1. **li (Load Immediate)**
   * Cú pháp: **li rd, immediate**
   * Chức năng: Tải một giá trị hằng số (immediate) vào một thanh ghi (register) **rd**.
   * Ví dụ: **li t6, 0** tải giá trị **0** vào thanh ghi **t6**.
2. **beq (Branch if Equal)**
   * Cú pháp: **beq rs1, rs2, label**
   * Chức năng: So sánh hai thanh ghi **rs1** và **rs2**. Nếu giá trị bằng nhau, nhảy đến nhãn **label** trong mã.
   * Ví dụ: **beq a3, t0, equal\_0** sẽ nhảy đến nhãn **equal\_0** nếu giá trị trong **a3** và **t0** bằng nhau.
3. **jal (Jump and Link)**
   * Cú pháp: **jal rd, label**
   * Chức năng: Nhảy đến nhãn **label** và lưu địa chỉ lệnh kế tiếp vào thanh ghi **rd** (thường là **ra** để dùng cho việc trở về sau khi hàm được gọi hoàn tất).
   * Ví dụ: **jal ra, display\_number** nhảy đến hàm **display\_number** và lưu địa chỉ trở về trong **ra**.
4. **la (Load Address)**
   * Cú pháp: **la rd, symbol**
   * Chức năng: Tải địa chỉ của biến hoặc nhãn **symbol** vào thanh ghi **rd**.
   * Ví dụ: **la a4, rows** tải địa chỉ của mảng **rows** vào thanh ghi **a4**.
5. **lw (Load Word)**
   * Cú pháp: **lw rd, offset(rs1)**
   * Chức năng: Tải một từ (32-bit) từ bộ nhớ vào thanh ghi **rd**. Địa chỉ bộ nhớ được xác định bởi **offset** cộng với giá trị trong **rs1**.
   * Ví dụ: **lw a1, 0(s0)** tải từ tại địa chỉ **s0** vào **a1**.
6. **sw (Store Word)**
   * Cú pháp: **sw rs2, offset(rs1)**
   * Chức năng: Lưu từ trong thanh ghi **rs2** vào địa chỉ bộ nhớ xác định bởi **offset** cộng với giá trị trong **rs1**.
   * Ví dụ: **sw t6, 0(a1)** lưu từ trong **t6** vào địa chỉ được xác định bởi **a1**.
7. **slli**

**rd**: Thanh ghi đích nơi kết quả sẽ được lưu.

**rs1**: Thanh ghi nguồn chứa giá trị cần dịch.

**shamt**: Số lượng bit dịch sang trái, là một giá trị hằng số.

1. MV rd, rs --- MOV
2. LB rd, offset(rs1)

**rd**: Là thanh ghi đích (destination register) để lưu trữ giá trị dữ liệu được tải từ bộ nhớ.

* **offset**: Là giá trị offset (độ lệch) chỉ ra vị trí cần đọc trong bộ nhớ.
* **rs1**: Là thanh ghi chứa địa chỉ cơ sở (base address) của vùng nhớ cần đọc.

1. **remu**

li t0, 15 # t0 = 15, số bị chia

li t1, 4 # t1 = 4, số chia

remu t2, t0, t1 # t2 = t0 % t1, phần dư là 3

1. lệnh divu ;

* **rd**: Là thanh ghi đích (destination register) để lưu trữ kết quả phép chia (phần thương).
* **rs1**: Là thanh ghi chứa bộ chia (dividend).
* **rs2**: Là thanh ghi chứa bộ chia (divisor).

1. SB

SB x3, 10(x1)

Lệnh này sẽ ghi giá trị trong thanh ghi **x3** vào vị trí **rs1 + 10** trong bộ nhớ, với độ dài là 1 byte.

1. SLT rd, rs1, rs2

**rd**: Là thanh ghi đích (destination register) để lưu trữ kết quả của phép so sánh.

**rs1**: Là thanh ghi chứa giá trị cần kiểm tra.

**rs2**: Là thanh ghi hoặc giá trị tham chiếu để so sánh với giá trị trong **rs1**.