

# Quickguide of ARM ISA

---

The document is based on ARM v8-A (e.g., Cortex-A72), which supports AArch64.

## Registers

### 31 General-purpose Registers:

X0 - X30: 64-bit

W0 - W30: lower 32-bit only

```
ADD W0, W1, W2
```

X29 is generally used as stack frame pointer (base address of the current stack frame)

X30 is generally used as link register (ret address)

### 32 Floating-point/Vector Registers

Floating-point mode:

Q0-Q31: 128-bit

D0-D31: 64-bit

S0-S31: 32-bit

H0-H31: 16-bit

B0-B31: 8-bit

```
FADD S0, S1, S2
```

Vector mode (multiple independent values in one register):

V0-V31: 128-bit

```
FADD V0.2D, V1.2D, V2.2D
```

```
ADD V0.1D, V1.1D, V2.1D
```

## More

ZXR (64-bit)/WZR (32-bit): zero register

SP: stack pointer

SPSR: saved process status register (similar to EFLAGS in X86-64)

## Calling Convention

Parameter: X0-X7

Result: X0-X1

Caller-saved Registers: X9-X15

Callee-saved Registers: X19-X28

# Instructions

## Instruction Format

Each instruction is 32-bit.

Valid instant numbers: 12 bit (4 bit for right rotation, and 8 bit for value), e.g., 0x00ab0000 is valid; 0x001ab0000 is invalid.

## Load/Store

Load the 32bit value at address X1 into W0

```
LDR W0, [X1]
```

Load the 32bit value at address X1+12 into W0

```
LDR W0, [X1, #12]
```

Pre-index Load: Load the 32bit value at address X1+12 into W0, update X1 as X1+12

```
LDR W0, [X1, #12]!
```

Post-index Load: Load the 32bit value at address X1 into W0, update X1 as X1+12

```
LDR W0, [X1], #12
```

## Integer Arithmetic

## Floating-point Arithmetic

## Branch

## Function Call

BL: branch with linkage (call)

RET

## More

## Reference:

[Armv8-A Instruction Set Architecture](#)