absolute address of memory variables `Var1` and `Var2` are given by the address labels `Var1` and `Var2` respectively.

(6 marks)

```
while (Var1 < Var2) {
   Var2 = Var1 - Var2;
   }
```

Figure Q2b

3. (a) Determine the most suitable memory to be used for each requirement 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.

(6 marks)

Table Q3a

Requirement	Memory Type
Memory to implement Translation Lookaside Buffer.	
64 GByte USB Thumb Drive	
Storage memory for a Network Access Server used to handle local cloud and backup functions at home.	

(b) Explain and illustrate with the logic diagram of a SRAM cell, how digital data is stored in a typical SRAM cell. What is the use of the pass transistors in the SRAM cell?

(4 marks)

(c) Give a brief description of the two requirements that need to be met before two electronic components are able to interface and communicate with each other.

(4 marks)

(d) Figure Q3 shows two identical processor systems (A1 and A2) connected via a UART link.

- System A1 transmits data to System A2 via a UART connection. One UART packet is transmitted with each call to the UART transmit routine in the CPU.

- Each time System A2's UART peripheral receives a packet of UART data, it will send a trigger signal to the DMAC, which will then proceed to transfer the data from the UART data register to the system memory.

Note: Question No. 3 continues on Page 5

Information of the processor system is listed below. Instruction cycle time refers to the time needed to execute one instruction.

- Fetch and Deposit DMA
- DMA mode supported: Burst, Cycle-Stealing
- Time needed to transfer system bus control between CPU and DMAC = 100 μs
- System Bus speed = 50 μs
- Minimum delay between DMAC bus requests (Cycle-Stealing) = 500μs
- CPU instruction cycle time = 20 μs
- Number of instructions in UART transmit routine = 25
- UART configuration: 1 Start, 8 Data, 1 Stop Bit and No Parity bit
- Allowable UART baud rate: 4800, 9600, 14400, 19200, 28800, 38400, 57600, 115200.

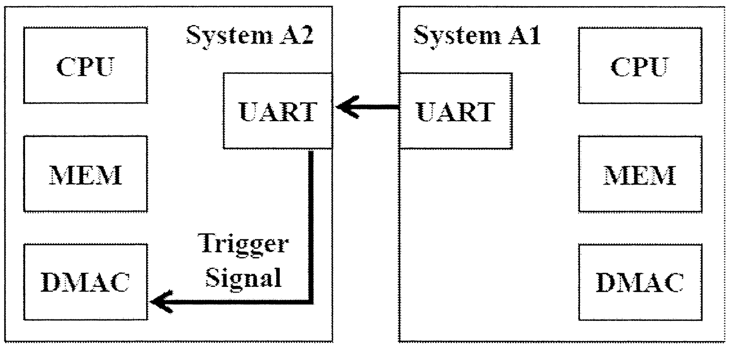


Figure Q3

With reference to Figure Q3 and given that the UART packets are being transferred one after another without delay between packets, answer the following questions:

(i) What is maximum baud rate allowed on the UART connection if the DMAC uses cycle stealing mode?

(6 marks)

(ii) Ignoring the bus control transfer time, what would be the maximum baud rate allowed on the UART connection if the DMAC uses burst mode?

(2 marks)

(iii) A parity scheme is not used in the UART transfer described above, what potential issue would this cause?

(3 marks)

4. (a) Describe what is wear levelling and explain why wear levelling is needed for solid state drives.

(4 marks)

(b) Consider a computer system with the following characteristics.

- Cache uses physical address to derive its block and tag field
- Main memory access time = 50 ns
- Translation Lookaside Buffer (TLB)
  - TLB access time = 10 ns
  - Average TLB hit rate = 80%
- TLB and Page Table access do not overlap
- Cache
  - Cache access time = 5 ns
  - Average cache hit rate = 90%
- Cache and Main memory access do not overlap

Starting from virtual addresses of code and data, what would be the effective access time of this hierarchical memory system?

(6 marks)

(c) Given a set of 8-bit unsigned fixed-point numbers each with a different number of fractional bits, i.e. their binary points are at different position. The numbers are to be computed with an algorithm consisting of addition, subtraction, multiplication and division operations. The result needs to be rounded to a fixed-point integer number. List a set of rules to follow in order to achieve the best accuracy in the result. Explain clearly the rationale behind the rules.

(6 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 take 1 cycle to complete
- This processor is not the VIP processor

(i) Describe three types of pipeline conflict that will lower the performance of the processor system above.

(3 marks)

Note: Question No. 4 continues on Page 7

(ii) Figure Q4 shows a code segment which performs accumulation of 10 sets of data. Based on the given processor characteristics, explain clearly the types of pipeline conflict that will occur in the code segment and how these conflicts can be resolved.

(6 marks)

```
      MOV   AR, #5      ; I1
      MOV   R0, #0x200  ; I2
      MOV   R1, #0x100  ; I3
Loop  ADD   [R0],[R1]   ; I4
      INC   R1          ; I5
      JDAR  Loop        ; I6
      MOV   R2, [R0]    ; I7
      MOV   R3, R1      ; 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.