FINAL

IMPERIAL COLLEGE OF SCIENCE, TECHNOLOGY AND MEDICINE

EXAMINATIONS 2018

BEng Honours Degree in Computing Part I
MEng Honours Degrees in Computing Part I
for Internal Students of the Imperial College of Science, Technology and Medicine

This paper is also taken for the relevant examinations for the Associateship of the City and Guilds of London Institute

PAPER C113

ARCHITECTURE

Thursday 10th May 2018, 10:00
Duration: 80 minutes

Answer ALL TWO questions

Paper contains 2 questions
Calculators required

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

1 a The 32-bit processor P1 has 32 registers and a 6-bit opcode. Its `load` instruction takes 5 cycles to load a value from memory into a register. Its `add` instruction takes 2 cycles to add the values from two registers, and place the result into a register. Its `shl` instruction takes 1 cycle to shift left a value in a register by a specified amount, and place the result into a register.

i) Show the instruction format for the above three instructions, indicating clearly which bits in the instruction are used for which purpose, and which bits are un-used.

ii) Suggest why different instructions can take different number of cycles to finish.

b A program Q scales up a collection of numbers by multiplying each one by a constant integer K.

When K = 5, provide a sequence of P1 instructions implementing the constant multiplication using only the `add` and the `shl` instructions. How many cycles does it take?

Repeat the above when K = 508.

c The 32-bit processor P2 is the same as P1 except that it has a `mult` instruction, which takes 10 cycles to multiply the values from two registers and place the result into a register. The `mult` instruction is used in implementing the program Q in Part b.

Calculate the percentage difference in run time of Q for P1 and P2, when:

i) K = 5,

ii) K = 508.

The three parts carry, respectively, 40%, 40%, and 20% of the marks.

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

2a *Assembly*

i) Assume these values are stored at the given memory addresses and registers:

Address	Value
0x104	0xAB
0x108	0x12
0x10C	0xFF
0x110	0x3

Register	Value
%rax	0x2
%rcx	0x108
%rdx	0x3

Fill in the table showing the types (*i.e.*, immediate, register, memory) and the values of the indicated operands:

Operand	Type	Mem. Address	Value
%rax			
0x110			
$0x110			
263(%rax, %rdx)			
0x100(,%rax, 4)			
(%rcx,%rax, 2)			

ii) *C*onsider the following x86-64 assembly function:

```
loop:
    pushq   %rbp
    movq    %rsp, %rbp
    movl    %edi, -20(%rbp)
    movl    %esi, -24(%rbp)
    movl    %edx, -28(%rbp)
    movl    -24(%rbp), %eax
    movl    %eax, -8(%rbp)
    movl    -20(%rbp), %edx
    movl    -28(%rbp), %eax
    addl    %edx, %eax
    movl    %eax, -4(%rbp)
    jmp     .L2
.L3:
    addl    $1, -8(%rbp)
    subl    $1, -4(%rbp)
.L2:
    cmpl    $0, -4(%rbp)
    jg      .L3
    movl    -8(%rbp), %eax
    popq    %rbp
    ret
```

Fill in the blanks in the corresponding `loop` C/Java function. You may only use the variable names `a`, `b`, `c`, `x` and `y`, not register names. Use array notation in showing accesses or updates to elements of `a`.

```
int loop(int a, int b, int c){
    int x, y;
    y = _____;
    for (_____; _____; _____) {
        _____;
    }
    return _____;
}
```

b *Floating point encoding.* Consider a six-bit floating point representation based on the IEEE floating point format, with one sign bit, two exponent bits ($k = 2$) and three fraction bits ($n = 3$).

i) What is the exponent bias?

ii) What is the largest value, other than infinity, that can be represented?

iii) What is the smallest positive value, other than 0, that can be represented?

iv) Encode the number 3 in the specified floating point format.

c *Memory Hierarchy.* Assume a cache memory with the following properties: The cache size (C) is 512 bytes and uses an LRU (least recently used) policy for eviction. Furthermore, the cache is initially empty. Now suppose that for the following sequence of addresses sent to the cache, **0, 2, 4, 8, 16, 32**. What is the hit rate if the cache has a block size of 8 bytes? Will the hit rate improve if we increase the block size to 16 bytes? Show your working.

The three parts carry, respectively, 45%, 25%, and 30% of the marks.