IMPERIAL COLLEGE OF SCIENCE, TECHNOLOGY AND MEDICINE

EXAMINATIONS 2009

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

Tuesday 12 May 2009, 14:30
Duration: 120 minutes

Answer THREE questions

Paper contains 4 questions
Calculators required

1 a What does CPI stand for? Explain how a compiler can affect CPI.

b Provide one benefit and one drawback for architectures supporting (i) register-register instructions, (ii) register-memory instructions.

c Measurements of the machine RR show that the fraction of ALU instructions is α and the fraction of load instructions is β; the remaining fraction is taken up by other instructions. The cycle count for (i) an ALU instruction is x, (ii) for a load instruction is y, and (iii) for all other instructions is z. What is the CPI for RR?

d A new machine RM is obtained by adding register-memory instructions for ALU operations to the instruction set of RR. These register-memory instructions have one source operand in memory, and each takes m cycles. Given that a fraction k of ALU operations directly use a loaded operand that is not used again, what is the CPI for RM? What is the ratio of execution time for RR and RM?

The four parts carry, respectively, 15%, 20%, 15%, 50% of the marks.

2 For this question, your diagrams do not need to show the internal details of half adders, full adders and multiplexers.

a Provide a diagram for ALU0, a 4-bit ALU that is capable of computing bitwise boolean *and* and boolean *or*.

b Explain, with the help of a diagram, how ALU0 can be extended to become ALU1, which supports a single-bit output producing a one when all outputs from ALU0 is zero.

c Explain, with the help of a diagram, how ALU1 can be extended to become ALU2, which supports a single-bit output producing a one when all the corresponding bits of the two inputs to ALU2 are the same. In other words, bit zero of both inputs are the same, bit one of both inputs are the same, and so on.

d Explain, with the help of a diagram, how ALU2 can be extended to become ALU3, which produces the absolute value of the 4-bit outputs from ALU2.

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

3 For this question, your diagrams do not need to show the internal details of half adders, full adders and multiplexers.

a Provide a diagram showing how a one-bit data variable for the token-passing hardware compilation approach can be implemented using a D flipflop and one or more combinational logic components.

b Provide a diagram for the control circuit for implementing the parallel statement for the token-passing hardware compilation approach, using an SR flipflop. Provide another diagram showing how the SR flipflop can be implemented using a D flipflop and one or more combinational logic components.

c Show how the control circuit in Part b can be simplified, provided that all the statements that run in parallel take either zero or one clock cycle.

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

4 a Explain why an instruction cache usually has lower miss rate than a data cache.

b Suggest one advantage and one disadvantage for (i) separate instruction cache and data cache, each of n bytes, (ii) a unified cache of $2n$ bytes for both instruction and data.

c Calculate the miss rate for a machine S with separate instruction cache and data cache, each of n bytes. There are i misses per k instructions for the instruction cache, and d misses per k instructions for the data cache. A fraction α of instructions involve data transfer, while a fraction β of instructions contain instruction references and the rest contain data references. A hit takes h cycles and the miss penalty is m cycles.

d Calculate the miss rate for a machine U with a unified cache of $2n$ bytes, for both instruction and data. There are u misses per k instructions for this unified cache. A fraction α of instructions involve data transfer.

e Calculate the Average Memory Access Time for S and U.

The five parts carry, respectively, 15%, 20%, 20%, 20%, 25% of the marks.

Department of Computing Examinations — 2008–2009 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

a CPI stands for cycles per instruction. Compiler affects CPI since it can select instructions that take fewer cycles to run in translating a program into machine code. 3

b. Register-register instructions : simple, fast, but high instruction count
Register-memory " : slower due to memory access, but dense code 4

c.
$$CPI_{RR} = CPI_{ALU} + CPI_{load} + CPI_{Store}$$
$$= \alpha x + \beta y + (1-\alpha-\beta) z$$
3

d Let CPI for RM for ALU, load and Store instructions be $CPI_{ALU'}$, $CPI_{load'}$, $CPI_{Store'}$

$$CPI_{ALU'} = (km + (1-k)x)\frac{\alpha}{1-k\alpha} = \frac{\alpha}{1-k\alpha}(km + x - kx)$$

$$CPI_{load'} = y\left(\frac{\beta - k\alpha}{1-k\alpha}\right)$$

$$CPI_{store'} = z\left(\frac{1-\alpha-\beta}{1-k\alpha}\right)$$

$$CPI_{RM} = CPI_{ALU'} + CPI_{load'} + CPI_{Store'}$$
$$= \frac{1}{1-k\alpha}(\alpha km + \alpha x - \alpha k x + \beta y - k\alpha y + z - \alpha z - \beta z)$$

ratio of run time
$$= \frac{\alpha x + \beta y + (1-\alpha-\beta)z}{(1-k\alpha)\left(\frac{\alpha km + \alpha x - \alpha k x + \beta y - k\alpha y + z - \alpha z - \beta z}{1-k\alpha}\right)}$$

$$= \frac{\alpha x + \beta y + (1-\alpha-\beta)z}{\alpha x + \beta y + (1-\alpha-\beta)z + \alpha k(m - x - y)}$$
10

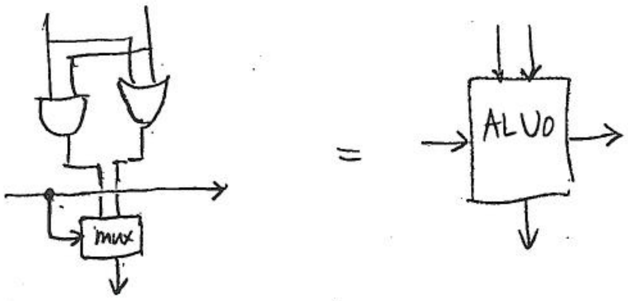
Department of Computing Examinations — 2008–2009 Session	***Confidential***

MODEL ANSWER and MARKING SCHEME

First Examiner	wl	Paper Code	C210 = E2.13
Second Examiner	kk leung	Question 2	Page 1 *out of* 1

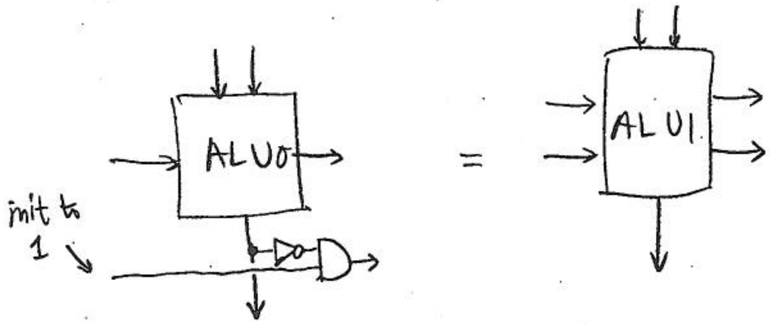
Question labels in left margin — Mark allocations in right margin

a



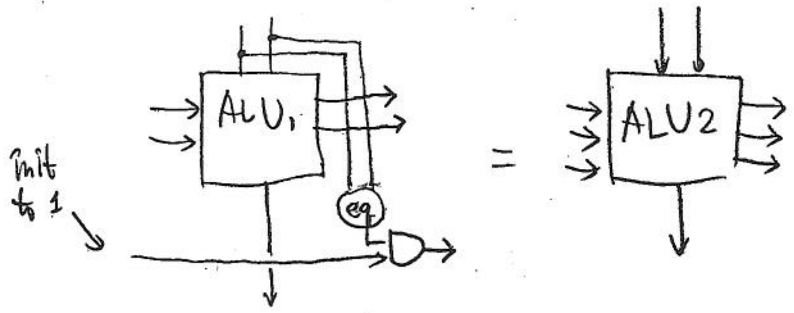
4

b



4

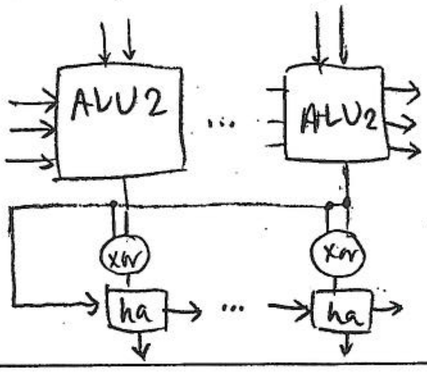
c



a	b	eq
0	0	1
0	1	0
1	0	0
1	1	1

6

d



a	b	xor
0	0	0
0	1	1
1	0	1
1	1	0

6

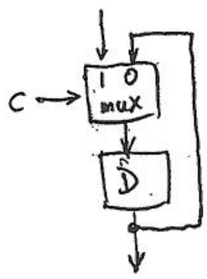
Department of Computing Examinations — 2008–2009 Session *Confidential*

MODEL ANSWER and MARKING SCHEME

First Examiner	wl	Paper Code	C210 = E2.13
Second Examiner	kk leung	Question 3	Page 1 *out of* 1

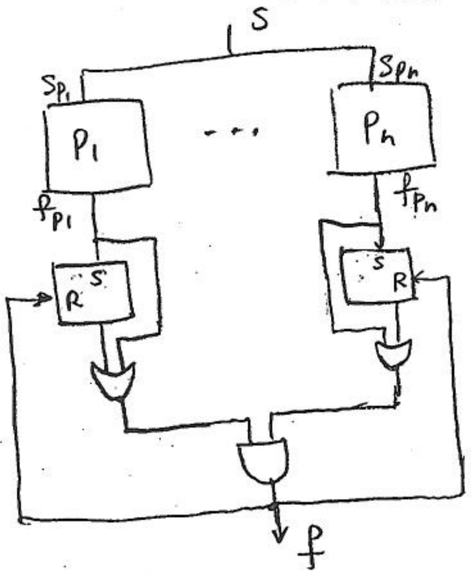
Question labels in left margin — Mark allocations in right margin

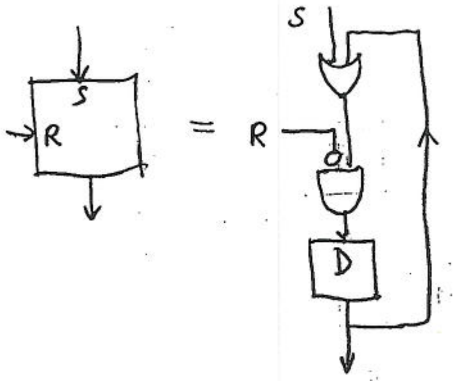
a



4

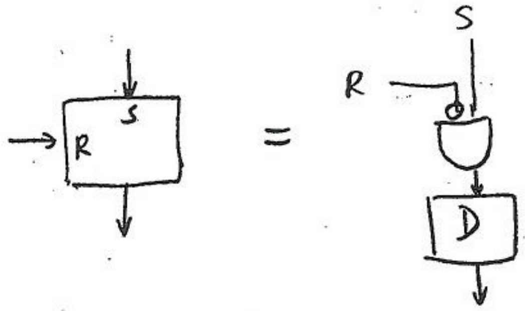
b





8

C



8

(revised)

Department of Computing Examinations — 2008–2009 Session			*Confidential*
MODEL ANSWER and MARKING SCHEME			
First Examiner	wl	Paper Code	C210 = E 2.13
Second Examiner	kk leung	Question 4	Page 1 *out of* 1
Question labels in left margin			Mark allocations in right margin

a successive instructions often benefit from spatial locality 3

b separate I & D caches: higher bandwidth, avoiding misses due to conflicts between I and D blocks; but each smaller
Unified I & D cache: more flexible partitioning between I & D reducing capacity misses, but conflict misses between I & D 4

c instr. miss for S: i/k
data miss for S: $\alpha d/k$ 4
overall miss for S: $(i + \alpha d)/k$

d overall miss for U: $(u + \alpha u)/k = u(\alpha + 1)/k$ 4

e AMAT: hit time + (miss rate) × (miss penalty)

$AMAT_S = h + (i + \alpha d)/k \times m$

(in cycles, per instr) $= (kh + m(i + \alpha d))/k$

$AMAT_U = h + u(\alpha + 1)k \times m$

$= (kh + mu(\alpha + 1))/k$ 5