4.3 Building a Datapath

Sign Extension vs. Zero Extension:

The immediate field of I-type instructions is 16 bits (26 bits in J-type)

The immediate value is often used in 32-bit operations

Sign Extension (***SignExtImm*** on green card):

MSb (bit 15) of immediate value is replicated 16 times for bits 31 through 16

SignExtImm = { 16{immediate[15]}, immediate}

The 2's complement sign of extended value is the same as original

Zero Extension (***ZeroExtImm*** on green card):

Bits 31 through 16 are filled with 16 zeros

ZeroExtImm = { 16{1'b0}, immediate}

The MSb (bit 31) is 0, so the value is positive (whether used in signed or unsigned arithmetic subsequently)

MIPS core instruction set (left column page 1 of green card) regularities:

16 bit immediate values are <u>always</u> treated as signed numbers in address calculations

Data addresses (load and store):

sw M[R[rs]+SignExtImm] = R[rt]

Data address immediate values <u>never</u> are shifted before use

Instruction addresses (branch):

beq if(R[rs] == R[rt]) PC = PC + 4 + BranchAddr

BranchAddr = { 14{immediate[15]}, immediate, 2'b0 }

Instruction address immediate values <u>always</u> are shifted 2 before use

26-bit immediate values in the Jump (j) and Jump and Link (jal) instructions are not treated as signed numbers

j PC = JumpAddr

JumpAddr = { PC+4[31:28], address, 2'b0 }

PC+4[31:28] the 4 most-significant bit of the value PC+4

address the 26-bit immediate field of the J-type inst.

Only two core instructions zero extend the immediate value

OR immediate	ori	R[rt] =R[rs] | ZeroExtImm
AND immediate	andi	R[rt] = R[rs] & ZeroExtImm

OR and AND are the only logic instructions with immediate versions

A "u" (unsigned) at the end of an instruction mnemonic never refers to a difference in the way an immediate value is treated

addiu R[rt] = R[rs] + SignExtImm

addi/addiu is an exception behavior difference

lbu R[rt] = { 24'b0, M[R[rs] + SignExtImm](7:0) }

Memory address is formed using sign extended immediate

The value read from memory is the object treated as unsigned

Sign/Zero Extension in Hardware (Figure 4.9)*:*

Chapter 4 subset of instructions using 16-bit immediate values

load word (lw) and store word (lw)

Need to sign extend immediate because address calculations always sign extend

Sign extended value is passed to ALU for data address calculation

branch equal (beq)

Need to sign extend immediate because address calculations always sign extend

Need to shift left 2 places because branch address calculations always assume 2 least-significant bits are 00 (branch addresses always point to word-aligned 32-bit instructions)

Sign extended and shifted value is passed to special-purpose *branch target address* adder

AND immediate (andi) and OR immediate (ori) are not part of the Chapter 4 subset, so 16-bit zero extension hardware is not included