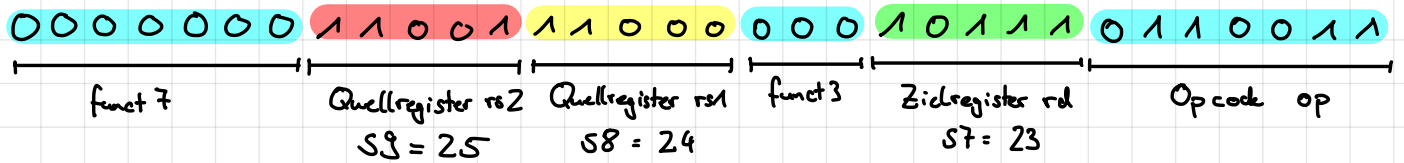


ERA-Übungsblatt 07

1. • Übersetzung Assembly → Maschinensprache in RISC-V-32 sehr einfach (weil fixe 32-Bit-Größe)
→ einfach in Tabellen nachschauen und daraus Binärzahl bauen

add s7, s8, s9

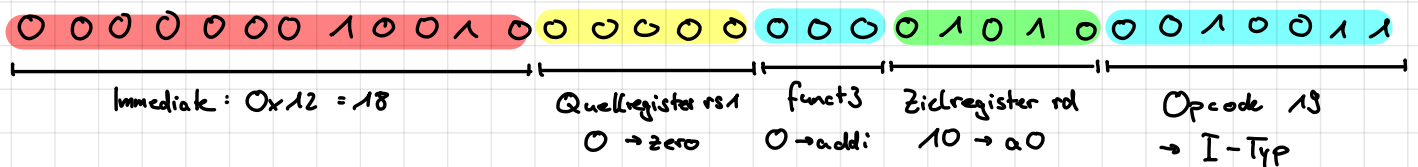


Das Layout der einzelnen Komponenten kann in der Tabelle nachgeschlagen werden, da wir wissen, dass "add" ein R-Typ-Befehl ist

Ins Hexadezimalsystem angewandelt: 0b00...011 = 0x013C0BB3
weiter siehe ML

• Jetzt umgekehrt! Da unabhängig vom Instruktionstypen die hintersten 7 Bit immer den Opcode enkodieren, können wir daraus den Befehlstyp ablesen und anschließend alle anderen Parameter auslesen.

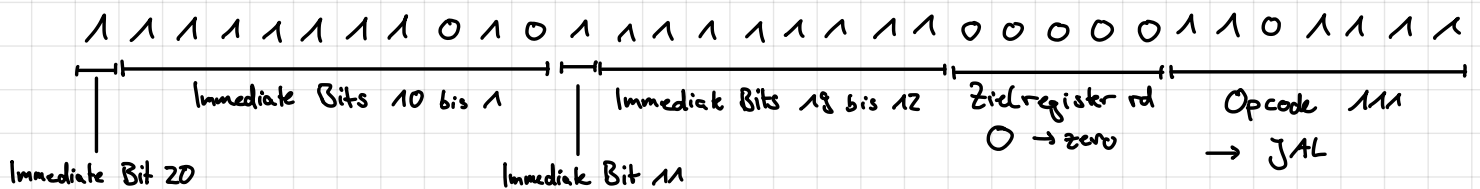
0x01200513 in Binär:



die dekodierte Instruktion lautet also: addi a0, zero, 18

ein weiteres (interessantes) Beispiel:

0xff5ff06f in Binär:



Wir können also den Immediate zusammen basteln:

imm = 1111111111111111010 = -12 (Zweierkomplement!)

Loop:
ins1
ins2
ins3
jal zero, loop

die dekodierte Instruktion lautet also jal, zero, -12

Hinweis: Der Immediate "-12" wurde wahrscheinlich vom Assembler auf einem Label generiert: -12 ≙ "Springe 3 Instruktionen zurück"