G:

S → “000000 [opcode]” [0] Format Function

Function → Arithmetic | UncJump | Logic | ConJump | Shift

Arithmetic → Sum | Sub | Mult | Div

Sum → “100000 [add]”[1] | “100001 [addu]”[1]

Sub→ “100010 [sub]”[4] | “100011 [subu]”[4]

Div → “011010 [div]”[5][20] | “011011 [divu]”[6][20]

Mult → “011000 [mult]”[7][20] | “011001 [multu]”[7][20]

UncJump → “001000 [jr]”

Logic → “100100 [and]”[8] | “100111 [nor]”[9] | “100101 [or]”[10]

ConJump → “101010 [slt]”[11] | “101001 [sltu]”[12]

Shift → “000000 [sll]”[13] | “000010 [srl]”[14] | “000011 [sra]”[15]

Format → “[00000-11111] [(rs | 00000)]”[19] B

B → “000000000000000 [shamt]” | “[00000-11111] [rt][22]” C

C → “0000000000 [shamt]” | “[00000-11111] [rd][21]” D

D → “00000 [shamt]” | “[00000-11111] [sa][23][25]”

[0] {S.opcode := “0110011”}
[1] {Sum.opfunct7 := “0000000”, Sum.opfunct3 := “000”}
[2] {C.rs1 := rs.value}
[3] {C.rs2 := rt.value}
[4] {Sub.opfunct7 := “0100000”, Sub.opfunct3 := “000”}
[5] {Div.opfunct7 := “0000001”, Div.opfunct3 := “100”}
[6] {Div.opfunct7 := “0000001”, Div.opfunct3 := “101”}
[7] {Mult.opfunct7 := “0000001”, Mult.opfunct3 := “000”}
[8] {Logic.opfunct7 := “0000000”, Logic.opfunct3 := “111”}
[9] {Logic.opfunct7 := “0000000”, Logic.opfunct3 := “000”}
[10] {Logic.opfunct7 := “0000000”, Logic.opfunct3 := “110”}
[11] {ConJump.opfunct7 := “0000000”, ConJump.opfunct3 := “010”}
[12] {ConJump.opfunct7 := “0000000”, ConJump.opfunct3 := “011’”}

[13] {Shift.opfunct7 := “0000000”, Shift.opfunct3 := “001”}
[14] {Shift.opfunct7 := “0000000”, Shift.opfunct3 := “101”}
[15] {Shift.opfunct7 := “0100000”, Shift.opfunct3 := “110”}
[16] {Shift.function7 := “0100000”, Shift.opfunct3 := “110”}
[15] {Shift.opfunct7 := “0100000”, Shift.opfunct3 := “110”}
[18] {Shift.opfunct7 := “0100000”, Shift.opfunct3 := “110”}
[19] {Format_rs := rs.lex}
[20] {insthi := “0000000000000000 rd.lex 00000010010”, instlo :=
“0000000000000000 rd.lex 00000010010”}
[21] {C_rd := rd.lex}
[22] {B_rt := rt.lex}
[23] {D_sa := sa.lex}
[24] {rv_instruction := funct7 + rs2 + rs + funct3 + rd + opcode}
[25] {rv_instructionAdd := “00100000000 + (register) + bin_n(D_sa)”}