

**RISC – V instrukcje**:

**U** – działania opierają się na liczbach całkowitych zapisanych w OPCODE (20 bitowe stałe)

**I** – działania opierają się na liczbach całkowitych zapisanych w rejestrach i w OPCODE (12 bitowe stałe)

**R** – działania opierają się na liczbach całkowitych zapisanych w rejestrach

**J** – instrukcje sterowania transferem (skoki)

**B** – w architekturze nie przewidziano rejestru flag przechowującego status po wykonanych operacjach, uznano że instrukcje rozgałęzienia z wbudowanym porównaniem rozwiążą problem

**Load-store** – ograniczone tryby adresowania (typowe dla risc)

Dostęp do pamięci możliwy za pomocą dwóch instrukcji:

**\*LOAD, typ I**

**\*STORE, typ S**

PROJEKT SYKO INSTRUKCJE RISC-V

**SW –** instrukcja typu STORE, typ S - zapisz w pamięci pod efektywny adres wartość 32 bitową z rs2

**SLL –** instrukcja typu R – logiczne przesunięcie w lewo, młodsze bity zapełnione zerami

Rd = rs1 << rs2[5…0]

**LUI –** intruckcja typu U – załadowanie do rs górnych 20 bitów z imm (dolne ustawione na 0)

Rd = imm[32…12] | zero [11…0]

**JAL –** instrukcja typu J, skok – bezwarunkowy skok relatywny - adres skoku wyznacza: PC+imm2

- imm2 to imm[19...0] rozszerzone ze znakiem i pomnożone przez 2 (tzw. word alignment)

- PC użyte do operacji wyznaczania adresu wskazuje na aktualna instrukcje, ta wartość

powiększona o 4 (następna instrukcja) trafia też do rd (typowo używane jest x1 lub x5)

- zasięg skoku +/- 1MiB

**AUIPC –** instrukcja typu U – tworzy 32bitowy offset sumując PC z górnymi 20bitami z imm (przyjmuje dolne 12 bitów jest równe zero

Rd = PC + (imm[32…12]|zero[11…0])

**ADD –** instrukcja typu R – jak w typie I , Rd = rs2+rs2 | rd = rs1-rs2

**SLTI –** instrukcja typu I – wstawia wartość 1 do rd jeżeli rs1 jest większy od imm w przeciwnym przypadku wstawia 0 do rd

**SB** – instrukcja typu STORE, typ S - zapisz w pamięci pod efektywny adres wartość 8 bitową – używaj młodszych bitów z rs2

**JALR –** instrukcja typu J, skok – bezwarunkowy skok nie bezpośredni

- imm jest tylko 12 bitowy ze znakiem

- adres skoku wyznacza: rs1+imm2

**BNE** – instrukcja typu B

- skok gdy rs1 i rs2 są różne

- gdy warunek nie jest spełniony - przejście do następnej instrukcji a gdy ma być wykonany skok - nowe miejsce w kodzie wyznacza: PC+imm2\*2

- imm2 to imm[12...0] rozszerzone ze znakiem i pomnożone przez 2 (tzw. word alignment)

**ADDI –** instrukcja typu I

- dodawanie imm\*) do rs1 i zapisanie w rd

- addi sp,sp,-16

- pseudo instrukcja: MV rd,rs1 może być emulowana przez: addi rd,rs1,0

- rozdział „RISC-V Assembly Programmer’s Handbook” specyfikacji zawiera „Table 25.2: RISC-V pseudoinstructions” opisująca wiele możliwości emulowania nie istniejących instrukcji

