Corrected Copy

IMPERIAL COLLEGE OF SCIENCE, TECHNOLOGY AND MEDICINE

EXAMINATIONS 2010

BEng Honours Degree in Computing Part II
MEng Honours Degrees in Computing Part II
BEng Honours Degree in Information Systems Engineering Part II
MEng Honours Degree in Information Systems Engineering Part II
MSc in Computing Science
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 C210=E2.13

COMPUTER ARCHITECTURE

Friday 14 May 2010, 14:30
Duration: 120 minutes

Answer THREE questions

Paper contains 4 questions
Calculators required

1 a Explain what are *CPI* and *MIPS rating*, and provide a formula relating the two.

b If machine A has a clock rate α times that of machine B, a MIPS rating β times that of machine B, and an instruction count γ times that of machine B for programs of interest, and α, β and γ are all larger than one, which machine would be faster and by how much?

c What does *MFLOPS rating* stand for? If a machine P has a higher MIPS rating than the machine Q, would P also have a higher MFLOPS rating than Q? Justify your answer.

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

2 A program involves scaling up a large set of numbers by multiplying each number by the constant integer C. The processor on which the program runs implements a multiplication instruction (`mult`) which takes 10 cycles, whereas addition (`add`) and subtraction (`sub`) take 2 cycles, and a shift (`srl`, `sll`) only 1 cycle.

Instruction	CPI	Action
`mult $a, $b, $c`	10	$R[a] = R[b] \times R[c]$
`add $a, $b, $c`	2	$R[a] = R[b] + R[c]$
`sub $a, $b, $c`	2	$R[a] = R[b] - R[c]$
`srl $a, $b, s`	1	$R[a] = R[b]$ shifted right (logical) by s bits
`sll $a, $b, s`	1	$R[a] = R[b]$ shifted left (logical) by s bits

a If the constant C is 5, 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?

b Consider the case when the constant C is 508. 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 $C = 508$ 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.

3a Provide the diagram of a 5-bit combinational circuit for implementing an unsigned divide-by-two operation. Label the input and output values on the diagram when the input to this circuit is 01100 (which is 12 in base ten).

b Consider a combinational circuit for implementing a restricted version of a divide-by-three operation, which accepts a 3-bit unsigned number and produces a 1-bit quotient and a 2-bit remainder.

i) Provide the truth table for this circuit, and explain why some input combinations will lead to "don't care" results.

ii) Provide the optimised Boolean equations for the 3 outputs in terms of the 3 inputs.

c The combinational circuit in Part b can be used as a repeating unit for an N-bit divide-by-three circuit, by connecting the remainder outputs of one circuit to two of the 3 inputs of another. Draw a circuit diagram of a 3-bit divide-by-three circuit constructed this way, and label the values on the internal and output wires when the input is 111.

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

4a What is a page table?

b A system can address 2^v bytes of virtual memory and 2^p bytes of physical memory. The page size is 2^s bytes, and each page table entry is 2^e bytes. Calculate:

i) the number of page table entries,

ii) the size of the page table.

c One way of reducing page table size is to keep only the active page table entries in physical memory, by having multiple levels of page tables.

For the system described in Part b, explain how address translation can work with multiple levels of page tables, and calculate:

i) the number of pages indexed per page,

ii) the number of levels of page tables needed, and

iii) the number of physical memory accesses needed for address translation in the event of a TLB miss.

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

Master - July 2010

Department of Computing Examinations — 2009–2010 Session		*Confidential*
MODEL ANSWER and MARKING SCHEME		
First Examiner: WL	Paper Code: C210 = E2.13	
Second Examiner: KK Leung	Question 1	Page 1 *out of* 1
Question labels in left margin		Mark allocations in right margin

1a CPI : cycles per instruction
MIPS rating : million instructions per second

$$\text{MIPS rating} \times \text{CPI} \times 10^6 = \text{clock rate} \quad \text{—— ①}$$

6

b $$\text{exec. time}_A = \text{instr. count}_A \times \text{CPI}_A \times \frac{1}{\text{clock rate}_A}$$

$$= \frac{\text{instr. count}_A}{\text{MIPS rating}_A \times 10^6} \quad \text{from ①}$$

$$\frac{\text{exec. time}_A}{\text{exec. time}_B} = \frac{\text{instr count}_A}{\text{instr. count}_B} \times \frac{\text{MIPS rating}_B}{\text{MIPS rating}_A},$$

$\text{MIPS rating}_A = \beta\ \text{MIPS rating}_B$

$$= \frac{\gamma}{\beta}$$

so A is faster than B by $\left(\frac{\beta}{\gamma}\right)$ times, if $\beta > \gamma$

8

c MFLOPS rating : million floating point operations per second
Higher MIPS rating need not imply higher MFLOPS rating, if floating point instructions form a low proportion of all the instructions of a given program, and B is more efficient at floating point while A is more efficient at other instructions, then B could have a higher MFLOPS while A has a higher MIPS

6

Department of Computing Examinations — 2009–2010 Session	*Confidential*
MODEL ANSWER and MARKING SCHEME	
First Examiner wl	Paper Code C 210 = E 2·13
Second Examiner kk leung	Question 2 Page 1 *out of* 1
Question labels in left margin	Mark allocations in right margin

a $n \times 5 = 4 \times n + n$, so

```
sll $2, $1 2
add $2, $1 $2
```

This takes 3 cycles, 3·3 times faster than the mult instruction — 6

b If c is 508 = 111111100 need 7 shifts and 6 adds
Below assumes orginal number in $1, result in $2,
temporary storage in $3

```
sll $2 $1 2
sll $3 $1 3
add $2 $2 $3
sll $3 $1 4
add $2 $2 $3
sll $3 $1 5
add $2 $2 $3
sll $3 $1 6
add $2 $2 $3
sll $3 $1 7
add $2 $2 $3
sll $3 $1 8
add $2 $2 $3
```

This takes $7 + 6 \times 2 = 19$ cycles,
1·9 times slower than the mult instruction — 6

c Using Booth's algorithm, replace the 6 adds by 1 subtract, since — 8

$n \times 508 = n \times (512 - 4) = n \times 512 - n \times 4$

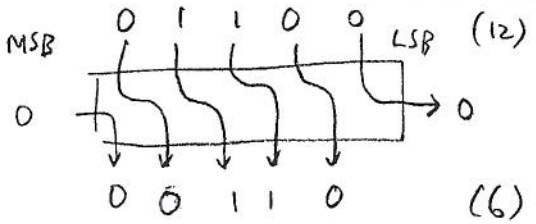
Multiply by 512 is shift left by 9 bits, so

```
sll $2 $1 2
sll $3 $1 9
sub $2 $3 $2
```

This takes 4 cycles, 2·5 times faster than the mult instruction

Department of Computing Examinations — 2009–2010 Session	*Confidential*
MODEL ANSWER and MARKING SCHEME	
First Examiner wl	Paper Code C210 = E2.13
Second Examiner kk leung	Question 3 Page 1 *out of* 1
Question labels in left margin	Mark allocations in right margin

3a unsigned divide by 2



4

b unsigned divide by 3 : 1 bit quotient (q), 2 bit remainder

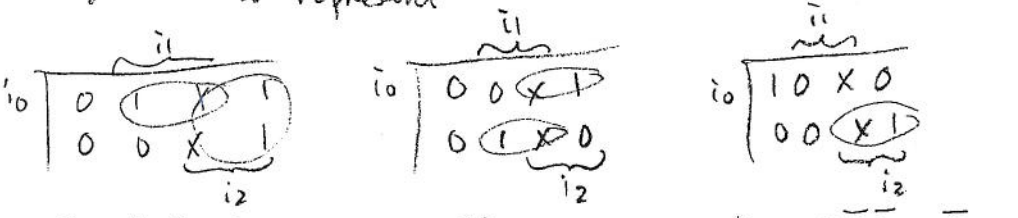
i) input i2 i1 i0

				q	r1	r0	
0	0	0	(0)	0	0	0	(0)
0	0	1	(1)	0	0	1	(1)
0	1	0	(2)	0	1	0	(2)
0	1	1	(3)	1	0	0	(0)
1	0	0	(4)	1	0	1	(1)
1	0	1	(5)	1	1	0	(2)

input 110, 111 are don't care, since they need 2 bit quotient to represent

4

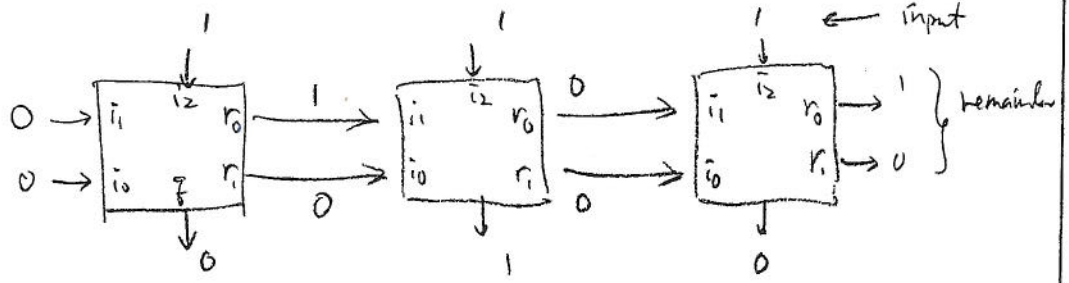
ii)



$q = i_0 \cdot \bar{i}_1 + i_2$ $\quad r_1 = \bar{i}_0 i_1 + i_0 \bar{i}_2$ $\quad r_0 = \bar{i}_0 \bar{i}_1 \bar{i}_2 + \bar{i}_0 i_2$

4

c



8

Department of Computing Examinations — 2009–2010 Session			*Confidential*
MODEL ANSWER and MARKING SCHEME			
First Examiner	wl	Paper Code	C210 = E213
Second Examiner	kk leung	Question 4	Page 1 *out of* 1
Question labels in left margin			Mark allocations in right margin

4 a In virtual memory systems, a virtual memory block forms a page, and the page table contains the virtual to physical address translation. 4

b. virtual memory 2^v bytes, physical memory 2^p bytes
page size 2^s bytes, page table entry 2^e bytes

i) number of page table entries $= \frac{2^v}{2^s} = 2^{v-s}$ (3)

ii) size of page table $= (2^{v-s}) \cdot (2^e) = 2^{v+e-s}$ bytes (3) 6

c how multiple levels of page table performs address translation:
look at first table using highest order bits;
if valid, use the next highest order bits to index the next table (4)

i) number of pages indexed per page: $\frac{2^s}{2^e} = 2^{s-e}$ (2)

ii) 2^{v-s} page table entries, so need: $\lceil \frac{v-s}{s-e} \rceil$ levels (2)

iii) same answer as ii) (2) 10