國立中央大學九十二學年度碩士班考試入學招生試題卷　　共 2 頁　第 1 頁

系所別：　資訊工程學系　　科目：　計算機結構

網路學習科技研究所

1. 詳細解釋下列名詞或回答下列問題　(每小題 5 分，　合計 25 分)

(a) Write down the IEEE 754 representation (in hex format) for the value "–13.125"

(b) What is the advantage of two's complement representation when compared with signed-magnitude representation?

(c) Given an n-bit 2's complement representation ($x_{n-1}\ x_{n-2}\ \ldots.\ x_1\ x_0$). What's the value that it represents (write down the equation but NOT any specific example)?

(d) Write down the Boolean equation of testing overflow in n-bit 2's complement addition.

(e) What is CPI?

2. About nanoprogramming technique:

(a) Sketch and explain the block diagram of nanoprogramming. (10%)

(b) Explain the reason why people use nanoprogramming for CISC processor control design. (5%)

3. For the hierarchical "carry-lookahead" adder design, please write down the Boolean equations for the following signals. (10%)

(a) 16-bit "Group Propagate" based on 4-bit Propagate Pi and 4-bit Generate Gi (i=0,1,2,3).

(b) 16-bit "Group Generate" based on 4-bit Propagate Pi and 4-bit Generate Gi (i=0,1,2,3).

4. Assume there is a 4KB cache with set-associative address mapping. And the cache is partitioned into 32 sets with 4 blocks in each set. The memory-address size is 32 bits, and the smallest addressable unit is byte.

(a) To what set of the cache is the address $000010AF_{16}$ assigned? (10%)

(b) If the addresses $000010AF_{16}$ and $FFFF7xyz_{16}$ can be as assigned to the same cache set, what values can the address digits xyz have? (10%)

注意：背面有試題

系所別:　資訊工程學系　　科目:　計算機結構
　　　　網路學習科技研究所

5.

(a) What kinds of hazard may occur on the pipeline architecture? (5%)

(b) Consider a pipeline machine with 5 stages (instruction fetch, instruction decode, execution, memory access, and write back) and load-store instruction set. What hazards will occur in the following program (please indicate the numbers of the instructions)? And how to solve it? (15%)

```
1.    Load     R1, 3(R2)          ;R1 ← Mem(R2+3)
2.    Add      R3, R2, R7         ;R3 ← R2+R7
3.    Store    0(R4), R3          ;Mem(R4) ← R3
4.    Sub      R2, R1, R5         ;R2 ← R1-R5
5.    Load     R6, 4(R3)          ;R6 ← Mem(R3+4)
6.    Add      R8, R6, R1         ;R8 ← R6+R1
7.    OR       R6, R4, R5         ;R6 ,← R4 or R5
8.    Sub      R3, R7, R2         ;R3 ← R7-R2
```

6. What is daisy-chain arbitration? And what is its application in the computer systems? (10%)