Fall, 2022

Week 7

2022.10.24

組別：_____ 簽名：_____

[group1]

1. Please order the sequence of the following steps for MIPS to invoke the exception handler.
    a. Save PC in the exception program counter (EPC) register
    b. Use mfc0 instruction to retrieve/copy EPC value to a general-purpose register
    c. Jump to the predefined handler address
    d. Return after the corrective action using jr instruction

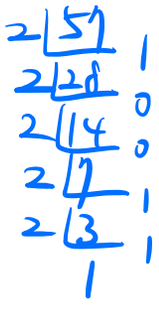
a → c. → b. → d.

[group2]

2. Let two decimal numbers A = 57, B = -84, please answer the following questions.

1. Compute A + B in 8-bit 2's complement
2. Compute A - B in 8-bit 2's complement

Does overflow occur in the answers of question (1) and (2)? Why or why not? Please answer and explain them individually.



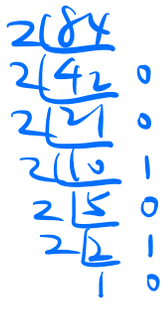
A = 57 = 00111001

B = -84 = 10101100

(1) A+B = 57-84 < -2^7

```
  00111001
+ 10101100
----------
  11100101
```

not overflow



+84 = 01010100

→ 01010011

-84 = 10101100

(2) A-B = 57+84 = 141 > $+2^7-1$

```
  00111001 → 57
+ 01010100 → 84
----------
 (1)00001101
```

∴ overflow

[group5]

3. Which of the following statements are true?

a. When we want to subtract b from a, we can add a to b', where b' is the action that 0 to 1,1 to 0 in every bit.

b. If we want to execute nor instruction, Ainvert , Bnegate, operation will be set to 1,1,01 respectively.

c. When detecting overflow, we can use one exclusive nor gate to complete the task.

d. It is expensive to build a fully carry look-ahead adder.

e. It is impossible to occur overflow when add a to b, where a is positive and b is negative.

[group12]

4. Please select the correct options

(a) When we implement 'set on less than' in ALU, the set value in ALU0 (LSB) is according to the result of adder in ALU31 (MSB) .

(b) We use 'XOR gate' to detect the carry into/out of most significant bit because the overflow occurs when the values in two bits are the same.

(c) 'Saturation' means that when a calculation overflows, the result is set to the largest positive number or the most negative number, rather than a modulo calculation as in 2's complement arithmetic.

(d) When we implement 'subtraction' in ALU, we add 'one' because of 2's complement.

[group7]

5. Choose the correct options

a. all the languages require raising an exception of overflow

b. some languages will save PC in exception program counter register when overflow

c. when adding operands with different signs, overflow may occur

d. when carry into MSB equal to carry out of MSB means overflow

e. add, addi will invoke exception handler during overflow

[group8] a, c, d.

6. Carry look-ahead adder can diminish the carry delay which dominates the delay of ripple carry adder. Generate (g_i) and propagate (p_i) functions are two main operations of carry look-ahead adder. Assume a and b are two operands and c_{i+1} is the carry out of level i and carry in of level $i+1$, which is (are) correct?

a. ✓ $g_i = a_i \cdot b_i$

b. ✗ $p_i = (a_i + b_i) \cdot c_i$ a_i XOR b_i

$C_{in(i+1)}$ $C_{oi} = g_i + \ldots$

c. ✓ If g_i equals to 1, we can say the carry out of level i is 1.

d. ✓ Carry look-ahead adder can be extended to multi-level style. The first group generate of a 3-bit group can then be defined as $G_o = g_2 + (p_2 \cdot g_1) + (p_2 \cdot p_1 \cdot g_0)$

hint : 2 bit group $G_o = g_1 + p_1 \cdot g_0$

[group14]

7. Suppose all numbers are 4-bits signed numbers, will the following formulas overflow?

(a) 0111 + 0100 = yes
(b) 0101 + 1011 = No
(c) 1000 + 1111 = yes
(d) 0010 + 0101 = No
(e) 1101 + 1100 = No

以2補數計算

0111 + 0100 = 1011

0101 + 1011 = 10000

1000 + 1111 = ~~0111~~

0010 + 0101 = 0111

1101 + 1100 = 11001

sign number → 7 ~ -7

(a) 0111 + 0100 = 1011 → overflow (7 + 4 = -5 + (-3))
(b) 0101 + 1011 = 0000 → no overflow (5 + (-5) = 0 + 0)
(c) 1000 + 1111 = 1111 → no overflow (-8 + (-1) = -9)
(d) 0010 + 0101 = 0111 → no overflow (2 + 5 = 7)
(e) 1101 + 1100 = ↵ → overflow (-5 + (-4) = -9)

以sign number計算