# NANYANG TECHNOLOGICAL UNIVERSITY

## SEMESTER 1 EXAMINATION 2013-2014

## CE1006/CZ1006 – COMPUTER ORGANIZATION AND ARCHITECTURE

Nov/Dec 2013 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 A.

---

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

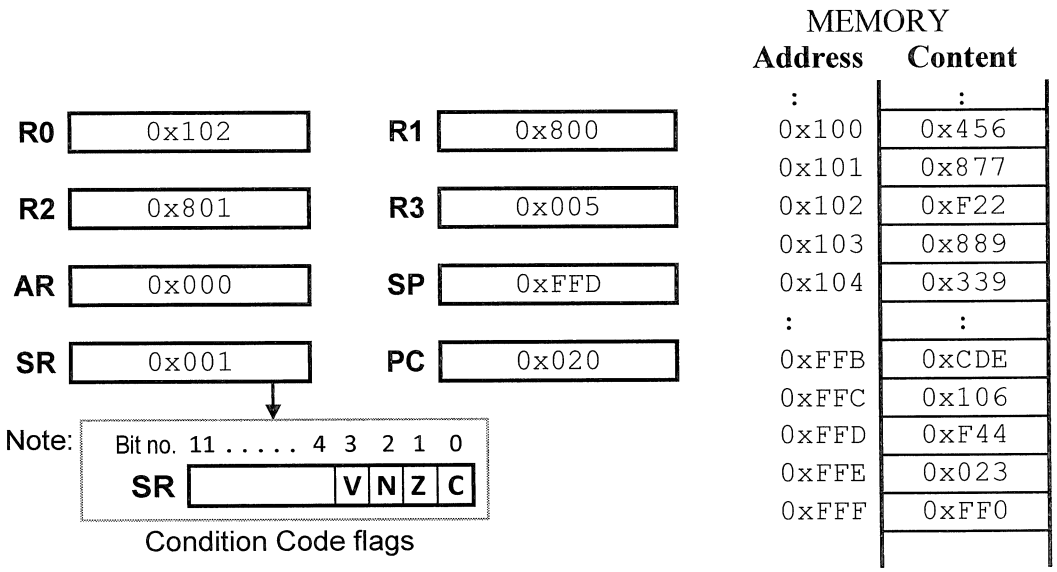


**Figure Q1a**

Note: Question No. 1 continues on Page 2

(a) Give (*in hexadecimal*) the 12-bit contents in the two registers **R1** and **SR**, *immediately after* the execution of each instruction given below.

**Note:** Instructions (i) to (vi) are **not consecutive instructions**. You must use the initial conditions shown in Figure Q1a to derive your answer for each of the instruction given below.

(i) `MOV R1,#0`

(ii) `MOV R1,[R0]`

(iii) `ADD R1, R2`

(iv) `RLC R1`

(v) `AND R1,[0x103]`

(vi) `POP R1`

(12 marks)

(b) Write the equivalent C high-level language program that is represented by the VIP assembly language program given in Figure Q1b. You may assume that your C integer variables **Var0**, **Var1**, **Var2**, **Var3** and **Var4** are represented by the memory addresses given by labels **Var0**, **Var1**, **Var2**, **Var3** and **Var4** respectively in Figure Q1b.

```
Start     MOV   [Var4],#0
          CMP   [Var2],[Var3]
          JGE   Label_1
          MOV   [Var1],#6
          JMP   Label_2
Label_1   MOV   [Var1],#4
Label_2   MOV   [Var0],#10
Label_3   CMP   [Var0],#0
          JEQ   Label_4
          ADD   [Var4],[Var1]
          DEC   [Var0]
          JMP   Label_3
Label_4   MOV   [Var2],[Var4]
```

**Figure Q1b**

(10 marks)

(c) Give a reason why some instruction set architecture (ISA) designs have adopted the Load-Store architecture?

(3 marks)

2. (a) With reference to the VIP assembly language program shown in Figure Q2, describe whether the parameter for memory variable **Data1** is passed to the subroutine `OddParity` by value or by reference. Give a reason for your answer.

(4 marks)

```
Address
0x000  Main  MOV   SP,#0xFFF   ; Initialize stack pointer          (I1)
0x002        PSH   #0x100      ; Push parameter to stack          (I2)
0x004        CALL  OddParity   ; Call subroutine to insert odd parity bit (I3)
0x006        ?                 ; Remove stack parameter           (I4)
0x008        :
```

| | Address | Contents |
|---|---|---|
| Data1 | 0x100 | 0x111 |
| | 0x101 | : |
| | 0x102 | : |

Before Entering Subroutine

| | Address | Contents |
|---|---|---|
| Data1 | 0x100 | 0x911 |
| | 0x101 | : |
| | 0x102 | : |

After Returning from Subroutine

**Figure Q2**

(b) To pass an odd parity error check, the number of binary ones in the 12-bit data word must be odd. To ensure this happens, all occurrences of binary ones in the least significant 11 bits in the word are counted and if the tally is an even number, the most significant bit (MSB) in the word is set to one, otherwise it is cleared to zero.

Write a VIP assembly language subroutine labeled `OddParity` to perform the odd parity encoding to a 12-bit memory variable. Figure Q2 shows how the single parameter is to be passed into the subroutine and the effect of the `OddParity` subroutine upon returning to the calling program. Your solution must ensure the subroutine is transparent.

(16 marks)

(c) With reference to Figure Q2, give two different VIP mnemonics that can be used to remove the parameter from the stack. Explain clearly which is the preferred choice and why this is so.

(5 marks)

3. (a) Figure Q3a shows a 6-T **SRAM** cell.

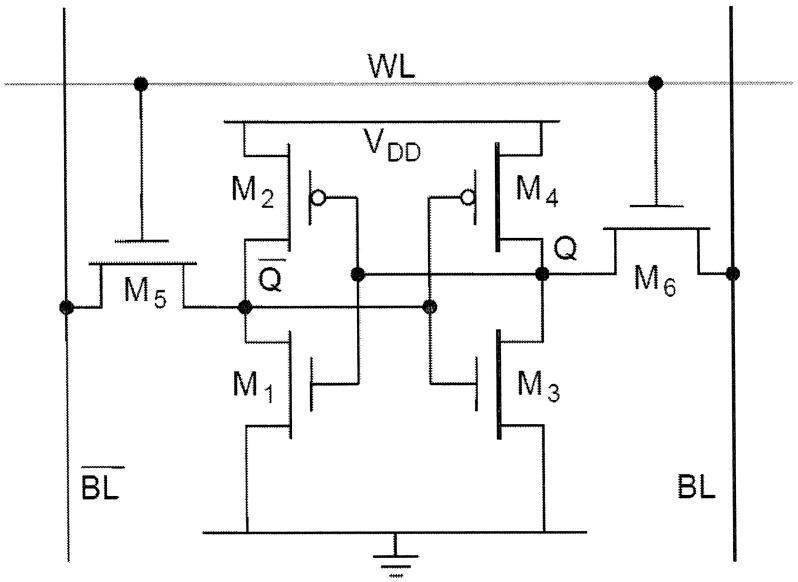


**Figure Q3a**

(i) Explain whether the SRAM cell is **volatile** or **non-volatile**.

(3 marks)

(ii) An SRAM device has 20-bit address bus and 16-bit data bus. How many memory locations and SRAM cells are there?

(4 marks)

(iii) Briefly describe the operations of the circuits in Figure Q3a for READ and WRITE access of the SRAM cell.

(4 marks)

(b) SRAM memory is often implemented as **cache memory** in computer systems.

(i) What is **cache memory** and what are its main purposes?

(4 marks)

Note: Question No. 3 continues on Page 5

(ii) An observation table for several direct-mapped cache memory accesses is shown in Table Q3b. The address reference strings are provided as hexadecimal numbers and the order of memory access starts from **3F8** and ends at **2EA.**

For each of these address reference strings,

- give the value of the binary bits in the Tag, Block and Word fields.
- state whether each cache memory access registers a hit or a miss, and if the cache content is replaced.

Assume the cache memory is initially empty. You may use a table similar to Table Q3b to present your answers.

(10 marks)

**Table Q3b**

| Direct-Mapped Cache Memory | | | | | |
|---|---|---|---|---|---|
| Address Reference String (Hex) | Main Memory Address | | | Cache Access (Hit/Miss?) | Cache Replace (Yes/No?) |
| | Tag (2 bits) | Block (4 bits) | Word (4 bits) | | |
| **3F8** | | | | | |
| **3F3** | | | | | |
| **3E2** | | | | | |
| **2E2** | | | | | |
| **2EA** | | | | | |

4. (a) An engineer would like to transfer data periodically from a data logger to a computer using direct interfacing.

(i) Should he consider using serial or parallel data transfer? List three key considerations and discuss their respective impact on his decision.

(9 marks)

He eventually decides to use the RS-232 protocol shown in Figure Q4. The clock period used in the data transfer is 1 μs. With reference to this decision, answer the following questions:

Note: Question No. 4 continues on Page 6

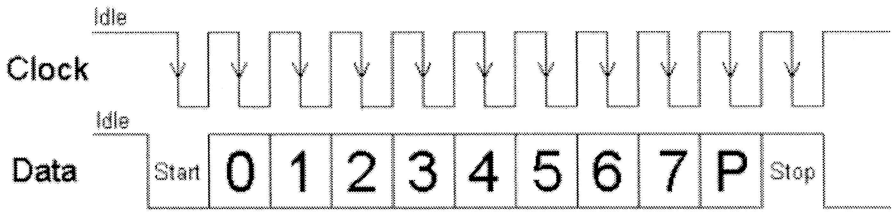


**Figure Q4**

(ii) What is the theoretical maximum data transfer rate?

(4 marks)

(iii) If the data logger needs to send 1 megabyte (MB) of data every 10 ms, discuss whether the engineer has made a correct choice in selecting the RS-232 protocol.

(4 marks)

(b) Using any of the **Rotate** instructions in the VIP Instruction Set given in Appendix A, write a VIP assembly code segment that will multiply the unsigned content in register **`R0`** by the constant value 3. Your 12-bit result should reside in register **`R0`** after the multiplication.

(4 marks)

(c) It is suggested register **`R0`** may represent a real number with value 2.611 using fixed floating-point representation where the unsigned integer has 5 bits and the fractional value has 7 bits. Determine the 12-bit hexadecimal value that is stored in register **`R0`** and the smallest incurred rounding error.

(4 marks)

# Appendix A

## 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<br>PC ← Effective address | |
| 5 | RET | n | PC ← (SP) + n; SP ← SP+1 | |
| 6 | | | See subgroup 3a | |
| 7 | RCN | n | Count for next rotate instruction.<br>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;<br>PC ← (SP); SP ← SP+1 |
| 6 | SWI | 1 | SP ← SP-1; (SP) ← PC;<br>SP ← SP-1; (SP) ← SR; PC ← (0x009) |
| 6 | WAIT | 2 | IE ← 1;<br>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<br>(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.