SimpleRISC Assembler

Assembly Code

```
; SimpleRISC Assembly Example
; Basic program demonstrating different instruction types

start:
    mov r1, 10      ; Load 10 into r1
    mov r2, 20      ; Load 20 into r2
    add r3, r1, r2  ; r3 = r1 + r2 = 30
    sub r4, r3, r1  ; r4 = r3 - r1 = 20

    ; Memory operations
    st r3, 0[r0]    ; Store r3 value at address 0
    ld r5, 0[r0]    ; Load value from address 0 into r5

    ; Branching
    cmp r1, r2      ; Compare r1 and r2
    bgt greater     ; Branch if r1 > r2
    b end           ; Unconditional branch to end

greater:
    mov r6, 1       ; This won't execute (r1 is not > r2)

end:
    mov r7, 255     ; End marker
    hlt             ; Halt execution
```

Load Example Load File

Machine Code

Binary Hex

```
01001100010000000000000000001010
01001100100000000000000000010100
00000000110001001000000000000000
00001001000011000100000000000000
01111100110000000000000000000000
01110101010000000000000000000000
00101000010000001000000000000000
10001111111111111111111111110100
10010111111111111111111111110011
01001101100000000000000000000001
01001101110000000000000011111111
11111000000000000000000000000000
```

Assemble Code Download HEX

Messages

Assembly completed successfully!

Instructions & Reference

▼ Opcodes Reference

Instruction	Opcode (Binary)
add	00000
sub	00001
mul	00010
div	00011
mod	00100
cmp	00101
and	00110
or	00111
not	01000
mov	01001
lsl	01010
lsr	01011
asr	01100
nop	01101
ld	01110
st	01111
beq	10000
bgt	10001
b	10010
call	10011
ret	10100

hlt	11111

▼ Registers Reference

Register	**Binary Code**
r1	0001
r2	0010
r3	0011
r4	0100
r5	0101
r6	0110
r7	0111
r8	1000
r9	1001
r10	1010
r11	1011
r12	1100
r13	1101
r14	1110
r0	0000

▼ Instruction Types

Type	**Description**	**Examples**
Type 0	No operands	nop, ret, hlt
Type 1	Branch instructions	call, b, beq, bgt
Type 2	Two operands	cmp, not, mov
Type 3	Three operands	add, sub, mul, div
Type 4	Memory operations	ld, st