IMPERIAL COLLEGE LONDON

DEPARTMENT OF ELECTRICAL AND ELECTRONIC ENGINEERING
EXAMINATIONS 2015

EIE PART II: MEng, BEng and ACGI

COMPUTER ARCHITECTURE

Corrected Copy

Wednesday, 6 May 2:30 pm

Time allowed: 1:15 hours

There are TWO questions on this paper.

Answer TWO questions.

Answer ALL questions.

Use separate answer books for Sections A and B.

Any special instructions for invigilators and information for candidates are on page 1.

Examiners responsible	First Marker(s) :	W. Luk, D.B. Thomas
	Second Marker(s) :	D.B. Thomas, W. Luk

Section A (Use a separate answer book for this Section)

1 a Explain, with the help of diagrams, what a half adder does, and how half adders are used in a full adder.

b Provide diagrams showing clearly the inputs and outputs when an array of full adders, possibly with additional circuit elements, are used to implement:

i) addition,

ii) subtraction.

c Provide a diagram of a circuit that performs either addition or subtraction, depending on the value of a one-bit input that selects one of these two operations. Your solution, when producing an n-bit output, cannot use more than n full adders.

d Provide a diagram of a 4-bit circuit that shows how the design in Part c can be extended to support two new operations:

i) shift one of the inputs one bit to the left,

ii) shift one of the inputs one bit to the right.

The value of any bits introduced to the output of the above design would be zero.

e Provide a diagram of a 4-bit circuit that shows how the design in Part c can be extended to support a new operation: given two input numbers A and B, the output is zero if A is less than B, otherwise the output is one.

The five parts carry, respectively, 10%, 20%, 20%, 25% and 25% of the marks.

Section B (Use a separate answer book for this Section)

2 The following function performs a very weak hash of an array of data:

```
unsigned hash(int N, const uint32_t *P, int K, int M)
{
  // Pre-condition: 1 <= K < N
  unsigned acc=0, offset=0;
  for(int i=0; i<N; i++){
    unsigned x = P[offset];    // Read next word
    acc = (acc/2) + M*x;       // Fold it into the running hash
    offset = (offset + K) % N;    // Wrap around offset modulo N
  }
  return acc;
}
```

This code is to be compiled for and executed in a standard 5-stage MIPS architecture.

a Describe what event will happen if the function is called with byte address P=4098, and how it will be handled.

b Identify a potential load-use data hazard in the code, and explain how a compiler would avoid it.

c Assume an L1 data-cache with 8 blocks containing 16-bytes each.

i) For a direct-mapped cache, give the sequence of block addresses and block indices accessed during hash for N=8, P=4100 (byte address), K=1.

ii) For an initially empty LRU fully-associative cache, how many bytes of data will be fetched into L1 during hash N=1024, byte address P=1024, K=8? Comment on the total bytes fetched into the cache, versus the total size of the P array, versus the total data used by the program.

d i) Describe how multiplication and division instructions specify the destination register, and explain why they do not use the same approach as other ALU instructions.

ii) For the statements involving division, show how each can be expressed without a division instruction, and whether it would be faster or not in a 5-stage MIPS.

The four parts carry, respectively, 15%, 15%, 35%, and 35% of the marks.