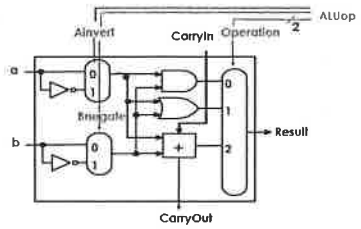
考試時間	月 日上午 下午第 節 （星期 ）晚間	份數		任課教師	

國立臺灣科技大學 107 學年度第 二 學期 博士班考試命題用紙 第 1 頁共 2 頁

考試科目：Computer Architecture

☐研究所 ☐大學部 ☐工程在職進修 系班別：資格考

1. The following figure is a basic implementation of 1-bit ALU. Please specify the 4-bit ALUop, {Ainvert, Bnegate, and Operation}, accordingly. **(A)** AND (2%), **(B)** ADD (2%), **(C)** SUB (2%), **(D)** NAND (2%).

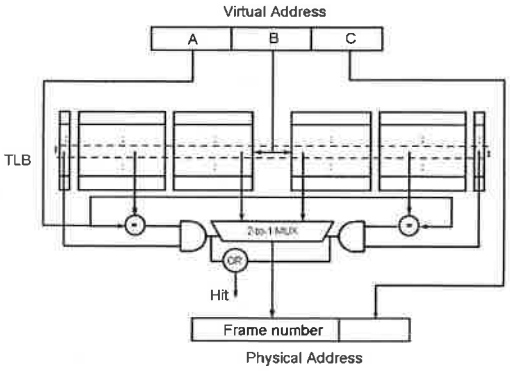


2. Compare the **(A)** performance (3%) and **(B)** hardware resource (3%) of the following adders: ripple carry adder (RCA), carry look-ahead adder (CLA), and carry select adder (CSA).

3. There are three well-known computer architecture challenges: (1) the power wall, (2) the memory wall, and (3) the instruction level parallelism (ILP) wall. **(A)** Discuss the impact of these challenges to a computer system? (6%) **(B)** How do we tackle these challenges? (6%)

4. Consider a pipelined MIPS processor implementation with instruction fetch (IF), instruction decode (ID), execution (EX), memory access (MEM), and write back (WB) stages. Explain the following hazards and discuss the techniques to tackle these hazards. **(A)** The structural hazard (4%), **(B)** the data hazard (4%), and **(C)** the control hazard (4%).

5. A virtual memory system with 32-bit virtual address, 128-entry 2-way set associative TLB, and 4KB page size. What are the sizes of fields **A** (4%), **B** (4%), and **C** (4%).



6. Multiple forms of addressing are generally called addressing modes. The MIPS addressing modes are the following:
(1) Register addressing, where the operand is a register.
(2) Base or displacement addressing, where the operand is at the memory location whose address is the sum of a register and a constant in the instruction.
(3) Immediate addressing, where the operand is a constant within the instruction itself.
(4) PC-relative addressing, where the address is the sum of the PC and a constant in the instruction.
(5) Pseudodirect addressing, where the jump address is the 26 bits of the instruction concatenated with the upper bits of the PC.

The following binary codes are corresponding to their MIPS instructions, respectively. **(A)** Indicate these two instructions belonging to which of the above addressing modes (4%) and **(B)** find binary codes of `"add $s4, $t3, $t2"` and `"lw $s0, 48($t1)"` (10%).

Hint:

```
add $t0, $s1, $s2    00000010 00110010 01000000 00100000
lw $t0, 32($s2)      10001110 01001000 00000000 00100000
```

7. The following program tries to copy words from the address in register `$a0` to the address in register `$a1` and count the number of words copied in register `$v0`. The program stops copying when it finds a word equal to 0. You do not have to preserve the contents of registers `$v1`, `$a0`, and `$a1`. This terminating word should be copied but not counted.

```
Loop:   lw $v1, 0($a0)
        addi $v0, $v0, 1
        sw $v1, 0($a0)
        addi $a0, $a0, 1
        addi $a1, $a1, 1
        bne $v1, $zero, loop
```

However, there are multiple bugs in this MIPS program. Please fix them and turn in bug-free version. (10%)

8. **(A)** When an equal is true, which one of the following two operations is performed for the MIPS assembly `"beq r1, r2, imm16"`? (3%) **(B)** Discuss the 3 differences between (1) and (2). (9%)
(1) `PC = PC + unsign_extension(Imm16)`?
(2) `PC = PC + 4 + sign_extension(Imm16)||00`$_b$?

9. Assume we have the following 3 ISA styles:
(1) Stack: All operations occur on top of stack where `PUSH` and `POP` are the only instructions that access memory;
(2) Accumulator: All operations occur between an Accumulator and a memory location;
(3) Load-Store: All operations occur in registers, and register-to-register instructions use 3 registers per instruction.
(A) For each of the above ISAs, write an assembly code for the following program segment using `LOAD`, `STORE`, `PUSH`, `POP`, `ADD`, and `SUB` and other necessary assembly language mnemonics. (9%)

```
A = A + C;
D = A - B;
```

(B) Some operations are not commutative (e.g., subtraction). Discuss what are the advantages and disadvantages of the above 3 ISAs when executing non-commutative operations. (5%)