IMPERIAL COLLEGE LONDON

DEPARTMENT OF ELECTRICAL AND ELECTRONIC ENGINEERING
EXAMINATIONS 2016

EIE PART II: MEng, BEng and ACGI

Corrected copy

COMPUTER ARCHITECTURE

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

© Imperial College London

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

1 A program involves scaling up a large set of numbers by multiplying each number by the constant integer K. The processor on which the program runs implements a multiplication instruction (`mult`) which takes 5n cycles, whereas addition (`add`) and subtraction (`sub`) take 2n cycles, and a shift (`srl`, `sll`) only n cycles.

Instruction	CPI	Action
`mult $a, $b, $c`	5n	$R[a] = R[b] \times R[c]$
`add $a, $b, $c`	2n	$R[a] = R[b] + R[c]$
`sub $a, $b, $c`	2n	$R[a] = R[b] - R[c]$
`srl $a, $b, s`	n	$R[a] = R[b]$ shifted right (logical) by s bits
`sll $a, $b, s`	n	$R[a] = R[b]$ shifted left (logical) by s bits

a If the constant K is 9, show how the multiplication can be implemented using `add` and `sll` only. How many clock cycles does the add/shift version take? What is the relative performance of using adds and shifts compared to using the `mult` instruction, assuming any constant which may be needed is already in a register?

b Consider the case when the constant K is 1022. Using the shift/add method as above, how many shift and add instructions are required to implement the multiplication? What is the performance compared with the `mult` instruction?

c Making use of Booth's algorithm, show how multiplication by $K = 1022$ can be implemented using subtraction, addition and shifts. How fast can you make this compared to the `mult` instruction?

The three parts carry, respectively, 30%, 30%, 40% of the marks.

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

2a i) Considering L1 vs L3 cache levels, describe two properties of caches and how you would expect them to vary between L1 and L3.

ii) Give a fragment of code that demonstrates both temporal locality and spatial locality. You can use any language as long as semantics are clear.

iii) Assume a 2-way set associative cache with `w` bytes per word, `b` words per block, and a total cache size of `c` bytes. Given a `byte_address`, give statements for calculating the `set_index`. Use intermediate variables that show your working.

b A CPU designer has suggested that the MIPS ISA is extended with an integer multiply-accumulate instruction, with the following syntax and semantics:

```
mul_add $a, $b, $c //a=a+b*c
```

where `$a`, `$b`, and `$c` are all registers.

i) How many data operand registers does a MIPS R-type instruction support?

ii) How would the register-file of a 5-stage pipelined MIPS need to be modified in order to support this instruction?

iii) Give a sequence of standard MIPS instructions that is equivalent to `mul_add`. You may use the register `$t` as a temporary register.

iv) Assume that both the standard MIPS and extended MIPS execute one instruction per cycle, and caches are warm. Provide a quantitative estimate for the speedup that `mul_add` could provide for the following function. State any assumptions, with a brief explanation of why they are reasonable.

```
uint32_t dot_product(uint32_t n, uint32_t *a, uint32_t *b)
{
  uint32_t acc=0;
  for(uint32_t i=0; i<n; i++){
    acc = acc + a[i] * b[i];
  }
  return acc;
}
```

The two parts carry, respectively, 45%, and 55% of the marks.