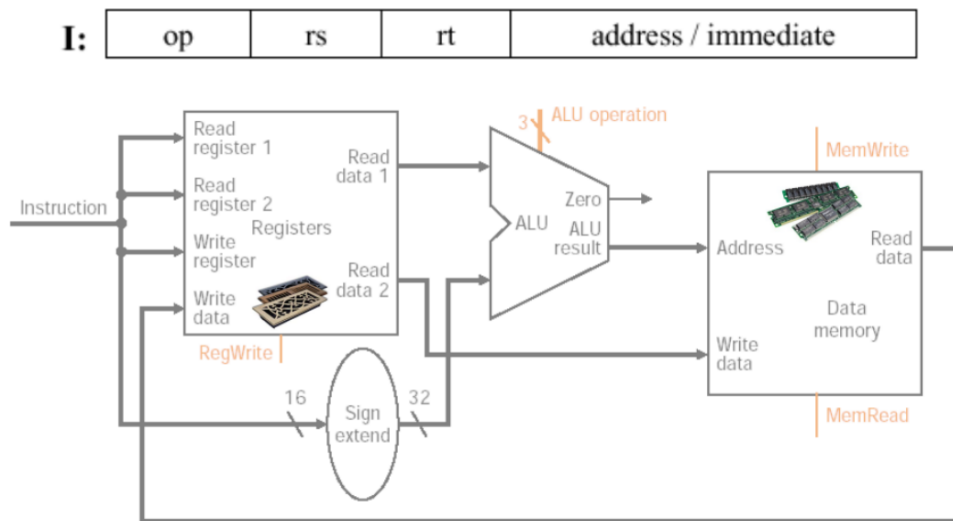


อธิบายการทำงานของคำสั่ง Load-Store



Consider: $r1 = M[r2 - 3]$

คำสั่ง Load-Store เป็นคำสั่งที่ใช้ในการถ่ายโอนข้อมูลระหว่างรีจิสเตอร์ (Registers) และหน่วยความจำ (Data Memory) โดยคำสั่ง **Load** จะอ่านข้อมูลจากหน่วยความจำมาเก็บไว้ในรีจิสเตอร์ ส่วนคำสั่ง **Store** จะเขียนข้อมูลจากรีจิสเตอร์ไปยังหน่วยความจำ

ลำดับการทำงานสำหรับ $r1 = M[r2 - 3]$:

1. ดึงคำสั่ง `lw r1, -3(r2)`
2. อ่านค่าจากรีจิสเตอร์ r2 (Read data 1)
3. Sign extend ค่า -3
4. ALU คำนวณ Address โดย $r2 + (-3)$
5. ใช้ Address ที่ได้ไปอ่านข้อมูลจาก Data Memory (MemRead = 1)
6. นำข้อมูลที่อ่านได้จาก Data Memory มาเขียนกลับลงในรีจิสเตอร์ r1 (RegWrite = 1, Write register = r1)

การทำงานของคำสั่ง Load-Store

การทำงานของคำสั่ง Load-Store สามารถแบ่งออกเป็นขั้นตอนหลักๆ ได้ดังนี้:

1. การดึงคำสั่ง (Instruction Fetch)

ในภาพไม่ได้แสดงส่วนของการดึงคำสั่ง แต่โดยทั่วไปแล้ว CPU จะดึงคำสั่ง มาจากหน่วยความจำคำสั่ง (Instruction Memory)

2. การถอดรหัสและอ่านรีจิสเตอร์ (Instruction Decode and Register Fetch)

- Instruction : คำสั่ง I-type ที่ดึงมาจะถูกส่งเข้ามาในส่วนของ Control Unit เพื่อถอดรหัส (Decode) และสร้างสัญญาณควบคุมต่างๆ
- Read register 1 (rs): ค่าจากฟิลด์ rs ของคำสั่งจะถูกใช้เพื่อเลือกรีจิสเตอร์ต้นทางตัวที่ 1 (rs) เพื่ออ่าน Read data 1 ซึ่งจะเป็นค่าที่ใช้ในการคำนวณ Address ฐาน (Base Address)
- Read register 2 (rt): ค่าจากฟิลด์ rt ของคำสั่งจะถูกใช้เพื่อเลือกรีจิสเตอร์ต้นทางตัวที่ 2 (rt) เพื่ออ่าน Read data 2 ซึ่งค่านี้จะถูกใช้เป็น Write data ในกรณีของคำสั่ง Store

3. การคำนวณ Address ด้วย ALU (Execute)

- Sign extend: ฟิลด์ address / immediate (ขนาด 16 บิต) จะถูกขยายเครื่องหมาย (Sign extended) ให้เป็น 32 บิต เพื่อให้สามารถนำไปใช้ในการคำนวณทางคณิตศาสตร์ได้
- ALU operation: ค่า Read data 1 (จากรีจิสเตอร์ rs) และค่า Sign extended immediate จะถูกส่งไปยัง ALU (Arithmetic Logic Unit)
- Address: ALU จะทำการบวกค่าทั้งสองเข้าด้วยกัน (โดยทั่วไปคือ Base Address + Offset) เพื่อคำนวณหา Address ทางกายภาพของหน่วยความจำที่ต้องการเข้าถึง

4. การเข้าถึงหน่วยความจำข้อมูล (Memory Access)

- Address: Address ที่คำนวณได้จาก ALU จะถูกส่งไปยัง Data memory เพื่อระบุตำแหน่งของข้อมูลที่ต้องการ

- MemWrite / MemRead:
 - MemRead (สำหรับคำสั่ง Load): สัญญาณ MemRead จะถูกเปิดใช้งาน ทำให้หน่วยความจำอ่านข้อมูลจากตำแหน่ง Address ที่ระบุ และส่งออกมาทางช่อง Read data
 - MemWrite (สำหรับคำสั่ง Store): สัญญาณ MemWrite จะถูกเปิดใช้งาน และค่า Write data (ซึ่งมาจาก Read data 2 ของรีจิสเตอร์ rt) จะถูกเขียนไปยังตำแหน่ง Address ที่ระบุในหน่วยความจำ

5. การเขียนผลลัพธ์กลับสู่รีจิสเตอร์ (Write Back)

- RegWrite: สัญญาณ RegWrite จะถูกเปิดใช้งาน
- Write data:
 - สำหรับคำสั่ง Load: ข้อมูลที่อ่านได้จากหน่วยความจำ (Read data) จะถูกส่งกลับมาเพื่อเขียนไปยังรีจิสเตอร์ปลายทางที่ระบุโดยฟิลด์ rt (ในภาพ: rt เป็น Destination Register)
 - สำหรับคำสั่ง Store: คำสั่ง Store จะไม่มีการเขียนข้อมูลกลับไปยังรีจิสเตอร์ (RegWrite จะเป็น 0) เพราะข้อมูลถูกเขียนลงหน่วยความจำแล้ว

จากตัวอย่างที่ให้มา $r1 = M[r2 - 3]$ ซึ่งเป็นคำสั่ง Load (คล้ายกับ lw r1, -3(r2) ใน MIPS)

- rs: จะชี้ไปที่รีจิสเตอร์ r2 (ซึ่งค่าใน r2 จะเป็น Base Address)
- rt: จะชี้ไปที่รีจิสเตอร์ r1 (ซึ่งจะเป็น Destination Register ที่จะเก็บข้อมูลที่โหลดมา)
- address / immediate: จะมีค่าเป็น -3 (ซึ่งจะถูก Sign extended และนำไปบวกกับค่าใน r2)