UNIVERSITY OF LONDON
IMPERIAL COLLEGE OF SCIENCE, TECHNOLOGY AND MEDICINE

EXAMINATIONS 2001

BEng Honours Degree in Computing Part I
MEng Honours Degrees in Computing Part I
BSc Honours Degree in Mathematics and Computer Science Part I
MSci Honours Degree in Mathematics and Computer Science 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
This paper is also taken for the relevant examinations for the Associateship of the Royal College of Science

PAPER C110=MC110

ARCHITECTURE I

Friday 11 May 2001, 16:00
Duration: 90 minutes
(Reading time 5 minutes)

Answer THREE questions

Paper contains 4 questions
Calculators required

1a Copy out and complete the entries in the following table. All binary numbers are 8-bit. Write *out-of-range* in a table cell if the value cannot be represented.

Decimal	Sign & Magnitude	1's Complement	2's Complement	Excess 75
–73				
	1010 1110			
		1001 1001		
				0100 1010

b The n-bit pattern $b_{n-1}\, b_{n-1} \ldots b_2\, b_1\, b_0$ has the value $\sum_{k=0}^{n-1} b_k \times 2^k$ when treated as an unsigned value. What is the value of the n-bit pattern for each of the following representations?

i) Sign and Magnitude *Hint*: $-1^1 = -1,\ -1^0 = +1$

ii) Two's Complement

iii) One's Complement

iv) Excess E

c Using your answer from part b(iii) show mathematically that the sum of a one's complement integer and its bitwise inverse is zero. *Hint*: The inverse of a bit b is $1–b$.

d Give a method for copying a 16-bit sign and magnitude integer ($h_{15}\, h_{14} \ldots h_1\, h_0$) to a 8-bit sign & magnitude integer ($b_7\, b_6 \ldots b_1\, b_0$). Your solution should indicate which bits are copied, and also which bits are tested to detect overflow before the copy.

The four parts carry, respectively, 20%, 40%, 30% and 10% of the marks.

2a Define the RET and IRET instructions clearly detailing the differences between them.

b The Pentium CPU supports a 'divide by zero' exception. What should an interrupt handler for this exception actually do? Where should the handler return once it is finished?

c Suppose that interrupt-driven I/O devices can be assigned priorities e.g. between 0 (Low) and 7 (High). How might a CPU use such priorities for interrupt processing? Order the priorities for the following 4 interrupts: (i) a millisecond resolution clock, (ii) a stack fault exception, (iii) a printer, and (iv) a video camera.

d In some processors an interrupt is only handled after the completion of the current instruction. Consider the possibility of a processor recognizing and processing interrupts while an instruction is executing. Discuss the difficulties that would arise and also give a situation where such interrupt processing would be desirable.

The four parts carry, respectively, 25%, 25%, 25%, and 25% of the marks.

3 Translate the following high-level language function into a **commented** Pentium assembly language version:

```
function Palindromic
  (var str : array 0..* of char, len : int) : boolean
var k : int

  for k := 0 to len div 2 - 1
   if str(k) not= str(len - k - 1)
        result false
   end if
  end for

  result true

end Palindromic
```

You should assume:

- a flat memory model for segments
- 32-bit integers

State any additional assumptions that you make.

4a For the Pentium architecture devise a parameter passing convention that primarily uses registers to pass parameters to a procedure or function instead of the stack. For your convention explain:

i) in what order parameters are allocated to registers.

ii) how 8-bit, 16-bit and 32-bit values are passed.

iii) how values larger than 32-bits are passed, e.g. large arrays

iv) how **var** parameters are handled.

v) how the convention handles procedure calls where there are more parameters than registers.

vi) how function results are returned (you can assume that only 8-bit, 16-bit or 32-bit values are ever returned).

vii) how local variables are allocated.

viii) when registers are saved and when registers are restored

b) Using the convention that you devised in part 4a give the sequence of Pentium instructions needed for the following assignment statement:

```
Result := Alpha ('A', 24, Vec, Counter)
```

Assume that **int**'s are 32-bit, and that `Vec` and `Counter` are suitably defined global variables. `Alpha` is a function with the following header:

```
function Alpha (a:char, b:int, c:array 0..3 of int,
                var d:int) : int
```

State any additional assumptions that you make.

c) Most programming languages support functions that can return values larger than the total register size of architecture, e.g. can return large arrays. Show how you would extend your calling convention to return large function results.

d) State the advantages and disadvantages of your register-based convention.

The four parts carry, respectively, 40%, 30%, 15% and 15% of the marks.