IMPERIAL COLLEGE LONDON

TIMED REMOTE ASSESSMENTS 2020-2021

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 assessments for the Associateship of the City and Guilds of London Institute

PAPER COMP40005

INTRODUCTION TO COMPUTER ARCHITECTURE

Wednesday 28 April 2021, 10:00
Duration: 95 minutes
Includes 15 minutes for access and submission

Answer ALL TWO questions
Open book assessment

This time-limited remote assessment has been designed to be open book. You may use resources which have been identified by the examiner to complete the assessment and are included in the instructions for the examination. You must not use any additional resources when completing this assessment.

The use of the work of another student, past or present, constitutes plagiarism. Giving your work to another student to use constitutes an offence. Collusion is a form of plagiarism and will be treated in a similar manner. This is an individual assessment and thus should be completed solely by you. The College will investigate all instances where an examination or assessment offence is reported or suspected, using plagiarism software, vivas and other tools, and apply appropriate penalties to students. In all examinations we will analyse exam performance against previous performance and against data from previous years and use an evidence-based approach to maintain a fair and robust examination. As with all exams, the best strategy is to read the question carefully and answer as fully as possible, taking account of the time and number of marks available.

Paper contains 2 questions

1 Consider the processor P, with two registers A and B. Both of them and the program counter PC, the instruction register IR and the memory M are all 16 bits wide. Memory access can be completed within one clock cycle. The ALU of P produces a signal, Negative, indicating whether the ALU operation gives a negative result. This signal is stored in a 1-bit register to form a condition code bit N, used for making jump decision.

There are only two instructions in P. The SUB (Subtraction) instruction includes two 7-bit memory addresses **d** and **s** respectively at bit 0-6 and bit 7-13 of the instruction. The content in the memory location of address **s** is subtracted from that at address **d**, and the result is stored in the memory location of address **d**. Bit 14 is not used, and bit 15 is '1' which is the opcode. The condition code N is set to '1' if the subtraction result is negative, otherwise it is cleared to '0'.

The JIN (Jump If Negative) instruction contains a 15-bit address for the next instruction if the condition code N = 1. The opcode at bit 15 is '0'.

Any values in the above instructions which are fewer than 16 bits would be extended by a bit-extension unit to become 16 bits when necessary.

a Describe the two instructions using a Register Transfer Language.

b The datapath for P contains the registers PC, IR, A, B and the memory M. A and B are at the inputs of the ALU. The ALU has two outputs: a 16-bit output connected to the data input of M, and a 1-bit output connected to the register for the condition code N, the output of which is connected to the control unit. The output of M is connected to A, B and IR. Bit 15 of IR is connected to the control unit. Appropriate bits in the IR are used as the address of the next instruction if N=1 for the JIN instruction. They are also used as the memory addresses for the SUB instruction. An adder is used to increment the PC. Moreover, there are multiplexors and bit-extension units.

Provide a diagram showing how the components of P are connected together to form the datapath. The diagram should include the multiplexors, the bit-extension units and the control unit, but does not need to show the wires connecting the outputs of the control unit to the multiplexors and other components.

c Provide a state diagram showing the appropriate register transfer assignments in each state of the control unit for the instruction SUB.

The three parts carry, respectively, 20%, 50%, and 30% of the marks.

2 a i) *Assembly.* Consider the following snipped code of a function:

```
int A[M][N];
int B[P][Q];

int mult_element (int i, int j){

    return A[i][j] * B[j][i];
}
```

In compiling with optimisation flags this program, GCC generates the following x86-64 assembly code:

```
mult_element:
    movslq  %esi, %rsi
    movslq  %edi, %rdi
    leaq    (%rsi, %rsi, 4), %rax
    leaq    (%rsi, %rdi, 4), %rdx
    addq    %rax, %rdi
    movl    A(, %rdx, 4), %eax
    imull   B(, %rdi, 4), %eax
    ret


    B:   .zero 80
    A:   .zero 48
```

Use reverse engineering skills to determine the values of M, N, P and Q based on the above assembly code.

ii) *Stack discipline.* Consider the following x86-64 assembly function:

```
half_increment:
        subq    $24, %rsp
        movl    %edi, %eax
        movl    %edi, 12(%rsp)
        testl   %edi, %edi
        jne     .L4
.L1:
        addq    $24, %rsp
        ret
.L4:
        movl    $2, %esi
        leaq    12(%rsp), %rdi
        movl    $0, %eax
        call    div
        addl    12(%rsp), %eax
        jmp     .L1
```

Fill in the blanks in the following `half_increment` function. You may only use the variable names `a` and `x`, not register names.

```
int half_increment(int x) {
        int ____ = ____;
        if(______)
        {
            ______;
        }
        else{
              __ = div(__,__);
              _______;
            }
}
```

b *Memory Hierarchy.* A 128 MB main memory has a 64KB direct-mapped cache with 16 bytes per line.

i) How many lines are there in the cache?

ii) Show how the main memory is partitioned.

The two parts carry, respectively, 70% and 30% of the marks.