🧠 Instruction Set

> **Version**: `0.5.0`
> **Total Instructions Supported**: `73`
> **Operand Size**: Up to **2 operands**, each of **64-bits**

🧩 Operand Types

Instructions in SASM accept a variety of operand types:

??? example "Immediate Values"

~~~
```
0, 1, 2, 3
```
~~~

??? example "Constants"

~~~
```
M, N, O, P
```
~~~

??? example "Registers"

~~~
```
L0, L1, L2, L3
```
~~~

🗂️ Registers

SASM registers are identified by **unique IDs**. Each register can be accessed either as a **reference** or as a **value** using *compile-time functions*.

```
ref(L1)  ; reference to register L1
val(L1)  ; value stored in register L1
```

> These are **compile-time functions** that tell the assembler how to interpret operands.

🔗 Related References

- 📄 Compile-time Functions
- 📄 Register Reference

✅ Notes:

- The `??? example` block is an enhanced feature of Material for MkDocs, using collapsible examples.
- Replace the `../reference/...` links with the actual paths to your documentation pages.
- Use emojis or icons lightly to improve scannability, especially for longer technical docs.
- Ensure all the referenced `.md` files are included in `mkdocs.yml`.

Let me know if you'd like this rendered live or exported!

Instruction Set

As of ver. `0.5.0`, SASM has support for 73 instructions!

Each instruction can have upto 2 operands, each 64-bits in size.

!!! Example "The operands can be either immediate values."

```
0,1,2,3
```

!!! Example "Or they can be constants"

```
M,N,O,P
```

!!! Example "Or Registers."

```
L0,L1,L2,L3
```

Registers

Registers in SASM are identified by unique IDs, with each register having its own.

Registers are addressed as either `references` or `values` via `compile-time functions`, namely ref() and val()

SEE {REFERENCE}

REFERENCE NEEDED `compile-time functions`

REFERENCE NEEDED `registers`

INSTRUCTIONS NOT DOCUMENTED:

```
INST_DONOP,
INST_INVOK,
INST_RETVL,
INST_PUSHR,
INST_SPOPR,
INST_SHUTS,
INST_SETR,
INST_GETR,
INST_CALL,
INST_LOOP,
INST_PUSH,
INST_SPOP,
INST_SWAP,

INST_JMPU,
INST_JMPC,
INST_ANDB,
INST_NOTB,
INST_COPY,
INST_DUPS,
INST_RET,
INST_NOT,
INST_EQI,
INST_GEI,
INST_GTI,
INST_LEI,
INST_LTI,
INST_NEI,
INST_EQU,
INST_GEU,
INST_GTU,
INST_LEU,
INST_LTU,
INST_NEU,
INST_EQF,
INST_GEF,
INST_GTF,
INST_LEF,
INST_LTF,
INST_NEF,
INST_ORB,
INST_XOR,
INST_SHR,
INST_SHL,
INST_I2F,
INST_U2F,
INST_F2I,
INST_F2U,
INST_READ1U,
INST_READ2U,
INST_READ4U,
INST_READ8U,
INST_READ1I,
INST_READ2I,
INST_READ4I,
INST_READ8I,
INST_WRITE1,
INST_WRITE2,
INST_WRITE4,
INST_WRITE8,
```