6.823
Handout#8

6.823 Computer System Architecture

L-MIPS ISA and Single-Cycle Datapath

http://csg.csail.mit.edu/6.823/

Ben Bitdiddle is unhappy with the performance of the MIPS processor discussed in the 6.823 lecture. He wants to extend the MIPS ISA with a new class of instructions, LOAD-ALU (L), described in Table 1. He calls his new ISA the L-MIPS ISA.

31 26	25 21	20 16	15 11	10 0
opcode	rs	rt	rd	imm

Instruction	**Format and Description**
Load and ALU	ALUM rd, offset(base), rt rd ← Mem[(rs) + SignExt(imm)] op (rt) Performs an ALU operation with one register and one memory operand. The effective address of the first operand is computed by adding the contents of register *rs* (base) to the sign-extended 11-bit *immediate* field (offset). The second operand is the contents of register *rt*. eg: ADDM R1, 4(R3), R2 XORM R1, -7(R1), R1

Table 1. L-MIPS LOAD-ALU(L) instruction type.

To implement the L-MIPS ISA, Ben modifies the datapath of the single-cycle MIPS processor as shown in Figure H-1. He adds an address generation unit, and moves the ALU unit after the data memory. Loads, stores and L-type instructions use the address generation unit before querying the memory, while other instructions use the ALU unit after the memory. Note that the address generation unit comprises only an adder.

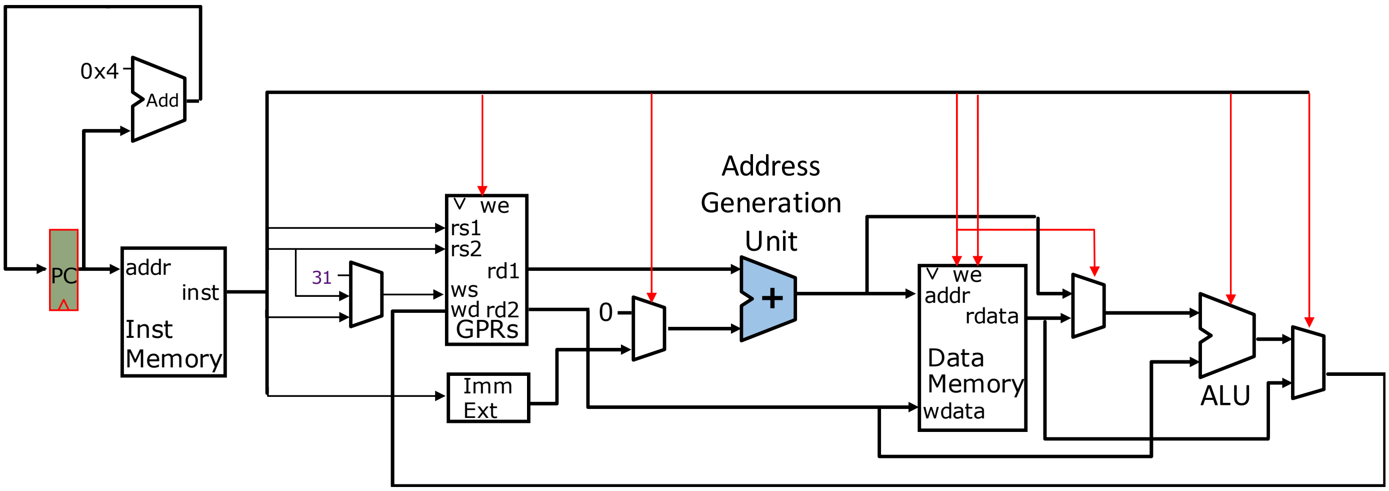

Figure H-1: L-MIPS single-cycle datapath.