The process\_instruction() function should be able to simulate the instruction-level execution of the following subset of entire RV32I instructions:

|  |  |
| --- | --- |
| Instruction type | instruction |
| R | add, slt |
| I | addi, slli |
| S | sw |
| SB | bne |
| U | auipc |
| UJ | jal |

A table with numbers and letters

Description automatically generated

|  |  |  |  |
| --- | --- | --- | --- |
| register | Name | Use | binary |
| x0 | Zero | Contant zero | 00000 |
| x1 | ra | Return Address | 00001 |
| x2 | sp | Stack pointer | 00010 |
| x3 | gp | Global pointer | 00011 |
| x4 | Tp | Thread pointer | 00100 |
| x5 | T0 | temporary | 00101 |
| x6 | T1 | temporary | 00110 |
| x7 | T2 | temporary | 00111 |
| x8 | S0/sp | Saved register/Frame pointer | 01000 |
| x9 | S1 | Saved register | 01001 |
| x10 | A0 | Function Arguments/return values | 01010 |
| x11 | A1 | Function Arguments/return values | 01011 |
| x12 | A2 | Function Arguments | 01100 |
| x13 | A3 | Function Arguments | 01101 |
| x14 | A4 | Function Arguments | 01110 |
| x15 | A5 | Function Arguments | 01111 |
| x16 | A6 | Function Arguments | 10000 |
| x17 | A7 | Function Arguments | 10001 |
| x18 | S2 | Saved Registers | 10010 |
| x19 | S3 | Saved Registers | 10011 |
| x20 | S4 | Saved Registers | 10100 |
| x21 | S5 | Saved Registers | 10101 |
| x22 | S6 | Saved Registers | 10110 |
| x23 | S7 | Saved Registers | 10111 |
| x24 | S8 | Saved Registers | 11000 |
| x25 | S9 | Saved Registers | 11001 |
| x26 | S10 | Saved Registers | 11010 |
| x27 | S11 | Saved Registers | 11011 |
| x28 | T3 | temporaries | 11100 |
| x29 | T4 | temporaries | 11101 |
| x30 | T5 | temporaries | 11110 |
| x31 | T6 | temporaries | 11111 |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Instr** | **Funct7** | **Rs2** | **Rs1** | **Funct3** | **Rd** | **Opcode** | **Assembly** | **Hex** |
| add | 0000000 | 00101 | 00110 | 000 | 00111 | 0110011 | Add x7, x6, x5 | 0x005303b3 |
| slt | 0000000 | 11100 | 11101 | 010 | 11110 | 0110011 | Slt, x30, x29, x28 | 0x01ceaf33 |

Table 1:R type instructions.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Instr** | **Imm[11:0]** | **Rs1** | **Funct3** | **Rd** | **Opcode** | **Assembly** | **Hex** |
| addi | 000000000111 | 00000 | 000 | 11111 | 0010011 | Addi x31, x0, 7 | 0x00700f93 |
| slli | 000000001010 | 00101 | 001 | 11110 | 0010011 | Slli x30, x5, 10 | 0x00a29f13 |

Table 2: I type instructions.

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Instr** | Imm[11:5] | **Rs2** | **Rs1** | **Funct3** | Imm[4:0] | **Opcode** | **Assembly** | **Hex** |
| sw | 00000 | 01001 | 10010 | 010 | 11111 | 0100011 | Sw x9, 31(x18) | 0x00992fa3 |

Table 3:S type instructions.

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Instr** | Imm[12|10:5] | **Rs2** | **Rs1** | **Funct3** | Imm[4:1|11] | **Opcode** | **Assembly** | **Hex** |
| bne | 0000101 | 00101 | 11111 | 001 | 10100 | 1100011 | bne x30, x0, 180 | 0x0a0f1a63 |

Table 4:SB type instructions.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Instruction** | Imm[31:12] | **Rd** | **Opcode** | **Assembly** | **Hex** |
| auipc | 01111111111111111111 | 11011 | 0010111 | auipc x27, 524287 | 0x7ffffd97 |

Table 5:U type instructions.

Note: auipc: (Add Upper Immediate to Program Counter): this sets rd to the sum of the current PC and a 32-bit value with the low 12 bits as 0 and the high 20 bits coming from the U-type immediate

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Instr** | Imm[20|10:1|11|19:12] | **Rd** | **Opcode** | **Assembly** | **Hex** |
| jal | 00000001000000000000 | 11011 | 1101111 | jal x27, 16 | 0x01000def |

Table 6:UJ type instructions.