CS/COE0447

Computer Organization & Assembly Language

Chapter 5 Part 3

Short reference version

Multi-Cycle Execution: R-type

- Instruction fetch
  - IR <= Memory[PC]; sub $t0,$t1,$t2
  - PC <= PC + 4;
- Decode instruction/register read
  - A <= Reg[IR[25:21]]; rs
  - B <= Reg[IR[20:16]]; rt
  - ALUOut <= PC + (sign-extend(IR[15:0])<<2);
- Execution
  - ALUOut <= A op B; op = add, sub, and, or,…
- Completion
  - Reg[IR[15:11]] <= ALUOut; $t0 <= ALU result

Multi-cycle Execution: lw

- Instruction fetch
  - IR <= Memory[PC]; lw $t0,-12($t1)
  - PC <= PC + 4;
- Instruction Decode/register read
  - A <= Reg[IR[25:21]]; rs
  - B <= Reg[IR[20:16]];
  - ALUOut <= PC + (sign-extend(IR[15:0])<<2);
- Execution
  - ALUOut <= A + sign-extend(IR[15:0]); $t1 + -12 (sign extended)
- Memory Access
  - MDR <= Memory[ALUOut]; M[$t1 + -12]
- Write-back
  - Load: Reg[IR[20:16]] <= MDR; $t0 <= M[$t1 + -12]