UNIVERSITY COMPUTER SCIENCE • EXERCISE SHEET

6.1 Instruction set architecture

Name: _____ Class: _____ Date: _____ **Total: 24 marks**

KEY WORDS instruction set architecture • RISC

Objective

Encode and decode instructions in both formats, and respect the immediate's range.

1 Key knowledge and worked examples

■ The formats

R-type: opcode(4) | rd(4) | rs(4) | rt(4) | shift(4) | funct(8) – 28 bits used ✓. I-type: opcode(4) | rt(4) | rs(4) | pad(4) | immediate(16) – 32 bits ✓.

■ Worked example: the range

A 16-bit two's complement immediate spans −32 768 to 32 767, so `addi R2, R0, 40000` cannot encode; word offsets are multiples of 4, reaching 64 KiB from the base ✓.

2 Practice

2.1 Give the field values for `add R3, R1, R2` (opcode 1, funct 1): six numbers in format order. [3]

2.2 State the largest and smallest encodable `addi` immediate. [3]

2.3 Name the two instructions that touch data memory in a load-store ISA. [3]

3 Exam-style questions

3.1 The teaching ISA encodes `addi R5, R2, imm` with opcode 2.

Write the field values for `addi R5, R2, 60` in I-type order (including the unused pad), naming each field. Decode the I-type word whose fields read opcode 3, rt 6, rs 3, immediate 12 as a `lw` (mnemonic and operands). State which single field tells these two instructions apart, and why the R-type format also needs a funct field. [8]

3.2 A program needs R7 to hold 40 000 before a loop, starting from R0 = 0.

State why one `addi` cannot do it. Write a two-instruction sequence that can (any correct pair, e.g. an addi of 20 000 then an add of the register to itself), giving each field's values. State the largest multiple of 4 reachable by one `lw` offset and what the ISA guarantees about R0 throughout. [7]

Solutions

2.1

- opcode 1, rd 3, rs 1, rt 2, shift 0, funct 1 ✓.

2.2

- Largest 32 767; smallest −32 768 ✓.

2.3

- `lw` and `sw` ✓.

3.1

- `addi R5, R2, 60`: opcode 2, rt 5, rs 2, pad 0, immediate $60 = 111100_2$ ✓.
- Opcode 3, rt 6, rs 3, imm 12 decodes to `lw R6, 12(R3)` ✓.
- The opcode alone tells them apart (2 versus 3) ✓.
- R-type reuses one opcode for many operations – the funct field completes the selection ✓.

3.2

- $40\,000 > 32\,767$: outside the 16-bit signed immediate ✓.
- E.g. `addi R7, R0, 20000` (opcode 2, rt 7, rs 0, imm 20000) then `add R7, R7, R7` (opcode 1, rd 7, rs 7, rt 7, shift 0, funct 1) ✓.
- Largest positive `lw` offset: 32 764 – the greatest multiple of 4 inside the signed 16-bit range; the field's full unsigned span covers $2^{16} = 64$ KiB of addresses around the base ✓.
- R0 is hardwired: every read yields zero and writes to it vanish ✓.