RV32 ABI and Registers cheat sheet

Register	ABI Name	Role in Calling Convention
x0	zero	Hardwired zero
x1	ra	Return address
x2	sp	Stack pointer
x3	gp	Global pointer
x4	tp	Thread pointer
x5	t0	Temporary register 0
x6	t1	Temporary register 1
x7	t2	Temporary register 2
x8	s0 / fp	Saved register / Frame pointer
x9	s1	Saved register
x10	a0	Argument / Return value 0
x11	a1	Argument / Return value 1
x12	a2	Argument 2
x13	a3	Argument 3
x14	a4	Argument 4
x15	a5	Argument 5
x16	a6	Argument 6
x17	a7	Argument 7
x18	s2	Saved register
x19	s3	Saved register
x20	s4	Saved register
x21	s5	Saved register
x22	s6	Saved register
x23	s7	Saved register
x24	s8	Saved register

x25	s9	Saved register
x26	s10	Saved register
x27	s11	Saved register
x28	t3	Temporary register 3
x29	t4	Temporary register 4
x30	t5	Temporary register 5
x31	t6	Temporary register 6

Temporary Registers (`t0`–`t6`): These are caller-saved registers. If the caller needs to preserve their values across function calls, it must save and restore them.

Saved Registers (`s0`–`s11`): These are callee-saved registers. The callee is responsible for saving and restoring their values if it uses them.

Argument Registers (`a0`–`a7`): These registers are used to pass arguments to functions and to return values.

Special Registers:

`x0` (zero): Always returns 0; writes to it are ignored.

`x1` (ra): Stores the return address for function calls.

`x2` (sp): Points to the top of the stack; used for stack operations.

`x3` (gp): Global pointer; points to the global data area.

`x4` (tp): Thread pointer; points to thread-local