Corrected Copy

IMPERIAL COLLEGE OF SCIENCE, TECHNOLOGY AND MEDICINE

EXAMINATIONS 2011

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
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

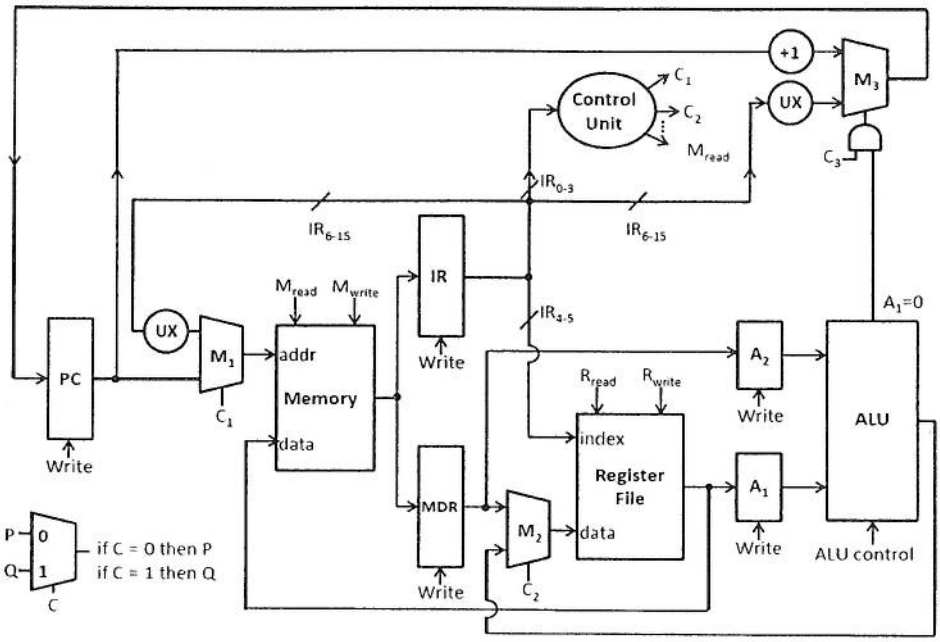
Friday 20 May 2011, 14:30
Duration: 75 minutes
(Reading time 5 minutes)

*Answer ALL TWO questions*

Paper contains 2 questions
Calculators required

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

1 The processor P1 takes16-bit instructions. Bits 0-3 are the opcode, bits 4-5 indicate a register in a register file, and the remaining 10 bits are used as address or data. The architecture of P1 is shown below. The registers PC, IR, MDR, A1 and A2 retain their value unless their "write" input is 1. When "ALU control" is 1, the ALU performs $A_1+A_2$; otherwise it performs $A_1-A_2$. The register file and memory have separate signals for reading and writing. The "index" input selects a register.




a The "load" instruction for P1 takes 3 cycles. In cycle 1, PC is incremented and the instruction is fetched to IR. In cycle 2, bits 6-15 from IR is extended by UX to be a 16-bit address for the memory, and the content from this address is stored in MDR. In cycle 3, the value in MDR is stored in a register in the register file. What are the values of the control signals $C_1$, $C_2$, $C_3$, $M_{read}$, $M_{write}$, $R_{read}$, $R_{write}$ etc in each cycle for the "load" instruction? No need to specify a control signal if its value is zero.

b The "add" instruction for P1 adds the value from a register to a value from memory with the 10 bits in the instruction as address. How many cycles does the "add" instruction take? What are the values of the control signals in each cycle?

c The ALU status signal "$A_1$=0" becomes 1 when the value from $A_1$ is 0. The "if-zero" instruction for P1 jumps to a memory address given by the 10 bits in the instruction if a given register is zero. How many cycles does the "if-zero" instruction take? What are the values of the control signals in each cycle?

d The "load-immediate" instruction takes the 10-bit field in the instruction as an unsigned number and loads it into a given register. Modify P1 to support this instruction, and describe the control signals in each cycle for this instruction.

*The four parts carry, respectively, 15%, 25%, 20%, 40% of the marks.*

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

2a i) What is the main goal of pipelining processor data-paths?

ii) What is the key problem when adding pipelining to a processor, and what are the three main classifications of this problem?

b The data-path for a small specialised 32-bit processor is built from logic primitives with the following latencies:

| Component | Delay |
|---|---|
| Instruction Memory | 10 ns |
| Register File | 8 ns |
| Adder | 8 ns |
| Multiplier | 40 ns |
| Dynamic Shifter | 10 ns |
| Two-Input Multiplexor | 2 ns |

The processor supports exactly four instructions: addition, multiplication, left shift, and move (register-register copy). The program counter is 12 bits wide, and the instruction memory contains 4096 words.

i) What is the minimum clock period of the processor if it is not pipelined?

ii) What is the maximum clock rate if the processor is pipelined into fetch, decode, and execute stages?

c It has been suggested that a direct-mapped instruction cache with 64 one-word blocks be added to the non-pipelined processor. The cache reduces the instruction fetch time to 5ns, and for cache misses the processor stalls for one cycle and is guaranteed to get a cache hit for the same instruction in the next cycle.

i) For a cache hit-rate $p$, calculate the average execution time per instruction of the non-pipelined processor with instruction cache.

ii) Given the restricted instruction set of the processor, discuss whether the instruction cache makes sense in this case.

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

| Department of Computing Examinations — 2010–2011 Session | | | *Confidential* |
|---|---|---|---|
| **MODEL ANSWER and MARKING SCHEME** | | | |
| First Examiner | wl | Paper Code | 210 |
| Second Examiner | d. thomasl | Question / | Page / *out of* / |
| Question labels in left margin | | | Mark allocations in right margin |

**1 a** For the load instruction,

cycle 1: $C_1 = 1$, $C_3 = 0$, $M_{read} = IR_{write} = PC_{write} = 1$, the rest zero

2: $C_1 = 0$, $M_{read} = MDR_{write} = 1$

3: $C_2 = 0$, $R_{write} = 1$ — 3

**b** 4 cycles for the add instruction

cycle 1: as in load instr.

cycle 2 $M_{read} = MDR_{write} = 1$

cycle 3: $R_{read} = A_1 write = A_2 write = 1$

cycle 4: $C_3 = R_{write} = ALU\ control = 1$ — 5

**c** 3 cycles for the if zero instruction

cycle 1: as in load instr

cycle 2: $R_{read} = A1\ write = 1$

cycle 3: $C_3 = PC\ write = 1$ — 4

**d** To support the "load immediate" instruction,

(a) connect output of UX next to $M_1$ to the input of $M_2$

(b) change $C_2$, control for $M_2$, to be 2 bits, such that the output of $M_2$ becomes the new connection when $C_2 = 2$ — 4

cycle 1: as in load instruction

cycle 2: $R_{write} = 1$, $C_2 = 2$ — 4

2a i) What is the main goal of pipelining processor data-paths?

*Increase clock rate / instructions per second, by breaking execution of each instruction across multiple cycles.*

**Marks:** 2

ii) What is the key problem when adding pipelining to a processor, and what are the three main classifications of this problem?

*Hazards mean one instructions may not be completed per cycle.*

*Data hazards; Control hazards; Structural hazard.*

**Marks:** 4

b The data-path for a small specialised 32-bit processor is built from logic primitives with the following latencies:

| Component | Delay |
|---|---|
| Instruction Memory | 10 ns |
| Register File | 8 ns |
| Adder | 8 ns |
| Multiplier | 40 ns |
| Dynamic Shifter | 10 ns |
| Two-Input Multiplexor | 2 ns |

The processor supports exactly four instructions: addition, multiplication, left shift, and move (register-register copy). The program counter is 12 bits wide, and the instruction memory contains 4096 words.

i) What is the minimum cycle time of the processor if it is not pipelined?

*D(CPU) = D(instr$) + D(reg) + D(ALU)*

*D(ALU) = multiply or shift or add or copy.*

$$= 10 + 8 + \max(40, \max(\max(8, 0) + 2, 10) + 2) + 2 \quad (1)$$
$$= 10 + 8 + 42 \quad (2)$$
$$= 60 \quad (3)$$

*The PC update is irrelevant here, as it is only every incremented, and there is no direct connection between data-path and control.*

**Marks:** 3

ii) What is the maximum clock rate if the processor is pipelined into fetch, decode, and execute stages?

$$= 10^9 / \max[10, 8, \max(40, \max(10, \max(8, 0) + 2) + 2) + 2] \quad (4)$$
$$= 10^9/42 \quad (5)$$
$$= 23.809523MHz \quad (6)$$

**Marks:** 3

c It has been suggested that a direct-mapped instruction cache with 64 one-word blocks be added to the non-pipelined processor. The cache reduces the instruction fetch time to 5ns, and for cache misses the processor stalls for one cycle and is guaranteed to get a cache hit for the same instruction in the next cycle.

i) For a cache hit-rate $p$, calculate the average cycles per instruction of the non-pipelined processor with instruction cache.

$$CPI = (60 - 5) * (p + (1 - p) * 2) \quad (7)$$
$$= 55 * (p + 2 - 2 * p) \quad (8)$$
$$= 55 * (2 - p) \quad (9)$$
$$= 110 - 55p \quad (10)$$

*If they realise that $p = 0$ at this stage and reduce to 110 then that is fine, as long as they indicate when they substitute.*

**Marks:** 4

ii) Given the restricted instruction set of the processor, discuss whether the instruction cache makes sense in this case.

*There are no jump instructions, so the program-counter can only increase. The program repeats every 4096 instructions, which is larger than the cache, and so the cache never hits.*

*So $p = 0$ and the instruction cache halves the performance of the processor.*

**Marks:** 4

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