

目录

前言	1.1
ARM概述	1.2
调用规范	1.2.1
ARM寄存器	1.3
寄存器架构	1.3.1
常用寄存器	1.3.2
IP	1.3.2.1
状态寄存器	1.3.2.2
其他寄存器	1.3.3
ARM汇编指令	1.4
指令列表	1.4.1
按字母	1.4.1.1
操作码	1.4.1.2
常用汇编指令	1.4.2
赋值	1.4.2.1
内存操作	1.4.2.2
比较	1.4.2.3
分支跳转	1.4.2.4
条件选择	1.4.2.5
寻址	1.4.2.6
算数运算	1.4.2.7
逻辑运算	1.4.2.8
SVC系统调用	1.4.2.9
其他	1.4.2.10
ARM常见用法和通用规则	1.5
函数调用	1.5.1
跳转指令	1.5.2
条件执行	1.5.3
Pre-index和Post-index	1.5.4
flexible second operand	1.5.5
附录	1.6
X86汇编	1.6.1
参考资料	1.6.2

最流行汇编语言：ARM

- 最新版本： v0.6
- 更新时间： 20221027

简介

整理最流行的汇编语言ARM的相关内容。包括先概述，以及调用规范；ARM寄存器，包括寄存器架构、常用寄存器，比如IP和状态寄存器，以及其他寄存器；再解释ARM汇编指令，包括ARM汇编指令列表，包括按字母表顺序排列的、带操作码的；以及各种常用汇编指令，包括赋值、内存操作、比较、分支跳转、条件选择、寻址、算数运算、逻辑运算、SVC系统调用以及其他；继续整理了ARM常用用法和通用规则，包括函数调用、跳转指令、条件执行、Pre-Index和Post-Index、fleiable second operand；最后附录上X86汇编。

源码+浏览+下载

本书的各种源码、在线浏览地址、多种格式文件下载如下：

HonKit源码

- [crifan/popular_assembly_arm: 最流行汇编语言：ARM](#)

如何使用此HonKit源码去生成发布为电子书

详见：[crifan/honkit_template: demo how to use crifan honkit template and demo](#)

在线浏览

- [最流行汇编语言：ARM book.crifan.org](#)
- [最流行汇编语言：ARM crifan.github.io](#)

离线下载阅读

- [最流行汇编语言：ARM PDF](#)
- [最流行汇编语言：ARM ePub](#)
- [最流行汇编语言：ARM Mobi](#)

版权和用途说明

此电子书教程的全部内容，如无特别说明，均为本人原创。其中部分内容参考自网络，均已备注了出处。
如有版权，请通过邮箱联系我 `admin 艾特 crifan.com`，我会尽快删除。谢谢合作。

各种技术类教程，仅作为学习和研究使用。请勿用于任何非法用途。如有非法用途，均与本人无关。

鸣谢

感谢我的老婆陈雪的包容理解和悉心照料，才使得我 `crifan` 有更多精力去专注技术专研和整理归纳出这些电子书和技术教程，特此鸣谢。

更多其他电子书

本人 `crifan` 还写了其他 100+ 本电子书教程，感兴趣可移步至：

[crifan/crifan_ebook_readme: Crifan的电子书的使用说明](#)

crifan.org, 使用[署名4.0国际\(CC BY 4.0\)协议](#)发布 all right reserved, powered by Gitbook最后更新：
2022-10-27 22:18:15

ARM概述

- 【整理】ARM汇编指令和架构

ARMv8-A has a 64-bit architecture called AArch64

crifan.org, 使用[署名4.0国际\(CC BY 4.0\)协议](#)发布 all right reserved, powered by Gitbook最后更新:
2022-06-15 08:54:42

调用规范

- 调用规范
 - github.com
 - [ARM-software/abi-aa: Application Binary Interface for the Arm® Architecture \(github.com\)](#)
 - aapcs32
 - [abi-aa/aapcs32.rst at main · ARM-software/abi-aa \(github.com\)](#)
 - aapcs64
 - [abi-aa/aapcs64.rst at main · ARM-software/abi-aa \(github.com\)](#)
 - [Releases · ARM-software/abi-aa \(github.com\)](#)
 - ARM64
 - ABI for the Arm 64-bit Architecture
 - Procedure Call Standard for the Arm 64-bit Architecture
 - pdf
 - <https://github.com/ARM-software/abi-aa/releases/download/2021Q1/aapcs64.pdf>
 - html
 - <https://github.com/ARM-software/abi-aa/blob/2bcab1e3b22d55170c563c3c7940134089176746/aapcs64/aapcs64.rst>
 - arm.com
 - 和软件开发相关资料的入口
 - [Develop Software – Arm Developer](#)
 - 各种软件资料
 - [System Architectures | Application Binary Interface \(ABI\) – Arm Developer](#)
 - [Procedure Call Standard for the Arm 64-bit Architecture](#)
- ARM调用规范
 - 【已解决】ARM的ARM64汇编语法和寄存器、调用规范等基础知识
 - 【已解决】32位ARM指令中cond即condition code field定义和语法
 - 【已解决】ARM64中寄存器用法规范
 - 【整理】ARM程序调用规范AAPCS：举例解释
 - 【已解决】IDA中xsp和xbp是什么意思如何定位地址
 - 【已解决】ARM中的寄存器 概述 概览 AAPCS

crifan.org, 使用[署名4.0国际\(CC BY 4.0\)协议](#)发布 all right reserved, powered by Gitbook最后更新：
2022-06-15 09:02:01

ARM寄存器

crifan.org, 使用[署名4.0国际\(CC BY 4.0\)协议](#)发布 all right reserved, powered by Gitbook最后更新：
2022-06-12 14:53:12

寄存器架构

TODO:

【整理】ARM汇编基础知识：寄存器名叫法 20220531

ARM根据位数分32位和64位，每种架构都有很多寄存器，以及对应的特定用法和叫法。

对于ARM寄存器的总体架构，整理如下：

- 在线浏览
 - [ARM寄存器架构 | ProcessOn免费在线作图](#)
- 本地查看

◦

crifan.org, 使用[署名4.0国际\(CC BY 4.0\)协议](#)发布 all right reserved, powered by Gitbook最后更新：
2022-06-15 08:54:01

常用寄存器

TODO:把x0到x31 把其中比较常用的，有需要单独解释的寄存器，整理出来

crifan.org, 使用[署名4.0国际\(CC BY 4.0\)协议](#)发布 all right reserved, powered by Gitbook最后更新：
2022-06-15 08:49:48

IP

- 【已解决】ARM中的R12寄存器IP

crifan.org, 使用[署名4.0国际\(CC BY 4.0\)协议](#)发布 all right reserved, powered by Gitbook最后更新:
2022-06-15 08:50:08

状态寄存器

- 【已解决】ARM汇编指令：CPSR当前程序状态寄存器和标志位

crifan.org, 使用[署名4.0国际\(CC BY 4.0\)协议](#)发布 all right reserved, powered by Gitbook最后更新：
2022-06-15 08:53:20

其他寄存器

TODO:

- 【已解决】ARM汇编寄存器: TPIDRRO_EL0
- 【已解决】ARM寄存器: q0 q1
- 【已解决】ARM寄存器: q0 q1 v0 S0 H0 B0

crifan.org, 使用[署名4.0国际\(CC BY 4.0\)协议](#)发布 all right reserved, powered by Gitbook最后更新:
2022-10-27 22:00:13

ARM汇编指令

crifan.org, 使用[署名4.0国际\(CC BY 4.0\)协议](#)发布 all right reserved, powered by Gitbook最后更新：
2022-06-12 14:53:36

ARM汇编指令列表

- 资料来源
 - 很全的指令的列表
 - [A Guide to ARM64 / AArch64 Assembly on Linux with Shellcodes and Cryptography | modexp \(wordpress.com\)](https://developer.arm.com/documentation/ddi0596/2020-12/Base-Instructions)
 - 官网的专业介绍
 - [Arm A64 Instruction Set Architecture](https://developer.arm.com/documentation/ddi0596/2020-12/Base-Instructions)
- ARM 64 汇编指令集
 - 包含
 - 核心指令
 - NEON和FPU指令集
 - 详见
 - <https://link.springer.com/content/pdf/bbm%3A978-1-4842-5881-1%2F1.pdf>

The screenshot shows a detailed view of the Arm A64 Instruction Set Architecture documentation. The main content area is titled "A64 -- Base Instructions (alphabetic order)". It lists various instructions such as ADC, ADDS, ADD, etc., each with a brief description. The left sidebar has a "DOCUMENT TABLE OF CONTENTS" section with a tree view of the instruction set architecture. The top navigation bar includes links for IP Explorer, Documentation, Downloads, Community, and Support.

ARM 64 Instruction Set - Appendix A.pdf

使用 WPS Office 打开

APPENDIX A

The ARM Instruction Set

This appendix lists the ARM 64-bit instruction in two sections: first, the core instruction set, then the NEON and FPU instructions. There is a brief description of each instruction:

{S} after an instruction indicates you can optionally set the condition flags.

† means the instruction is an alias.

ARM 64-Bit Core Instructions

Instruction	Description
ADC{S}	Add with carry
ADD{S}	Add
ADDG	Add with tag
ADR	Form PC relative address
ADRP	Form PC relative address to 4KB page
AND{S}	Bitwise AND

(continued)

© Stephen Smith 2020
S. Smith, *Programming with 64-Bit ARM Assembly Language*,
<https://doi.org/10.1007/978-1-4842-5881-1>

367

按字母排序的ARM汇编指令列表

- 指令列表
 - A64 -- Base Instructions (alphabetic order)

```

ADC: Add with Carry.
ADCS: Add with Carry, setting flags.
ADD (extended register): Add (extended register).
ADD (immediate): Add (immediate).
ADD (shifted register): Add (shifted register).
ADDG: Add with Tag.
ADDS (extended register): Add (extended register), setting flags.
ADDS (immediate): Add (immediate), setting flags.
ADDS (shifted register): Add (shifted register), setting flags.
ADR: Form PC relative address.
ADRP: Form PC relative address to 4KB page.
AND (immediate): Bitwise AND (immediate).
AND (shifted register): Bitwise AND (shifted register).
ANDS (immediate): Bitwise AND (immediate), setting flags.
ANDS (shifted register): Bitwise AND (shifted register), setting flags.
ASR (immediate): Arithmetic Shift Right (immediate): an alias of SBFM.
ASR (register): Arithmetic Shift Right (register): an alias of ASRV.
ASRV: Arithmetic Shift Right Variable.
AT: Address Translate: an alias of SYS.
AUTDA, AUTDZA: Authenticate Data address, using key A.
AUTDB, AUTDZB: Authenticate Data address, using key B.
AUTIA, AUTIA1716, AUTIASP, AUTIAZ, AUTIZA: Authenticate Instruction address, using key A.
AUTIB, AUTIB1716, AUTIBSP, AUTIBZ, AUTIZB: Authenticate Instruction address, using key B.
AXFLAG: Convert floating point condition flags from Arm to external format.
B: Branch.
B.cond: Branch conditionally.
BFC: Bitfield Clear: an alias of BFM.
BFI: Bitfield Insert: an alias of BFM.
BFM: Bitfield Move.
BFXIL: Bitfield extract and insert at low end: an alias of BFM.
BIC (shifted register): Bitwise Bit Clear (shifted register).
BICS (shifted register): Bitwise Bit Clear (shifted register), setting flags.
BL: Branch with Link.
BLR: Branch with Link to Register.
BLRAA, BLRAAZ, BLRAB, BLRABZ: Branch with Link to Register, with pointer authentication.
BR: Branch to Register.
BRAA, BRAAZ, BRAB, BRABZ: Branch to Register, with pointer authentication.
BRK: Breakpoint instruction.
BTI: Branch Target Identification.
CAS, CASA, CASAL, CASL: Compare and Swap word or doubleword in memory.
CASB, CASAB, CASALB, CASLB: Compare and Swap byte in memory.
CASH, CASAH, CASALH, CASLH: Compare and Swap halfword in memory.
CASP, CASPA, CASPAL, CASPL: Compare and Swap Pair of words or doublewords in memory.
CBNZ: Compare and Branch on Nonzero.
CBZ: Compare and Branch on Zero.
CCMN (immediate): Conditional Compare Negative (immediate).
CCMN (register): Conditional Compare Negative (register).
CCMP (immediate): Conditional Compare (immediate).

```

CCMP (register): Conditional Compare (register).
CFINV: Invert Carry Flag.
CFP: Control Flow Prediction Restriction by Context: an alias of SYS.
CINC: Conditional Increment: an alias of CSINC.
CINV: Conditional Invert: an alias of CSINV.
CLREX: Clear Exclusive.
CLS: Count Leading Sign bits.
CLZ: Count Leading Zeros.
CMN (extended register): Compare Negative (extended register): an alias of ADDS (extended register).
CMN (immediate): Compare Negative (immediate): an alias of ADDS (immediate).
CMN (shifted register): Compare Negative (shifted register): an alias of ADDS (shifted register).
CMP (extended register): Compare (extended register): an alias of SUBS (extended register).
CMP (immediate): Compare (immediate): an alias of SUBS (immediate).
CMP (shifted register): Compare (shifted register): an alias of SUBS (shifted register).
CMPP: Compare with Tag: an alias of SUBPS.
CNEG: Conditional Negate: an alias of CSNEG.
CPP: Cache Prefetch Prediction Restriction by Context: an alias of SYS.
CRC32B, CRC32H, CRC32W, CRC32X: CRC32 checksum.
CRC32CB, CRC32CH, CRC32CW, CRC32CX: CRC32C checksum.
CSDB: Consumption of Speculative Data Barrier.
CSEL: Conditional Select.
CSET: Conditional Set: an alias of CSINC.
CSETM: Conditional Set Mask: an alias of CSINV.
CSINC: Conditional Select Increment.
CSINV: Conditional Select Invert.
CSNEG: Conditional Select Negation.
DC: Data Cache operation: an alias of SYS.
DCPS1: Debug Change PE State to EL1..
DCPS2: Debug Change PE State to EL2..
DCPS3: Debug Change PE State to EL3.
DGH: Data Gathering Hint.
DMB: Data Memory Barrier.
DRPS: Debug restore process state.
DSB: Data Synchronization Barrier.
DVP: Data Value Prediction Restriction by Context: an alias of SYS.
EON (shifted register): Bitwise Exclusive OR NOT (shifted register): an alias of EOR.
EOR (immediate): Bitwise Exclusive OR (immediate): an alias of EON.
EOR (shifted register): Bitwise Exclusive OR (shifted register): an alias of EON.
ERET: Exception Return.
ERETAA, ERETAB: Exception Return, with pointer authentication.
ESB: Error Synchronization Barrier.
EXTR: Extract register.
GMI: Tag Mask Insert.
HINT: Hint instruction.
HLT: Halt instruction.
HVC: Hypervisor Call.
IC: Instruction Cache operation: an alias of SYS.
IRG: Insert Random Tag.
ISB: Instruction Synchronization Barrier.
LD64B: Single copy Atomic 64 byte Load.
LDADD, LDADDA, LDADDAL, LDADDL: Atomic add on word or doubleword in memory.
LDADDB, LDADDAB, LDADDALB, LDADDLB: Atomic add on byte in memory.
LDADDH, LDADDAH, LDADDALH, LDADDLH: Atomic add on halfword in memory.
LDAPR: Load Acquire RCpc Register.

LDAPRB: Load Acquire RCpc Register Byte.
LDAPRH: Load Acquire RCpc Register Halfword.
LDAPUR: Load Acquire RCpc Register (unscaled).
LDAPURB: Load Acquire RCpc Register Byte (unscaled).
LDAPURH: Load Acquire RCpc Register Halfword (unscaled).
LDAPURSB: Load Acquire RCpc Register Signed Byte (unscaled).
LDAPURSH: Load Acquire RCpc Register Signed Halfword (unscaled).
LDAPURSW: Load Acquire RCpc Register Signed Word (unscaled).
LDAR: Load Acquire Register.
LDARB: Load Acquire Register Byte.
LDARH: Load Acquire Register Halfword.
LDAXP: Load Acquire Exclusive Pair of Registers.
LDAPXR: Load Acquire Exclusive Register.
LDAXRB: Load Acquire Exclusive Register Byte.
LDAXRH: Load Acquire Exclusive Register Halfword.
LDCLR, LDCLRA, LDCLRAL, LDCLRL: Atomic bit clear on word or doubleword in memory.
LDCLRB, LDCLRAB, LDCLRALB, LDCLRLB: Atomic bit clear on byte in memory.
LDCLRH, LDCLRAH, LDCLRALH, LDCLRLH: Atomic bit clear on halfword in memory.
LDEOR, LDEORA, LDEORAL, LDEORL: Atomic exclusive OR on word or doubleword in memory.
LDEORB, LDEORAB, LDEORALB, LDEORLB: Atomic exclusive OR on byte in memory.
LDEORH, LDEORAH, LDEORALH, LDEORLH: Atomic exclusive OR on halfword in memory.
LDG: Load Allocation Tag.
LDGM: Load Tag Multiple.
LDLAR: Load LOAcquire Register.
LDLARB: Load LOAcquire Register Byte.
LDLARH: Load LOAcquire Register Halfword.
LDNP: Load Pair of Registers, with non temporal hint.
LDP: Load Pair of Registers.
LDPSW: Load Pair of Registers Signed Word.
LDR (immediate): Load Register (immediate).
LDR (literal): Load Register (literal).
LDR (register): Load Register (register).
LDRAA, LDRAB: Load Register, with pointer authentication.
LDRB (immediate): Load Register Byte (immediate).
LDRB (register): Load Register Byte (register).
LDRH (immediate): Load Register Halfword (immediate).
LDRH (register): Load Register Halfword (register).
LDRSB (immediate): Load Register Signed Byte (immediate).
LDRSB (register): Load Register Signed Byte (register).
LDRSH (immediate): Load Register Signed Halfword (immediate).
LDRSH (register): Load Register Signed Halfword (register).
LDRSW (immediate): Load Register Signed Word (immediate).
LDRSW (literal): Load Register Signed Word (literal).
LDRSW (register): Load Register Signed Word (register).
LDSET, LDSETA, LDSETAL, LDSETL: Atomic bit set on word or doubleword in memory.
LDSETB, LDSETAB, LDSETALB, LDSETLB: Atomic bit set on byte in memory.
LDSETH, LDSETAH, LDSETALH, LDSETLH: Atomic bit set on halfword in memory.
LDSMAX, LDSMAXA, LDSMAXAL, LDSMAXL: Atomic signed maximum on word or doubleword in memory.
LDSMAXB, LDSMAXAB, LDSMAXALB, LDSMAXLB: Atomic signed maximum on byte in memory.
LDSMAXH, LDSMAXAH, LDSMAXALH, LDSMAXLH: Atomic signed maximum on halfword in memory.
LDSMIN, LDSMINA, LDSMINAL, LDSMINL: Atomic signed minimum on word or doubleword in memory.
LDSMINB, LDSMINAB, LDSMINALB, LDSMINLB: Atomic signed minimum on byte in memory.
LDSMINH, LDSMINAH, LDSMINALH, LDSMINLH: Atomic signed minimum on halfword in memory.
LDTR: Load Register (unprivileged).
LDTRB: Load Register Byte (unprivileged).
LDTRH: Load Register Halfword (unprivileged).

LDTRSB: Load Register Signed Byte (unprivileged).
LDTRSH: Load Register Signed Halfword (unprivileged).
LDTRSW: Load Register Signed Word (unprivileged).
LDUMAX, LDUMAXA, LDUMAXAL, LDUMAXL: Atomic unsigned maximum on word or doubleword in memory.
LDUMAXB, LDUMAXAB, LDUMAXALB, LDUMAXLB: Atomic unsigned maximum on byte in memory.
LDUMAXH, LDUMAXAH, LDUMAXALH, LDUMAXLH: Atomic unsigned maximum on halfword in memory.
LDUMIN, LDUMINA, LDUMINAL, LDUMINL: Atomic unsigned minimum on word or doubleword in memory.
LDUMINB, LDUMINAB, LDUMINALB, LDUMINLB: Atomic unsigned minimum on byte in memory.
LDUMINH, LDUMINAH, LDUMINALH, LDUMINLH: Atomic unsigned minimum on halfword in memory.
LDUR: Load Register (unscaled).
LDURB: Load Register Byte (unscaled).
LDURH: Load Register Halfword (unscaled).
LDURSB: Load Register Signed Byte (unscaled).
LDURSH: Load Register Signed Halfword (unscaled).
LDURSW: Load Register Signed Word (unscaled).
LDXP: Load Exclusive Pair of Registers.
LDXR: Load Exclusive Register.
LDXRB: Load Exclusive Register Byte.
LDXRH: Load Exclusive Register Halfword.
LSL (immediate): Logical Shift Left (immediate): an alias of UBFM.
LSL (register): Logical Shift Left (register): an alias of LSLV.
LSLV: Logical Shift Left Variable.
LSR (immediate): Logical Shift Right (immediate): an alias of UBFM.
LSR (register): Logical Shift Right (register): an alias of LSRV.
LSRV: Logical Shift Right Variable.
MADD: Multiply Add.
MNEG: Multiply Negate: an alias of MSUB.
MOV (bitmask immediate): Move (bitmask immediate): an alias of ORR (immediate).
MOV (inverted wide immediate): Move (inverted wide immediate): an alias of MOVN.
MOV (register): Move (register): an alias of ORR (shifted register).
MOV (to from SP): Move between register and stack pointer: an alias of ADD (immediate).
MOV (wide immediate): Move (wide immediate): an alias of MOVZ.
MOVK: Move wide with keep.
MOVN: Move wide with NOT.
MOVZ: Move wide with zero.
MRS: Move System Register.
MSR (immediate): Move immediate value to Special Register.
MSR (register): Move general purpose register to System Register.
MSUB: Multiply Subtract.
MUL: Multiply: an alias of MADD.
MVN: Bitwise NOT: an alias of ORN (shifted register).
NEG (shifted register): Negate (shifted register): an alias of SUB (shifted register).
NEGS: Negate, setting flags: an alias of SUBS (shifted register).
NGC: Negate with Carry: an alias of SBC.
NGCS: Negate with Carry, setting flags: an alias of SBCS.
NOP: No Operation.
ORN (shifted register): Bitwise OR NOT (shifted register).
ORR (immediate): Bitwise OR (immediate).
ORR (shifted register): Bitwise OR (shifted register).
PACDA, PACDZA: Pointer Authentication Code for Data address, using key A.
PACDB, PACDZB: Pointer Authentication Code for Data address, using key B.
PACGA: Pointer Authentication Code, using Generic key.
PACIA, PACIA1716, PACIASP, PACIAZ, PACIZA: Pointer Authentication Code for Instruction address, using key A.
PACIB, PACIB1716, PACIBSP, PACIBZ, PACIZB: Pointer Authentication Code for Instruction address, using key B.

PRFM (immediate): Prefetch Memory (immediate).
PRFM (literal): Prefetch Memory (literal).
PRFM (register): Prefetch Memory (register).
PRFUM: Prefetch Memory (unscaled offset).
PSB CSYNC: Profiling Synchronization Barrier.
PSSBB: Physical Speculative Store Bypass Barrier.
RBIT: Reverse Bits.
RET: Return from subroutine.
RETA, RETAB: Return from subroutine, with pointer authentication.
REV: Reverse Bytes.
REV16: Reverse bytes in 16 bit halfwords.
REV32: Reverse bytes in 32 bit words.
REV64: Reverse Bytes: an alias of REV.
RMIIF: Rotate, Mask Insert Flags.
ROR (immediate): Rotate right (immediate): an alias of EXTR.
ROR (register): Rotate Right (register): an alias of RORV.
RORV: Rotate Right Variable.
SB: Speculation Barrier.
SBC: Subtract with Carry.
SCBS: Subtract with Carry, setting flags.
SBFIZ: Signed Bitfield Insert in Zero: an alias of SBFM.
SBFM: Signed Bitfield Move.
SBFX: Signed Bitfield Extract: an alias of SBFM.
SDIV: Signed Divide.
SETF8, SETF16: Evaluation of 8 or 16 bit flag values.
SEV: Send Event.
SEVL: Send Event Local.
SMADDL: Signed Multiply Add Long.
SMC: Secure Monitor Call.
SMNEGJ: Signed Multiply Negate Long: an alias of SMSUBL.
SMSUBL: Signed Multiply Subtract Long.
SMULH: Signed Multiply High.
SMULL: Signed Multiply Long: an alias of SMADDL.
SSBB: Speculative Store Bypass Barrier.
ST2G: Store Allocation Tags.
ST64B: Single copy Atomic 64 byte Store without Return.
ST64BV: Single copy Atomic 64 byte Store with Return.
ST64BV0: Single copy Atomic 64 byte EL0 Store with Return.
STADD, STADDL: Atomic add on word or doubleword in memory, without return: an alias of LDADD, LDADDL.
STADDB, STADDLB: Atomic add on byte in memory, without return: an alias of LDADDB, LDADDAB, LDADLB, LDADDLB.
STADDH, STADDLH: Atomic add on halfword in memory, without return: an alias of LDADDH, LDADDAH, LDADDALH, LDADDLH.
STCLR, STCLRL: Atomic bit clear on word or doubleword in memory, without return: an alias of LDCLR, LDCLRA, LDCLRAL, LDCLRL.
STCLRB, STCLRLB: Atomic bit clear on byte in memory, without return: an alias of LDCLRB, LDCLRA, LDCLRALB, LDCLRLB.
STCLRH, STCLRLH: Atomic bit clear on halfword in memory, without return: an alias of LDCLRH, LDCLRAH, LDCLRALH, LDCLRLH.
STEOR, STEORL: Atomic exclusive OR on word or doubleword in memory, without return: an alias of LDEOR, LDEORA, LDEORAL, LDEORL.
STEORB, STEORLB: Atomic exclusive OR on byte in memory, without return: an alias of LDEORB, LDEORAB, LDEORALB, LDEORLB.
STEORH, STEORLH: Atomic exclusive OR on halfword in memory, without return: an alias of LDEORH, LDEORAH, LDEORALH, LDEORLH.

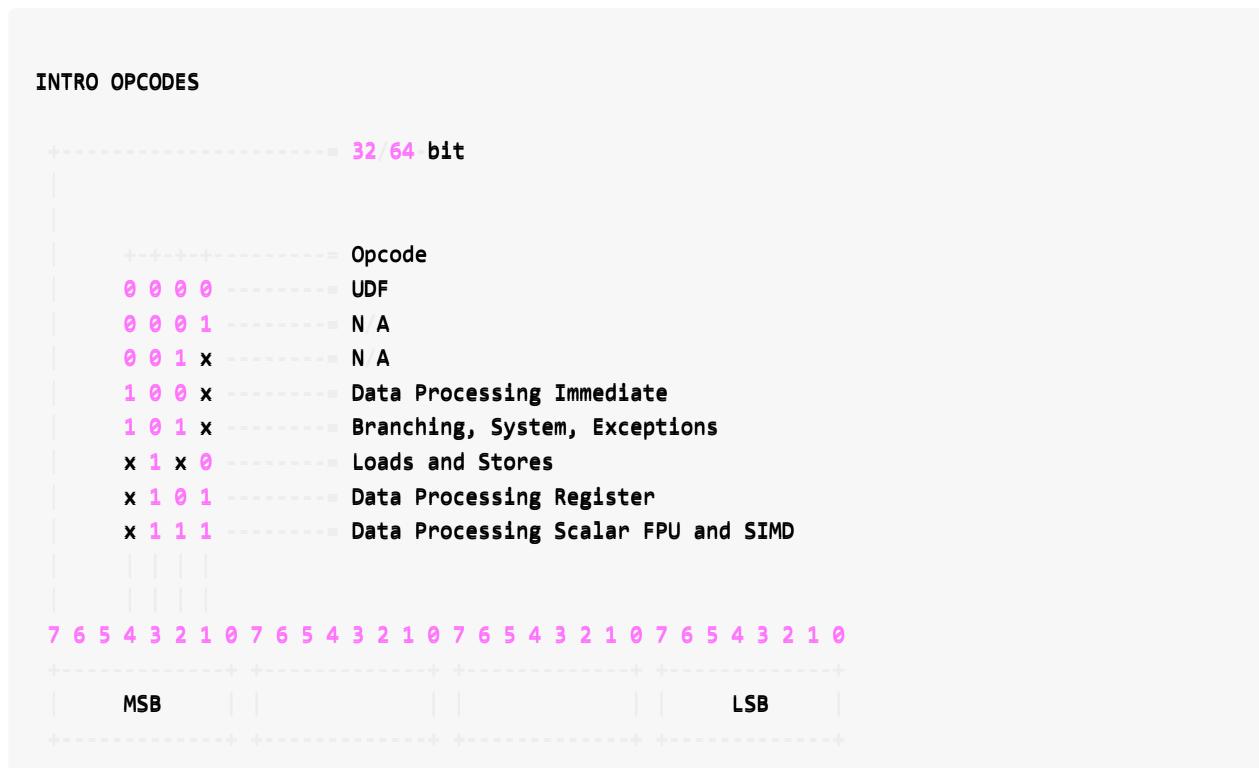
STG: Store Allocation Tag.
STGM: Store Tag Multiple.
STGP: Store Allocation Tag and Pair of registers.
STLLR: Store Lorelease Register.
STLLRB: Store Lorelease Register Byte.
STLLRH: Store Lorelease Register Halfword.
STLR: Store Release Register.
STLRB: Store Release Register Byte.
STLRH: Store Release Register Halfword.
STLUR: Store Release Register (unscaled).
STLURB: Store Release Register Byte (unscaled).
STLURH: Store Release Register Halfword (unscaled).
STLXP: Store Release Exclusive Pair of registers.
STLXR: Store Release Exclusive Register.
STLXRB: Store Release Exclusive Register Byte.
STLXRH: Store Release Exclusive Register Halfword.
STNP: Store Pair of Registers, with non temporal hint.
STP: Store Pair of Registers.
STR (immediate): Store Register (immediate).
STR (register): Store Register (register).
STRB (immediate): Store Register Byte (immediate).
STRB (register): Store Register Byte (register).
STRH (immediate): Store Register Halfword (immediate).
STRH (register): Store Register Halfword (register).
STSET, STSETL: Atomic bit set on word or doubleword in memory, without return: an alias of LDSE T, LDSETA, LDSETAL, LDSETL.
STSETB, STSETLB: Atomic bit set on byte in memory, without return: an alias of LDSETB, LDSETAB, LDSETALB, LDSETLB.
STSETH, STSETLH: Atomic bit set on halfword in memory, without return: an alias of LDSETH, LDSE TAH, LDSETALH, LDSETLH.
STS MAX, STSMAXL: Atomic signed maximum on word or doubleword in memory, without return: an alias of LDSMAX, LDSMAXA, LDSMAXAL, LDSMAXL.
STS MAXB, STSMAXLB: Atomic signed maximum on byte in memory, without return: an alias of LDSMAXB , LDSMAXAB, LDSMAXALB, LDSMAXLB.
STS MAXH, STSMAXLH: Atomic signed maximum on halfword in memory, without return: an alias of LDS MAXH, LDSMAXAH, LDSMAXALH, LDSMAXLH.
STS MIN, STSMINL: Atomic signed minimum on word or doubleword in memory, without return: an alias of LDSMIN, LDSMINA, LDSMINAL, LDSMINL.
STS MINB, STSMINLB: Atomic signed minimum on byte in memory, without return: an alias of LDSMINB , LDSMINAB, LDSMINALB, LDSMINLB.
STS MINH, STSMINLH: Atomic signed minimum on halfword in memory, without return: an alias of LDS MINH, LDSMINAH, LDSMINALH, LDSMINLH.
STTR: Store Register (unprivileged).
STTRB: Store Register Byte (unprivileged).
STTRH: Store Register Halfword (unprivileged).
STUMAX, STUMAXL: Atomic unsigned maximum on word or doubleword in memory, without return: an alias of LDUMAX, LDUMAXA, LDUMAXAL, LDUMAXL.
STUMAXB, STUMAXLB: Atomic unsigned maximum on byte in memory, without return: an alias of LDUM AXB, LDUMAXAB, LDUMAXALB, LDUMAXLB.
STUMAXH, STUMAXLH: Atomic unsigned maximum on halfword in memory, without return: an alias of LDUMAXH, LDUMAXAH, LDUMAXALH, LDUMAXLH.
STUMIN, STUMINL: Atomic unsigned minimum on word or doubleword in memory, without return: an alias of LDUMIN, LDUMINA, LDUMINAL, LDUMINL.
STUMINB, STUMINLB: Atomic unsigned minimum on byte in memory, without return: an alias of LDUMI NB, LDUMINAB, LDUMINALB, LDUMINLB.
STUMINH, STUMINLH: Atomic unsigned minimum on halfword in memory, without return: an alias of LDUMINH, LDUMINLH.

DUMINH, LDUMINAH, LDUMINALH, LDUMINLH.
STUR: Store Register (unscaled).
STURB: Store Register Byte (unscaled).
STURH: Store Register Halfword (unscaled).
STXP: Store Exclusive Pair of registers.
STXR: Store Exclusive Register.
STXRB: Store Exclusive Register Byte.
STXRH: Store Exclusive Register Halfword.
STZ2G: Store Allocation Tags, Zeroing.
STZG: Store Allocation Tag, Zeroing.
STZGM: Store Tag and Zero Multiple.
SUB (extended register): Subtract (extended register).
SUB (immediate): Subtract (immediate).
SUB (shifted register): Subtract (shifted register).
SUBG: Subtract with Tag.
SUBP: Subtract Pointer.
SUBPS: Subtract Pointer, setting Flags.
SUBS (extended register): Subtract (extended register), setting flags.
SUBS (immediate): Subtract (immediate), setting flags.
SUBS (shifted register): Subtract (shifted register), setting flags.
SVC: Supervisor Call.
SWP, SWPA, SWPAL, SWPL: Swap word or doubleword in memory.
SWPB, SWPAB, SWPALB, SWPLB: Swap byte in memory.
SWPH, SWPAH, SWPALH, SWPLH: Swap halfword in memory.
SXTB: Signed Extend Byte: an alias of SBFM.
SXTH: Sign Extend Halfword: an alias of SBFM.
SXTW: Sign Extend Word: an alias of SBFM.
SYS: System instruction.
SYSL: System instruction with result.
TBNZ: Test bit and Branch if Nonzero.
TBZ: Test bit and Branch if Zero.
TLBI: TLB Invalidate operation: an alias of SYS.
TSB CSYNC: Trace Synchronization Barrier.
TST (immediate): Test bits (immediate): an alias of ANDS (immediate).
TST (shifted register): Test (shifted register): an alias of ANDS (shifted register).
UBFIZ: Unsigned Bitfield Insert in Zero: an alias of UBFM.
UBFM: Unsigned Bitfield Move.
UBFX: Unsigned Bitfield Extract: an alias of UBFM.
UDF: Permanently Undefined.
UDIV: Unsigned Divide.
UMADDL: Unsigned Multiply Add Long.
UMNEGL: Unsigned Multiply Negate Long: an alias of UMSUBL.
UMSUBL: Unsigned Multiply Subtract Long.
UMULH: Unsigned Multiply High.
UMULL: Unsigned Multiply Long: an alias of UMADDL.
UXTB: Unsigned Extend Byte: an alias of UBFM.
UXTH: Unsigned Extend Halfword: an alias of UBFM.
WFE: Wait For Event.
WFET: Wait For Event with Timeout.
WFI: Wait For Interrupt.
WFIT: Wait For Interrupt with Timeout.
XAFLAG: Convert floating point condition flags from external format to Arm format.
XPACD, XPACI, XPAACLRI: Strip Pointer Authentication Code.
YIELD: YIELD.

带操作码的ARM汇编指令列表

- 来源
 - [ARM \(groupoid.github.io\)](https://groupoid.github.io)

操作码格式介绍



普通指令

```

0 0 0 0 0 0 0 0 0 0 0 % C6.2.312 UDF
0 x 0 0 1 0 0 0 0 0 0 % C6.2.286 STXRB % C6.2.253 STLXRB % C6.2.254 STLXRH
0 x 0 0 1 0 0 0 0 i 1 % C6.2.039 CASP, CASPA, CASPAL, CASPL
0 x 0 0 1 0 0 0 0 1 0 % C6.2.106 LDAXRH % C6.2.105 LDAXRB % C6.2.163 LDXRB
0 0 0 0 1 0 0 0 1 0 0 % C6.2.246 STLRB
0 0 0 0 1 0 0 0 1 1 0 % C6.2.101 LDARB % C6.2.113 LDLARB
0 0 0 0 1 0 0 0 1 0 0 % C6.2.242 STLLRB % C6.2.243 STLLRH
0 0 0 0 1 0 0 0 1 i 1 % C6.2.037 CASB, CASAB, CASALB, CASLB
0 1 0 0 1 0 0 0 0 1 0 % C6.2.164 LDXRH
0 1 0 0 1 0 0 0 1 0 0 % C6.2.247 STLRH
0 1 0 0 1 0 0 0 1 1 0 % C6.2.102 LDARH % C6.2.114 LDLARH
0 1 0 0 1 0 0 0 1 i 1 % C6.2.038 CASH, CASAH, CASALH, CASLH
0 1 1 0 1 0 0 i 1 1 _ % C6.2.118 LDPSW
x 0 1 0 1 0 0 0 0 0 _ % C6.2.255 STNP
x 0 1 0 1 0 0 0 0 1 _ % C6.2.116 LDNP
x 0 1 0 1 0 0 0 1 0 _ % C6.2.256 STP
x 0 1 0 1 0 0 0 1 1 _ % C6.2.117 LDP
1 x 0 0 1 0 0 0 0 0 0 % C6.2.252 STLXR % C6.2.285 STXR
1 x 0 0 1 0 0 0 0 0 1 % C6.2.251 STLXP % C6.2.284 STXP

```

```

1 x 0 0 1 0 0 0 0 1 0 % C6.2.104 LDAXR % C6.2.162 LDXR
1 x 0 0 1 0 0 0 0 1 1 % C6.2.103 LDAXP % C6.2.161 LDXP
1 x 0 0 1 0 0 0 1 0 0 % C6.2.244 STLLR % C6.2.245 STLR
1 x 0 0 1 0 0 0 1 1 0 % C6.2.100 LDAR % C6.2.115 LDLAR
1 x 0 0 1 0 0 0 1 i 1 % C6.2.040 CAS, CASA, CASAL, CASL
x 0 0 0 1 0 1 0 s s 0 % C6.2.012 AND (shifted register)
x 0 0 0 1 0 1 0 s s 1 % C6.2.029 BIC (shifted register)
x 0 1 0 1 0 1 0 0 0 % C6.2.177 MOV (register)
x 0 1 0 1 0 1 0 s s 1 % C6.2.186 MVN % C6.2.192 ORN (shifted register)
x 1 0 0 1 0 1 0 s s 0 % C6.2.078 EOR (shifted register)
x 1 0 0 1 0 1 0 s s 1 % C6.2.076 EON (shifted register)
x 1 1 0 1 0 1 0 s s 0 % C6.2.014 ANDS (shifted register) % C6.2.308 TST (shifted register)
x 1 1 0 1 0 1 0 s s 1 % C6.2.030 BICS (shifted register)
x 0 0 0 1 0 1 1 0 0 % C6.2.001 ADC % C6.2.003 ADD (extended register)
x 0 0 0 1 0 1 1 s s 0 % C6.2.005 ADD (shifted register)
x 0 1 0 1 0 1 1 0 0 % C6.2.006 ADDS (extended register) % C6.2.53 CMN (extended register)
x 0 1 0 1 0 1 1 s s 0 % C6.2.008 ADDS (shifted register) % C6.2.55 CMN (shifted register)
x 1 0 0 1 0 1 1 0 0 % C6.2.288 SUB (extended register)
x 1 0 0 1 0 1 1 s s 0 % C6.2.290 SUB (shifted register) % C6.2.187 NEG (shifted register)
x 1 1 0 1 0 1 1 0 0 % C6.2.056 CMP (extended register) % C6.2.291 SUBS (extended register)
x 1 1 0 1 0 1 1 s s 0 % C6.2.058 CMP (shifted register) % C6.2.293 SUBS (shifted register) % C6
.C6.2.188 NEGS
x 1 i 1 0 0 0 0 _ _ _ % C6.2.009 ADR % C6.2.010 ADRP
x 0 0 1 0 0 0 1 0 s _ % C6.2.004 ADD (immediate) % C6.2.173 MOV (to from SP)
x 1 0 1 0 0 0 1 0 s _ % C6.2.289 SUB (immediate)
x 0 1 1 0 0 0 1 0 s _ % C6.2.007 ADDS (immediate) % C6.2.54 CMN (immediate)
x 1 1 1 0 0 0 1 0 s _ % C6.2.057 CMP (immediate) % C6.2.292 SUBS (immediate)
x 0 0 1 0 0 1 0 0 n _ % C6.2.011 AND (immediate)
x 0 0 1 0 0 1 0 1 h w % C6.2.174 MOV (inverted wide immediate) % C6.2.179 MOVN
                           % C6.2.221 SBFX % C6.2.298 SXTB % C6.2.299 SXTH % C6.2.300 SXTW
x 0 0 1 0 0 1 1 0 n _ % C6.2.016 ASR (immediate) % C6.2.219 SBFIZ % C6.2.220 SBFM
                           % C6.2.221 SBFX % C6.2.298 SXTB % C6.2.299 SXTH % C6.2.300 SXTW
x 0 0 1 0 0 1 1 1 n 0 % C6.2.082 EXTR % C6.2.214 ROR (immediate)
x 0 1 1 0 0 1 0 0 n _ % C6.2.176 MOV (bitmask immediate) % C6.2.193 ORR (immediate)
x 0 1 1 0 0 1 1 0 n _ % C6.2.025 BFC % C6.2.26 BFI % C6.2.27 BFM % C6.2.28 BFXIL
x 1 0 1 0 0 1 0 0 n _ % C6.2.077 EOR (immediate)
x 1 0 1 0 0 1 0 1 h w % C6.2.175 MOV (wide immediate) % C6.2.180 MOVZ
x 1 0 1 0 0 1 1 0 n _ % C6.2.166 LSL (immediate) % C6.2.169 LSR (immediate) % C6.2.309 UBFIZ %
C6.2.310 UBFM
                           % C6.2.311 UBFX % C6.2.319 UXTB % C6.2.320 UXTH
x 1 1 1 0 0 1 0 0 n _ % C6.2.013 ANDS (immediate) % C6.2.307 TST (immediate)
x 1 1 1 0 0 1 0 1 h w % C6.2.178 MOVK
0 0 0 1 0 1 _ _ _ _ % C6.2.024 B
0 1 0 1 0 1 0 0 _ _ _ % C6.2.023 B.cond
x 0 1 1 0 1 0 0 _ _ _ % C6.2.042 CBZ
x 0 1 1 0 1 0 1 _ _ _ % C6.2.041 CBNZ
1 0 0 1 0 1 _ _ _ _ % C6.2.031 BL
1 1 0 1 0 1 0 0 0 0 % C6.2.085 HVC % C6.2.227 SMC % C6.2.294 SVC
1 1 0 1 0 1 0 0 0 1 % C6.2.036 BRK
1 1 0 1 0 1 0 0 0 1 0 % C6.2.084 HLT
1 1 0 1 0 1 0 0 1 0 1 % C6.2.070 DCPS1 % C6.2.071 DCPS2 % C6.2.072 DCPS3 % C6.2.073 DMB
1 1 0 1 0 1 0 1 0 0 0 % C6.2.018 AT % C6.2.047 CFINV % C6.2.050 CLREX
                           % C6.2.021 AUTIA, AUTIA1716, AUTIASP, AUTIAZ, AUTIZA
                           % C6.2.022 AUTIB, AUTIB1716, AUTIBSP, AUTIBZ, AUTIZB
                           % C6.2.062 CSDB % C6.2.069 DC % C6.2.075 DSB % C6.2.081 ESB
                           % C6.2.083 HINT % C6.2.086 IC % C6.2.087 ISB % C6.2.182 MSR (immediate)

```

PACIZA	% C6.2.183 MSR (register) % C6.2.198 PACIA, PACIA1716, PACIASP, PACIAZ, % C6.2.204 PSB CSYNC % C6.2.205 PSSBB % C6.2.224 SEV % C6.2.225 SEVL % C6.2.232 SSBB % C6.2.301 SYS % C6.2.305 TLBI % C6.2.306 TSB CSYNC % C6.2.321 WFE % C6.2.322 WFI % C6.2.324 YIELD % C6.2.191 NOP 1 1 0 1 0 1 0 1 0 0 1 % C6.2.181 MRS % C6.2.302 SYSL x 0 1 1 0 1 1 0 _ _ _ % C6.2.304 TBZ x 0 1 1 0 1 1 1 _ _ _ % C6.2.303 TBNZ 1 1 0 1 0 1 1 0 0 0 0 % C6.2.034 BR 1 1 0 1 0 1 1 0 0 0 1 % C6.2.032 BLR 1 1 0 1 0 1 1 0 0 1 0 % C6.2.207 RET % C6.2.208 RETAA, RETAB 1 1 0 1 0 1 1 0 1 0 0 % C6.2.079 ERET % C6.2.080 ERETTAA, ERETAB 1 1 0 1 0 1 1 0 1 0 1 % C6.2.074 DRPS 1 1 0 1 0 1 1 z 0 0 0 % C6.2.035 BRAA, BRAAZ, BRAB, BRABZ 1 1 0 1 0 1 1 z 0 0 1 % C6.2.033 BLRAA, BLRAAZ, BLRAB, BLRABZ 0 x 0 1 1 0 0 0 _ _ _ % C6.2.120 LDR (literal) 0 0 1 1 1 0 0 0 0 0 0 % C6.2.273 STTRB % C6.2.282 STURB 0 0 1 1 1 0 0 0 0 0 1 % C6.2.260 STRB (register) 0 0 1 1 1 0 0 0 0 1 0 % C6.2.144 LDTRB % C6.2.156 LDURB 0 0 1 1 1 0 0 0 0 1 1 % C6.2.124 LDRB (register) 0 0 1 1 1 0 0 0 1 0 1 % C6.2.092 LDAPRB 0 0 1 1 1 0 0 0 1 1 1 % C6.2.128 LDRSB (register) 0 0 1 1 1 0 0 0 1 x 0 % C6.2.146 LDTRSB % C6.2.158 LDURSB 0 0 1 1 1 0 0 0 a r 1 % C6.2.088 LDADDDB, LDADDAB, LDADDALB, LDADDLB % C6.2.233 STADDB, STADDLB % C6.2.107 LDCLRB, LDCLRAB, LDCLRALB, LDCLRLB % C6.2.236 STCLR, STCLRLB % C6.2.110 LDEORB, LDEORAB, LDEORALB, LDEORLB % C6.2.239 STEORB, STEORLB % C6.2.134 LDSETB, LDSETAB, LDSETALB, LDSETLB % C6.2.263 STSETB, STSETLB % C6.2.137 LDSMAXB, LDSMAXAB, LDSMAXALB, LDSMAXLB % C6.2.266 STSMAXB, ST
SMAXLB	% C6.2.140 LDSMINB, LDSMINAB, LDSMINALB, LDSMINLB % C6.2.269 STSMINB, ST
SMINLB	% C6.2.149 LDUMAXB, LDUMAXAB, LDUMAXALB, LDUMAXLB % C6.2.275 STUMAXB, ST
UMAXLB	% C6.2.152 LDUMINB, LDUMINAB, LDUMINALB, LDUMINLB % C6.2.278 STUMINB, ST
UMINLB	% C6.2.295 SWPB, SWPAB, SWPALB, SWPLB 0 0 1 1 1 0 0 1 0 0 _ % C6.2.259 STRB (immediate) 0 0 1 1 1 0 0 1 0 1 i % C6.2.123 LDRB (immediate) 0 0 1 1 1 0 0 1 1 1 i % C6.2.127 LDRSB (immediate) 0 1 1 1 1 0 0 0 0 0 0 % C6.2.274 STTRH % C6.2.283 STURH 0 1 1 1 1 0 0 0 0 0 1 % C6.2.262 STRH (register) 0 1 1 1 1 0 0 0 0 1 0 % C6.2.145 LDTRH % C6.2.157 LDURH 0 1 1 1 1 0 0 0 0 1 1 % C6.2.126 LDRH (register) 0 1 1 1 1 0 0 0 1 0 1 % C6.2.093 LDAPRH 0 1 1 1 1 0 0 0 1 x 0 % C6.2.147 LDTRSH % C6.2.159 LDURSH 0 1 1 1 1 0 0 0 1 x 1 % C6.2.130 LDRSH (register) 0 1 1 1 1 0 0 0 a r 1 % C6.2.089 LDADDH, LDADDAH, LDADDALH, LDADDLB % C6.2.234 STADDH, STADDLB % C6.2.108 LDCLRH, LDCLRAB, LDCLRALB, LDCLRLB % C6.2.237 STCLR, STCLRLB % C6.2.111 LDEORH, LDEORAH, LDEORALH, LDEORLB % C6.2.240 STEORH, STEORLB % C6.2.135 LDSETH, LDSETAB, LDSETALH, LDSETLB % C6.2.264 STSETH, STSETLB % C6.2.138 LDSMAXH, LDSMAXAB, LDSMAXALB, LDSMAXLB % C6.2.267 STSMAXH, ST
SMAXLH	% C6.2.141 LDSMINH, LDSMINAH, LDSMINALH, LDSMINLB % C6.2.270 STSMINH, ST
SMINLH	% C6.2.150 LDUMAXH, LDUMAXAB, LDUMAXALB, LDUMAXLB % C6.2.276 STUMAXH, ST
UMAXLH	

		% C6.2.153 LDUMINH, LDUMINAH, LDUMINALH, LDUMINLH % C6.2.279 STUMINH, ST
UMLNH		C6.2.296 SWPH, SWPAH, SWPALH, SWPLH
0 1 1 1 1 0 0 1 0 0 _		% C6.2.261 STRH (immediate)
0 1 1 1 1 0 0 i 0 1 i		C6.2.125 LDRH (immediate)
0 1 1 1 1 0 0 i 1 x i		C6.2.129 LDRSH (immediate)
0 1 1 1 1 0 0 i 0 0 _		% C6.2.261 STRH (immediate)
0 1 1 1 1 0 0 i 0 1 i		C6.2.125 LDRH (immediate)
0 1 1 1 1 0 0 i 1 x i		C6.2.129 LDRSH (immediate)
0 1 0 1 1 0 0 1 0 0 0		% C6.2.250 STLURH
0 1 0 1 1 0 0 1 0 1 0		% C6.2.096 LDAPURH
0 1 0 1 1 0 0 1 1 i 0		% C6.2.098 LDAPURSH
0 0 0 1 1 0 0 1 0 0 0		% C6.2.249 STLURB
0 0 0 1 1 0 0 1 0 1 0		% C6.2.095 LDAPURB
0 0 0 1 1 0 0 1 1 i 0		% C6.2.097 LDAPURSB
1 0 1 1 1 0 0 0 1 0 0		C6.2.148 LDTRSW % C6.2.160 LDURSW
1 0 1 1 1 0 0 i 1 0 i		% C6.2.131 LDRSW (immediate) % C6.2.133 LDRSW (register)
1 0 0 1 1 0 0 0 _ _ _		C6.2.132 LDRSW (literal)
1 0 0 1 1 0 0 1 1 0 0		C6.2.099 LDAPURSW
1 x 0 1 1 0 0 1 0 0 0		C6.2.248 STLUR
1 x 0 1 1 0 0 1 0 1 0		% C6.2.094 LDAPUR
1 x 1 1 1 0 0 0 0 0 0		% C6.2.272 STTR % C6.2.281 STUR
1 x 1 1 1 0 0 0 0 0 1		% C6.2.258 STR (register)
1 x 1 1 1 0 0 0 0 1 0		% C6.2.143 LDTR % C6.2.155 LDUR
1 x 1 1 1 0 0 0 0 1 1		% C6.2.121 LDR (register)
1 x 1 1 1 0 0 0 1 0 1		% C6.2.091 LDAPR
1 x 1 1 1 0 0 0 a r 1		% C6.2.090 LDADD, LDADDA, LDADDAL, LDADDL % C6.2.235 STADD, STADDL % C6.2.109 LDCLR, LDCLRA, LDCLRAL, LDCLRL % C6.2.238 STCLR, STCLRL % C6.2.112 LDEOR, LDEORA, LDEORAL, LDEORL % C6.2.241 STEOR, STEORL % C6.2.136 LDSET, LDSETA, LDSETAL, LDSETL % C6.2.265 STSET, STSETL % C6.2.139 LDSMAX, LDSMAXA, LDSMAXAL, LDSMAXL % C6.2.268 STSMAX, STSMAXL % C6.2.142 LDSMIN, LDSMINA, LDSMINAL, LDSMINL % C6.2.271 STSMIN, STSMINL % C6.2.151 LDUMAX, LDUMAXA, LDUMAXAL, LDUMAXL % C6.2.277 STUMAX, STUMAXL % C6.2.154 LDUMIN, LDUMINA, LDUMINAL, LDUMINL % C6.2.280 STUMIN, STUMINL % C6.2.297 SWP, SWPA, SWPAL, SWPL
1 x 1 1 1 0 0 1 0 0 _		% C6.2.257 STR (immediate)
1 x 1 1 1 0 0 1 0 1 i		% C6.2.119 LDR (immediate)
1 1 0 1 1 0 0 0 _ _ _		C6.2.201 PRFM (literal)
1 1 1 1 1 0 0 0 1 0 0		% C6.2.203 PRFUM
1 1 1 1 1 0 0 0 1 0 1		% C6.2.202 PRFM (register)
1 1 1 1 1 0 0 0 m s 1		% C6.2.122 LDRAA, LDRAB
1 1 1 1 1 0 0 0 m s 1		% C6.2.122 LDRAA, LDRAB
1 1 1 1 1 0 0 1 1 0 _		% C6.2.200 PRFM (immediate)
0 0 1 1 1 0 1 0 0 0 0		% C6.2.223 SETF8, SETF16
x 0 0 1 1 1 0 1 0 1 0 0		% C6.2.048 CINC % C6.2.66 CSINC % C6.2.64 CSET % C6.2.063 CSEL
x 0 0 1 1 1 0 1 0 1 1 0		% C6.2.015 ASR (register) % C6.2.17 ASRV % C6.2.060 CRC32B, CRC32H, CRC3
2W, CRC32X		% C6.2.061 CRC32CB, CRC32CH, CRC32CW, CRC32CX % C6.2.165 LSL (register) % C6.2.167 LSLV % C6.2.168 LSR (register) % C6
.2.170 LSRV		% C6.2.215 ROR (register) % C6.2.216 RORV % C6.2.222 SDIV % C6.2.313 UDI
V		
x 0 0 1 1 0 1 1 0 0 0		% C6.2.171 MADD % C6.2.185 MUL % C6.2.172 MNEG % C6.2.184 MSUB
x 0 1 1 1 0 1 0 0 0 0		% C6.2.002 ADCS
x 0 1 1 1 0 1 0 0 1 0		% C6.2.043 CCMN (immediate) % C6.2.044 CCMN (register)
x 0 1 1 1 0 1 0 0 0 0		% C6.2.002 ADCS

```

x 1 0 1 1 0 1 0 0 0 % C6.2.189 NGC % C6.2.217 SBC
x 1 0 1 1 0 1 0 1 0 % C6.2.049 CINV % C6.2.65 CSETM % C6.2.67 CSINV
x 1 0 1 1 0 1 0 1 0 % C6.2.059 CNEG % C6.2.68 CSNEG
x 1 0 1 1 0 1 0 1 1 % C6.2.051 CLS
x 1 0 1 1 0 1 0 1 1 % C6.2.052 CLZ
x 1 0 1 1 0 1 0 1 1 % C6.2.206 RBIT
x 1 0 1 1 0 1 0 1 1 % C6.2.209 REV % C6.2.210 REV16 % C6.2.212 REV64
x 1 0 1 1 0 1 0 0 0 % C6.2.189 NGC % C6.2.217 SBC
x 1 1 1 1 0 1 0 0 0 % C6.2.190 NGCS % C6.2.218 SBCS
x 1 1 1 1 0 1 0 0 1 % C6.2.045 CCMP (immediate) % C6.2.046 CCMP (register)
1 0 0 1 1 0 1 0 1 0 % C6.2.197 PACGA
1 0 0 1 1 0 1 1 0 0 1 % C6.2.226 SMADDL % C6.2.231 SMULL % C6.2.228 SMNEGL % C6.2.229 SMSUBL
1 0 0 1 1 0 1 1 0 1 0 % C6.2.230 SMULH
1 0 0 1 1 0 1 1 1 0 1 % C6.2.314 UMADDL % C6.2.318 UMULL % C6.2.315 UMNEGL % C6.2.316 UMSUBL
1 0 0 1 1 0 1 1 1 1 0 % C6.2.317 UMULH
1 0 1 1 1 0 1 0 0 0 0 % C6.2.213 RMIF
1 1 0 1 1 0 1 0 1 1 0 % C6.2.019 AUTDA, AUTDZA % C6.2.020 AUTDB, AUTDZB
% C6.2.021 AUTIA, AUTIA1716, AUTIASP, AUTIAZ, AUTIZA
% C6.2.022 AUTIB, AUTIB1716, AUTIBSP, AUTIBZ, AUTIZB
% C6.2.195 PACDA, PACDZA % C6.2.196 PACDB, PACDZB
% C6.2.211 REV32 % C6.2.323 XPACD, XPACI, XPAACLRI
1 1 0 1 1 0 1 0 1 1 1 % C6.2.199 PACIB, PACIB1716, PACIBSP, PACIBZ, PACIZB

```

SIMD NEON

0 x 0 s 1 1 1 0 s s 1 ... 0	1 0 1 1 1 0 % C7.2.001 ABS
0 x 0 s 1 1 1 0 s s 1 ... -	1 0 0 0 0 1 % C7.2.002 ADD (vector)
0 x 0 0 1 1 1 0 s s 1 ... -	0 1 0 0 0 0 % C7.2.003 ADDHN, ADDHN2
0 1 0 1 1 1 1 0 s s 1 ... 1	1 0 1 1 1 0 % C7.2.004 ADDP (scalar)
0 x 0 0 1 1 1 0 s s 1 ... -	1 0 1 1 1 1 % C7.2.005 ADDP (vector)
0 x 0 0 1 1 1 0 s s 1 ... 1	1 0 1 1 1 0 % C7.2.006 ADDV
0 1 0 0 1 1 1 0 0 0 1 ... 0	0 1 0 1 1 0 % C7.2.007 AESD - AES single round decryption.
0 1 0 0 1 1 1 0 0 0 1 ... 0	0 1 0 0 1 0 % C7.2.008 AESE - AES single round encryption.
0 1 0 0 1 1 1 0 0 0 1 ... 0	0 1 1 1 1 0 % C7.2.009 AESIMC - AES inverse mix columns.
0 1 0 0 1 1 1 0 0 0 1 ... 0	0 1 1 0 1 0 % C7.2.010 AESMC - AES mix columns.
0 x 0 0 1 1 1 0 0 0 1 ... -	0 0 0 1 1 1 % C7.2.011 AND (vector)
1 1 0 0 1 1 1 0 0 0 1 ... -	0 - - - - % C7.2.012 BCAX
0 x 1 0 1 1 1 1 0 0 0 ... c	x x x 1 0 1 % C7.2.013 BIC (vector, immediate)
0 x 0 0 1 1 1 0 0 1 1 ... -	0 0 0 1 1 1 % C7.2.014 BIC (vector, register)
 C7.2.119 FMLS (vector)	
0 x s 0 1 1 1 1 1 0 i ... -	1 1 0 0 h 0 % C7.2.120 FMLSL, FMLSL2 (by element)
0 x s 0 1 1 1 0 1 0 1 ... -	1 1 0 0 1 1 % C7.2.121 FMLSL, FMLSL2 (vector)
0 x x 0 1 1 1 1 0 0 0 ... -	1 1 1 1 0 1 % C7.2.122 FMOV (vector, immediate)
0 0 0 1 1 1 1 0 x x 1 ... 0	0 1 0 0 0 0 % C7.2.123 FMOV (register)
x 0 0 1 1 1 1 0 x x 1 ... -	0 0 0 0 0 0 % C7.2.124 FMOV (general)
0 0 0 1 1 1 1 0 x x 1 ... -	- - - 1 0 0 % C7.2.125 FMOV (scalar, immediate)
0 0 0 1 1 1 1 1 x x 1 ... -	1 - - - - % C7.2.126 FMSUB
0 x 1 0 1 1 1 0 1 x 1 ... 0	1 1 1 1 1 0 % C7.2.132 FNNEG (vector)
0 0 0 1 1 1 1 0 s s 1 ... 1	0 1 0 0 0 0 % C7.2.133 FNNEG (scalar)
0 0 0 1 1 1 1 1 s s 1 ... -	0 - - - - % C7.2.134 FNMADD
0 0 0 1 1 1 1 1 s s 1 ... -	1 - - - - % C7.2.135 FNMSUB
0 0 0 1 1 1 1 0 s s 1 ... -	1 0 0 0 1 0 % C7.2.136 FN_MUL (scalar)

0 1 0 1 1 1 1 0 1 x 1 ... 1	1 1 0 1 1 0 % C7.2.137 FRECPE
0 1 0 1 1 1 1 0 0 x 1 ... -	p p 1 1 1 1 % C7.2.138 FRECPS
0 1 0 1 1 1 1 0 1 x 1 ... 1	1 1 1 1 1 0 % C7.2.139 FRECPX
0 x 1 0 1 1 1 1 0 0 x 1 ... 1	1 0 0 0 1 0 % C7.2.140 FRINTA (vector)
0 0 0 1 1 1 1 0 s s 1 ... 0	0 1 0 0 0 0 % C7.2.141 FRINTA (scalar)
0 x 1 0 1 1 1 1 0 1 x 1 ... 1	1 0 0 1 1 0 % C7.2.142 FRINTI (vector)
0 0 0 1 1 1 1 0 s s 1 ... 1	1 1 0 0 0 0 % C7.2.143 FRINTI (scalar)
0 x 0 0 1 1 1 1 0 0 x 1 ... 1	1 0 0 1 1 0 % C7.2.144 FRINTM (vector)
0 0 0 1 1 1 1 0 s s 1 ... 1	0 1 0 0 0 0 % C7.2.145 FRINTM (scalar)
0 x 0 0 1 1 1 1 0 0 x 1 ... 1	1 0 0 0 1 0 % C7.2.146 FRINTN (vector)
0 0 0 1 1 1 1 0 s s 1 ... 0	0 1 0 0 0 0 % C7.2.147 FRINTN (scalar)
0 x 0 0 1 1 1 1 0 1 x 1 ... 1	1 0 0 0 1 0 % C7.2.148 FRINTP (vector)
0 0 0 1 1 1 1 0 s s 1 ... 0	1 1 0 0 0 0 % C7.2.149 FRINTP (scalar)
0 x 1 0 1 1 1 1 0 0 s 1 ... 1	1 0 0 1 1 0 % C7.2.150 FRINTX (vector)
0 0 0 1 1 1 1 0 s s 1 ... 1	0 1 0 0 0 0 % C7.2.151 FRINTX (scalar)
0 x 0 0 1 1 1 1 0 0 1 x ... 1	1 0 0 1 1 0 % C7.2.152 FRINTZ (vector)
0 0 0 1 1 1 1 0 s s 1 ... 1	1 1 0 0 0 0 % C7.2.153 FRINTZ (scalar)
0 1 1 1 1 1 1 0 1 x x ... 1	1 1 0 1 1 0 % C7.2.154 FRSQRTE
0 1 0 1 1 1 1 0 1 x x ... -	p p 1 1 1 1 % C7.2.155 FRSQRTS
0 x 1 0 1 1 1 0 1 1 1 ... 1	1 1 1 1 1 0 % C7.2.156 FSQRT (vector)
0 0 0 1 1 1 1 0 s s 1 ... 1	1 1 0 0 0 0 % C7.2.157 FSQRT (scalar)
0 x 0 0 1 1 1 1 0 1 x 1 ... -	p p 0 1 0 1 % C7.2.158 FSUB (vector)
0 0 0 1 1 1 1 0 s s 1 ... -	0 0 1 1 1 0 % C7.2.159 FSUB (scalar)
0 x 0 0 1 1 0 0 i 1 0 ... -	x x 1 x s s % C7.2.162 LD1 (multiple structures)
0 x 0 0 1 1 0 1 i 1 0 ... -	1 1 0 0 s s % C7.2.164 LD1R
0 x 0 0 1 1 0 1 i 1 0 ... -	x x 0 s s s % C7.2.163 LD1 (single structure)
0 x 0 0 1 1 0 0 i 1 0 ... -	1 0 0 0 s s % C7.2.165 LD2 (multiple structures)
0 x 0 0 1 1 0 1 i 1 1 ... -	1 1 0 0 s s % C7.2.167 LD2R
0 x 0 0 1 1 0 1 i 1 1 ... -	x x 0 s s s % C7.2.166 LD2 (single structure)
0 x 0 0 1 1 0 0 i 1 0 ... -	0 1 0 0 s s % C7.2.168 LD3 (multiple structures)
0 x 0 0 1 1 0 1 0 1 0 ... 0	1 1 1 0 s s % C7.2.170 LD3R
0 x 0 0 1 1 0 1 i 1 0 ... -	x x 1 s s s % C7.2.169 LD3 (single structure)
0 x 0 0 1 1 0 0 0 1 0 ... 0	0 0 0 0 s s % C7.2.171 LD4 (multiple structures)
0 x 0 0 1 1 0 1 i 1 1 ... -	1 1 1 0 s s % C7.2.173 LD4R
0 x 0 0 1 1 0 1 i 1 1 ... -	x x 1 s s s % C7.2.172 LD4 (single structure)
x x 1 0 1 1 0 0 0 1 ... -	- - - - - % C7.2.174 LDNP (SIMD and FP)
x x 1 0 1 1 0 i 1 1 ... -	- - - - - % C7.2.175 LDP (SIMD and FP)
x x 1 1 1 1 0 0 x 1 0 ... -	- - - 0 1 % C7.2.176 LDR (immediate, SIMD and FP)
x x 0 1 1 1 0 0 ... -	- - - - - % C7.2.177 LDR (literal, SIMD and FP)
x x 1 1 1 1 0 0 x 1 1 ... -	- - - 1 0 % C7.2.178 LDR (register, SIMD and FP)
x x 1 1 1 1 0 0 x 1 0 ... -	- - - 0 0 % C7.2.179 LDUR (SIMD and FP)
0 x 1 0 1 1 1 1 s s i ... -	0 0 0 0 h 0 % C7.2.180 MLA (by element)
0 x 0 0 1 1 1 0 s s 1 ... -	1 0 0 1 0 1 % C7.2.181 MLA (vector)
0 x 1 0 1 1 1 1 s s i ... -	0 1 0 0 h 0 % C7.2.182 MLS (by element)
0 x 1 0 1 1 1 0 s s 1 ... -	1 0 0 1 0 1 % C7.2.183 MLS (vector)
0 1 0 1 1 1 1 0 0 0 ... -	0 0 0 0 0 1 % C7.2.184 MOV (scalar) % C7.2.32 DUP (element)
0 1 1 0 1 1 1 0 0 0 ... -	0 - - - - 1 % C7.2.185 MOV (element) % C7.2.160 INS (element)
0 1 0 0 1 1 1 0 0 0 ... -	0 0 0 1 1 1 % C7.2.186 MOV (from general) % C7.2.161 INS (general)
)	
0 x 0 0 1 1 1 0 1 0 1 ... -	0 0 0 1 1 1 % C7.2.187 MOV (vector) % C7.2.198 ORR (vector, register)
0 x 0 0 1 1 1 0 0 0 ... 0	0 0 1 1 1 1 % C7.2.188 MOV (to general)
0 x s 0 1 1 1 1 0 0 0 ... -	- - - 0 1 % C7.2.189 MOVI
0 x 0 0 1 1 1 1 s s i ... -	1 0 0 0 h 0 % C7.2.190 MUL (by element)
0 x 0 0 1 1 1 0 s s 1 ... -	1 0 0 1 1 1 % C7.2.191 MUL (vector)
0 x 1 0 1 1 1 0 0 0 1 ... 0	0 1 0 0 1 1 % C7.2.192 MVN % C7.2.195 NOT

0 x 1 0 1 1 1 1 0 0 0 ... -	0 1 % C7.2.193 MVNI
0 x 1 0 1 1 1 0 s s 1 ... 0	1 0 1 1 1 1 % C7.2.194 NEG (vector)
0 x 0 0 1 1 1 0 1 1 1 ... -	0 0 0 1 1 1 % C7.2.196 ORN (vector)
0 x 0 0 1 1 1 1 0 0 0 ... -	- - - 1 0 1 % C7.2.197 ORR (vector, immediate)
0 x 1 0 1 1 1 0 s s 1 ... -	1 0 0 1 1 1 % C7.2.199 PMUL
0 x 0 0 1 1 1 0 s s 1 ... -	1 1 1 0 0 0 % C7.2.200 PMULL, PMULL2
0 x 1 0 1 1 1 0 s s 1 ... -	0 1 0 0 0 0 % C7.2.201 RADDHN, RADDHN2
1 1 0 0 1 1 1 0 0 1 1 ... -	1 0 0 0 1 1 % C7.2.202 RAX1
0 x 1 0 1 1 1 0 0 1 1 ... 0	0 1 0 1 1 0 % C7.2.203 RBIT (vector)
0 x 0 0 1 1 1 0 s s 1 ... 0	0 0 0 1 1 0 % C7.2.204 REV16 (vector)
0 x 1 0 1 1 1 0 s s 1 ... 0	0 0 0 0 1 0 % C7.2.205 REV32 (vector)
0 x 0 0 1 1 1 0 s s 1 ... 0	0 0 0 0 1 0 % C7.2.206 REV64
0 x 0 0 1 1 1 1 0 ... -	1 0 0 0 1 1 % C7.2.207 RSHRN, RSHRN2
0 x 1 0 1 1 1 0 s s 1 ... -	0 1 1 0 0 0 % C7.2.208 RSUBHN, RSUBHN2
0 x 0 0 1 1 1 0 s s 1 ... -	0 1 1 1 1 1 % C7.2.209 SABA
0 x 0 0 1 1 1 0 s s 1 ... -	0 1 0 1 0 0 % C7.2.210 SABAL, SABAL2
0 x 0 0 1 1 1 0 s s 1 ... -	0 1 1 1 0 1 % C7.2.211 SABD
0 x 0 0 1 1 1 0 s s 1 ... -	0 1 1 1 1 0 % C7.2.212 SABDL, SABDL2
0 x 0 0 1 1 1 0 s s 1 ... 0	0 1 1 0 1 0 % C7.2.213 SADALP
0 x 0 0 1 1 1 0 s s 1 ... -	0 0 0 0 0 0 % C7.2.214 SADDL, SADDL2
0 x 0 0 1 1 1 0 s s 1 ... 0	0 0 1 0 1 0 % C7.2.215 SADDLP
0 x 0 0 1 1 1 0 s s 1 ... 0	0 0 1 1 1 0 % C7.2.216 SADDLV
0 x 0 0 1 1 1 0 s s 1 ... -	0 0 0 1 0 0 % C7.2.217 SADDW, SADDW2
0 x 0 i 1 1 1 1 0 ... -	1 1 1 0 0 1 % C7.2.218 SCVTF (vector, fixed point)
0 x 0 i 1 1 1 0 0 i 1 ... 1	1 1 0 1 1 0 % C7.2.219 SCVTF (vector, integer)
x 0 0 1 1 1 1 0 s s 0 ... 0	- - - - % C7.2.220 SCVTF (scalar, fixed point)
x 0 0 1 1 1 1 0 s s 1 ... 0	0 0 0 0 0 0 % C7.2.221 SCVTF (scalar, integer)
0 x 0 0 1 1 1 1 s s i ... -	1 1 1 0 h 0 % C7.2.222 SDOT (by element)
0 x 0 0 1 1 1 0 s s 0 ... -	1 0 0 1 0 1 % C7.2.223 SDOT (vector)
0 1 0 1 1 1 1 0 0 0 0 ... -	0 0 0 0 0 0 % C7.2.224 SHA1C
0 1 0 1 1 1 1 0 0 0 1 ... 0	0 0 0 0 1 0 % C7.2.225 SHA1H
0 1 0 1 1 1 1 0 0 0 0 ... -	0 0 1 0 0 0 % C7.2.226 SHA1M
0 1 0 1 1 1 1 0 0 0 0 ... -	0 0 0 1 0 0 % C7.2.227 SHA1P
0 1 0 1 1 1 1 0 0 0 0 ... -	0 0 1 1 0 0 % C7.2.228 SHA1SU0
0 1 0 1 1 1 1 0 0 0 1 ... 0	0 0 0 1 1 0 % C7.2.229 SHA1SU1
0 1 0 1 1 1 1 0 0 0 0 ... -	0 1 0 1 0 0 % C7.2.230 SHA256H2
0 1 0 1 1 1 1 0 0 0 0 ... -	0 1 0 0 0 0 % C7.2.231 SHA256H
0 1 0 1 1 1 1 0 0 0 1 ... 0	0 0 1 0 1 0 % C7.2.232 SHA256SU0
0 1 0 1 1 1 1 0 0 0 0 ... -	0 1 1 0 0 0 % C7.2.233 SHA256SU1
1 1 0 0 1 1 1 1 0 0 1 1 ... -	1 0 0 0 0 0 % C7.2.234 SHA512H
1 1 0 0 1 1 1 1 0 0 1 1 ... -	1 0 0 0 0 1 % C7.2.235 SHA512H2
1 1 0 0 1 1 1 1 0 1 1 0 ... 0	1 0 0 0 0 0 % C7.2.236 SHA512SU0
1 1 0 0 1 1 1 1 0 0 1 1 ... -	1 0 0 0 1 0 % C7.2.237 SHA512SU1
0 x 0 0 1 1 1 0 s s 1 ... -	0 0 0 0 0 1 % C7.2.238 SHADD
0 x 0 i 1 1 1 1 0 ... -	0 1 0 1 0 1 % C7.2.239 SHL
0 x 1 0 1 1 1 0 s s 1 ... 1	0 0 1 1 1 0 % C7.2.240 SHLL, SHLL2
0 x 0 0 1 1 1 1 0 ... -	1 0 0 0 0 1 % C7.2.241 SHR, SHR2
0 x 0 0 1 1 1 0 s s 1 ... -	0 0 1 0 0 1 % C7.2.242 SHSUB
0 x 1 i 1 1 1 1 0 ... -	0 1 0 1 0 1 % C7.2.243 SLI
1 1 0 0 1 1 1 1 0 0 1 1 ... -	1 1 0 0 0 0 % C7.2.244 SM3PARTW1
1 1 0 0 1 1 1 1 0 0 1 1 ... -	1 1 0 0 0 1 % C7.2.245 SM3PARTW2
1 1 0 0 1 1 1 1 0 0 1 0 ... -	0 - - - - % C7.2.246 SM3SS1
1 1 0 0 1 1 1 1 0 0 1 0 ... -	1 0 1 1 0 0 % C7.2.247 SM3TT1A
1 1 0 0 1 1 1 1 0 0 1 0 ... -	1 0 1 1 0 1 % C7.2.248 SM3TT1B
1 1 0 0 1 1 1 1 0 0 1 0 ... -	1 0 1 1 1 0 % C7.2.249 SM3TT2A
1 1 0 0 1 1 1 1 0 0 1 0 ... -	1 0 1 1 1 1 % C7.2.250 SM3TT2B

1 1 0 0 1 1 1 0 1 1 0 ... 0	1 0 0 0 0 1 % C7.2.251 SM4E
1 1 0 0 1 1 1 0 0 1 1 ... -	1 1 0 0 1 0 % C7.2.252 SM4EKEY
0 x 0 0 1 1 1 0 s s 1 ... -	0 1 1 0 0 1 % C7.2.253 SMAX
0 x 0 0 1 1 1 0 s s 1 ... -	1 0 1 0 0 1 % C7.2.254 SMAXP
0 x 0 0 1 1 1 0 s s 1 ... 0	1 0 1 0 1 0 % C7.2.255 SMAXV
0 x 0 0 1 1 1 0 s s 1 ... -	0 1 1 0 1 1 % C7.2.256 SMIN
0 x 0 0 1 1 1 0 s s 1 ... -	1 0 1 0 1 1 % C7.2.257 SMINP
0 x 0 0 1 1 1 0 s s 1 ... 1	1 0 1 0 1 0 % C7.2.258 SMINV
0 x 0 0 1 1 1 1 s s 1 ... -	0 0 1 0 h 0 % C7.2.259 SMLAL, SMLAL2 (by element)
0 x 0 0 1 1 1 0 s s 1 ... -	1 0 0 0 0 0 % C7.2.260 SMLAL, SMLAL2 (vector)
0 x 0 0 1 1 1 1 s s 1 ... -	0 1 1 0 h 0 % C7.2.261 SMLSLSL, SMLSLSL2 (by element)
0 x 0 0 1 1 1 0 s s 1 ... -	1 0 1 0 0 0 % C7.2.262 SMLSLSL, SMLSLSL2 (vector)
0 x 0 0 1 1 1 0 0 0 0 ...	0 0 1 0 1 1 % C7.2.263 SMOV
0 x 0 0 1 1 1 1 s s i ... -	1 0 1 0 h 0 % C7.2.264 SMULL, SMULL2 (by element)
0 x 0 0 1 1 1 0 s s 1 ... -	1 1 0 0 0 0 % C7.2.265 SMULL, SMULL2 (vector)
0 x 0 i 1 1 1 0 s s 1 ... 0	0 1 1 1 1 0 % C7.2.266 SQABS
0 x 0 i 1 1 1 0 s s 1 ... -	0 0 0 0 1 1 % C7.2.267 SQADD
0 x 0 i 1 1 1 1 s s i ... -	0 0 1 1 h 0 % C7.2.268 SQDMLAL, SQDMLAL2 (by element)
0 x 0 i 1 1 1 0 s s 1 ... -	1 0 0 1 0 0 % C7.2.269 SQDMLAL, SQDMLAL2 (vector)
0 x 0 i 1 1 1 1 s s i ... -	0 1 1 1 h 0 % C7.2.270 SQDMLSL, SQDMLSL2 (by element)
0 x 0 i 1 1 1 0 s s 1 ... -	1 0 1 1 0 0 % C7.2.271 SQDMLSL, SQDMLSL2 (vector)
0 x 0 i 1 1 1 1 s s 1 ... -	1 1 0 0 h 0 % C7.2.272 SQDMULH (by element)
0 x 0 i 1 1 1 0 s s 1 ... -	1 0 1 1 0 1 % C7.2.273 SQDMULH (vector)
0 x 0 0 1 1 1 1 s s i ... -	1 0 1 1 h 0 % C7.2.274 SQDMULL, SQDMULL2 (by element)
0 x 0 i 1 1 1 0 s s 1 ... -	1 1 0 1 0 0 % C7.2.275 SQDMULL, SQDMULL2 (vector)
0 x 1 i 1 1 1 0 s s 1 ... -	0 1 1 1 1 0 % C7.2.276 SQNEG
0 x 1 i 1 1 1 1 s s i ... -	1 1 0 1 h 0 % C7.2.277 SQRDMLAH (by element)
0 x 1 i 1 1 1 0 s s 0 ... -	1 0 0 0 0 1 % C7.2.278 SQRDMLAH (vector)
0 x 1 i 1 1 1 1 s s i ... -	1 1 1 1 h 0 % C7.2.279 SQRDMLSH (by element)
0 x 1 i 1 1 1 0 s s 0 ... -	1 0 0 0 1 1 % C7.2.280 SQRDMLSH (vector)
0 x 0 i 1 1 1 1 s s i ... -	1 1 0 1 h 0 % C7.2.281 SQRDMULH (by element)
0 x 1 i 1 1 1 0 s s 1 ... -	1 0 1 1 0 1 % C7.2.282 SQRDMULH (vector)
0 x 0 i 1 1 1 0 s s 1 ... -	0 1 0 1 1 1 % C7.2.283 QRSHL
0 x 0 i 1 1 1 1 0 -	1 0 0 1 1 1 % C7.2.284 QRSHRN, QRSHRN2
0 x 1 i 1 1 1 1 0 -	1 0 0 0 1 1 % C7.2.285 QRSHRUN, QRSHRUN2
0 x 0 i 1 1 1 1 0 -	0 1 1 1 0 1 % C7.2.286 SQSHL (immediate)
0 x 0 i 1 1 1 0 s s 1 ... -	0 1 0 0 1 1 % C7.2.287 SQSHL (register)
0 x 1 0 1 1 1 1 0 -	0 1 1 0 0 1 % C7.2.288 SQSHLU
0 x 0 i 1 1 1 1 0 -	1 0 0 1 0 1 % C7.2.289 SQSHRN, SQSHRN2
0 x 1 i 1 1 1 1 0 -	1 0 0 0 0 1 % C7.2.290 SQSHRUN, SQSHRUN2
0 x 0 0 1 1 1 0 s s 1 ... -	0 0 1 0 1 1 % C7.2.291 SQSUB
0 x 0 0 1 1 1 0 s s 1 ... 1	0 1 0 0 1 0 % C7.2.292 SQXTN, SQXTN2
0 x 1 0 1 1 1 0 s s 1 ... 1	0 0 1 0 1 0 % C7.2.293 SQXTUN, SQXTUN2
0 x 0 0 1 1 1 0 s s 1 ... -	0 0 0 1 0 1 % C7.2.294 SRHADD
0 x 1 i 1 1 1 1 0 -	0 1 0 0 0 1 % C7.2.295 SRI
0 x 0 i 1 1 1 0 s s 1 ... -	0 1 0 1 0 1 % C7.2.296 SRSHL
0 x 0 i 1 1 1 1 0 -	0 0 1 0 0 1 % C7.2.297 SRSHR
0 x 0 i 1 1 1 1 0 -	0 0 1 1 0 1 % C7.2.298 SRSRA
0 x 0 i 1 1 1 0 s s 1 ... -	0 1 0 0 0 1 % C7.2.299 SSHL
0 x 0 i 1 1 1 1 0 -	0 0 0 0 0 1 % C7.2.301 SSHR
0 x 0 i 1 1 1 1 0 -	0 0 0 1 0 1 % C7.2.302 SSRA
0 x 0 0 1 1 1 0 s s 1 ... -	0 0 1 0 0 0 % C7.2.303 SSUBL, SSUBL2
0 x 0 0 1 1 1 0 s s 1 ... -	0 0 1 1 0 0 % C7.2.304 SSUBW, SSUBW2
0 x 0 0 1 1 0 0 i 0 0 ... -	x x 1 x s s % C7.2.305 ST1 (multiple structures)
0 x 0 0 1 1 0 1 i 0 0 ... -	x x 0 s s s % C7.2.306 ST1 (single structure)
0 x 0 0 1 1 0 0 i 0 0 ... -	1 0 0 0 s s % C7.2.307 ST2 (multiple structures)

0 x 0 0 1 1 0 1 i 0 1 ... -	x x 0 s s s % C7.2.308 ST2 (single structure)
0 x 0 0 1 1 0 0 i 0 0 ... -	0 1 0 0 s s % C7.2.309 ST3 (multiple structures)
0 x 0 0 1 1 0 1 i 0 0 ... -	x x 1 s s s % C7.2.310 ST3 (single structure)
0 x 0 0 1 1 0 0 x 0 0 ... -	0 0 0 0 s s % C7.2.311 ST4 (multiple structures)
0 x 0 0 1 1 0 1 0 0 1 ... 0	x x 1 s s s % C7.2.312 ST4 (single structure)
S s 1 0 1 1 0 0 0 0 ... -	- - - - C7.2.313 STNP (SIMD and FP)
S s 1 0 1 1 0 i i 0 ... -	- - - - % C7.2.314 STP (SIMDFP)
S s 1 1 1 1 0 0 x 0 0 ... -	- - - - i 1 C7.2.317 STUR (SIMD and FP)
S s 1 1 1 1 0 0 x 0 1 ... -	- - - - 1 0 C7.2.316 STR (register, SIMD and FP)
S s 1 1 1 1 0 0 x 0 0 ... -	- - - - 0 0 % C7.2.317 STUR (SIMD and FP)
0 x 1 0 1 1 1 0 s s 1 ... -	1 0 0 0 0 1 % C7.2.318 SUB (vector)
0 x 0 0 1 1 1 0 s s 1 ... -	0 1 1 0 0 0 % C7.2.319 SUBHN, SUBHN2
0 x 0 0 1 1 1 0 s s 1 ... 0	0 0 1 1 1 0 % C7.2.320 SUQADD
0 x 0 0 1 1 1 1 0 ... - 0	1 0 1 0 0 1 % C7.2.321 SXTL, SXTL2 % C7.2.300 SSHLL, SSHLL2
0 x 0 0 1 1 1 0 0 0 ... -	0 i 1 0 0 0 % C7.2.322 TBL
0 x 0 0 1 1 1 0 0 0 ... -	0 i 1 1 0 0 % C7.2.323 TBX
0 x 0 0 1 1 1 0 s s 0 ... -	0 0 1 0 1 0 % C7.2.324 TRN1
0 x 0 0 1 1 1 0 s s 0 ... -	0 1 1 0 1 0 % C7.2.325 TRN2
0 x 1 0 1 1 1 0 s s 1 ... -	0 1 1 1 1 1 % C7.2.326 UABA
0 x 1 0 1 1 1 0 s s 1 ... -	0 1 0 1 0 0 % C7.2.327 UABAL, UABAL2
0 x 1 0 1 1 1 0 s s 1 ... -	0 1 1 1 0 1 % C7.2.328 UABD
0 x 1 0 1 1 1 0 s s 1 ... -	0 1 1 1 1 0 % C7.2.329 UABDL, UABDL2
0 x 1 0 1 1 1 0 s s 1 ... 0	0 1 1 0 1 0 % C7.2.330 UADALP
0 x 1 0 1 1 1 0 s s 1 ... -	0 0 0 0 0 0 % C7.2.331 UADDL, UADDL2
0 x 1 0 1 1 1 0 s s 1 ... 0	0 0 1 1 0 1 % C7.2.332 UADDLP
0 x 1 0 1 1 1 0 s s 1 ... 0	0 0 1 1 1 0 % C7.2.333 UADDLV
0 x 1 0 1 1 1 1 0 ... - -	1 1 1 0 0 1 % C7.2.335 UCVTF (vector, fixed point)
0 x 1 1 1 1 1 0 0 s 1 ... 1	1 1 0 1 1 0 % C7.2.336 UCVTF (vector, integer)
x 0 0 1 1 1 1 0 s s 0 ... 1	- - - - % C7.2.337 UCVTF (scalar, fixed point)
x 0 0 1 1 1 1 0 s s 1 ... 1	0 0 0 0 0 0 % C7.2.338 UCVTF (scalar, integer)
0 x 1 0 1 1 1 1 s s i ... -	1 1 1 0 h 0 % C7.2.339 UDOT (by element)
0 x 1 0 1 1 1 1 0 s s 0 ... -	1 0 0 1 0 1 % C7.2.340 UDOT (vector)
0 x 1 0 1 1 1 0 s s 1 ... -	0 0 0 0 0 1 % C7.2.341 UHADD
0 x 1 0 1 1 1 0 s s 1 ... -	0 0 1 0 0 1 % C7.2.342 UHSUB
0 x 1 0 1 1 1 0 s s 1 ... -	0 1 1 0 0 1 % C7.2.343 UMAX
0 x 1 0 1 1 1 0 s s 1 ... -	1 0 1 0 0 1 % C7.2.344 UMAXP
0 x 1 0 1 1 1 0 s s 1 ... 0	1 0 1 0 1 0 % C7.2.345 UMAXV
0 x 1 0 1 1 1 0 s s 1 ... -	0 1 1 0 1 1 % C7.2.346 UMIN
0 x 1 0 1 1 1 0 s s 1 ... -	1 0 1 0 1 1 % C7.2.347 UMINP
0 x 1 0 1 1 1 0 s s 1 ... 1	1 0 1 0 1 0 % C7.2.348 UMINV
0 x 1 0 1 1 1 1 s s i ... -	0 0 1 0 h 0 % C7.2.349 UMLAL, UMLAL2 (by element)
0 x 1 0 1 1 1 0 s s 1 ... -	1 0 0 0 0 0 % C7.2.350 UMLAL, UMLAL2 (vector)
0 x 1 0 1 1 1 1 s s i ... -	0 1 1 0 h 0 % C7.2.351 UMLSL, UMLSL2 (by element)
0 x 1 0 1 1 1 0 s s 1 ... -	1 0 1 0 0 0 % C7.2.352 UMLSL, UMLSL2 (vector)
0 x 0 0 1 1 1 0 0 0 ... -	0 0 1 1 1 1 % C7.2.353 UMOV
0 x 1 0 1 1 1 1 s s i ... -	1 0 1 0 h 0 % C7.2.354 UMULL, UMULL2 (by element)
0 x 1 0 1 1 1 1 0 s s 1 ... -	1 1 0 0 0 0 % C7.2.355 UMULL, UMULL2 (vector)
0 x 1 0 1 1 1 0 s s 1 ... -	0 0 0 0 1 1 % C7.2.356 UQADD
0 x 1 0 1 1 1 0 s s 1 ... -	0 1 0 1 1 1 % C7.2.357 UQRSHL
0 x 1 0 1 1 1 1 0 ... - -	1 0 0 1 1 1 % C7.2.358 UQRSHRN, UQRSHRN2
0 x 1 i 1 1 1 1 0 ... - -	0 1 1 1 0 1 % C7.2.359 UQSHL (immediate)
0 x 1 i 1 1 1 0 s s 1 ... -	0 1 0 0 1 1 % C7.2.360 UQSHL (register)
0 x 1 0 1 1 1 1 0 ... - -	1 0 0 1 0 1 % C7.2.361 UQSHRN, UQSHRN2
0 x 1 i 1 1 1 0 s s 1 ... -	0 0 1 0 1 1 % C7.2.362 UQSUB - Unsigned saturating Subtract.
0 x 1 i 1 1 1 0 s s 1 ... 1	0 1 0 0 1 0 % C7.2.363 UQXTN, UQXTN2 - Unsigned saturating extract Narrow.

<code>0 x 0 0 1 1 1 0 1 s 1 ... 1</code>	<code>1 1 0 0 1 0</code>	<code>% C7.2.364 URECPE - Unsigned Reciprocal Estimate.</code>
<code>0 x 1 0 1 1 1 0 s s 1 ... _</code>	<code>0 0 0 1 0 1</code>	<code>% C7.2.365 URHADD Unsigned Rounding Halving Add.</code>
<code>0 x 1 i 1 1 1 0 s s 1 ... _</code>	<code>0 1 0 1 0 1</code>	<code>% C7.2.366 URSHL - Unsigned Rounding Shift Left (register).</code>
<code>0 1 1 1 1 1 1 1 0 _ _ ... _</code>	<code>0 0 1 0 0 1</code>	<code>% C7.2.367 URSHR - Unsigned Rounding Shift Right (immediate).</code>
<code>0 x 1 0 1 1 1 0 1 s 1 ... 1</code>	<code>1 1 0 0 1 0</code>	<code>% C7.2.368 URSQRTE - Unsigned Reciprocal Square Root Estimate.</code>
<code>0 1 1 1 1 1 1 1 0 _ _ ... _</code>	<code>0 0 1 1 0 1</code>	<code>% C7.2.369 URSRA</code>
<code>0 1 1 1 1 1 1 0 s s 1 ... _</code>	<code>0 1 0 0 0 1</code>	<code>% C7.2.370 USHL - Unsigned Shift Left (register).</code>
<code>0 x 1 0 1 1 1 1 0 _ _ ... 0</code>	<code>1 0 1 0 0 1</code>	<code>% C7.2.371 USHLL, USHLL2 % C7.2.377 UXTL, UXTL2</code>
<code>0 1 1 1 1 1 1 1 0 _ _ ... _</code>	<code>0 0 0 0 0 1</code>	<code>% C7.2.372 USHR - Unsigned Shift Right (immediate).</code>
<code>0 x 1 s 1 1 1 0 s s 1 ... 0</code>	<code>0 0 0 1 1 1</code>	<code>% C7.2.373 USQADD</code>
<code>0 1 1 1 1 1 1 1 0 _ _ ... _</code>	<code>0 0 0 1 0 1</code>	<code>% C7.2.374 USRA</code>
<code>0 x 1 0 1 1 1 0 s s 1 ... _</code>	<code>0 0 1 0 0 0</code>	<code>% C7.2.375 USUBL, USUBL2</code>
<code>0 x 1 0 1 1 1 0 s s 1 ... _</code>	<code>0 0 1 1 0 0</code>	<code>% C7.2.376 USUBW, USUBW2 - Unsigned Subtract Wide.</code>
<code>0 x 0 0 1 1 1 0 s s 0 ... _</code>	<code>0 0 0 1 1 0</code>	<code>% C7.2.378 UZP1 - Unzip vectors (primary).</code>
<code>0 x 0 0 1 1 1 0 s s 0 ... _</code>	<code>0 1 0 1 1 0</code>	<code>% C7.2.379 UZP2 - Unzip vectors (secondary).</code>
<code>1 1 0 0 1 1 1 0 1 0 0 ... _</code>	<code>- - - - -</code>	<code>% C7.2.380 XAR - Exclusive OR and Rotate.</code>
<code>0 x 0 0 1 1 1 0 s s 1 ... 1</code>	<code>0 0 1 0 1 0</code>	<code>% C7.2.381 XTN, XTN2 - Extract Narrow.</code>
<code>0 x 0 0 1 1 1 0 s s 0 ... _</code>	<code>0 0 1 1 1 0</code>	<code>% C7.2.382 ZIP1 - Zip vectors (primary).</code>
<code>0 x 0 0 1 1 1 0 s s 0 ... _</code>	<code>0 1 1 1 1 0</code>	<code>% C7.2.383 ZIP2 - Zip vectors (secondary).</code>

crifan.org, 使用[署名4.0国际\(CC BY 4.0\)协议](#)发布 all right reserved, powered by Gitbook最后更新：
2022-10-27 22:17:14

常用汇编指令

crifan.org, 使用[署名4.0国际\(CC BY 4.0\)协议](#)发布 all right reserved, powered by Gitbook最后更新：
2022-06-12 18:18:50

赋值

TODO:

下面多个帖子

MOV系列

- 【已解决】ARM汇编指令：FMOV
- 【已解决】ARM汇编指令：MOVK、MOVZ、MOVN

位操作

- 【已解决】ARM汇编指令：ubfx

涉及状态寄存器

- 【已解决】ARM汇编指令：MRS

crifan.org, 使用[署名4.0国际\(CC BY 4.0\)协议](#)发布 all right reserved, powered by Gitbook最后更新：
2022-10-27 21:57:28

内存操作

TODO:

下面多个帖子

- 【已解决】ARM汇编指令：LDR、STR
- 【已解决】ARM汇编指令：LDP、STP

Load

- 【已解决】ARM汇编指令：STUR
- 【基本解决】ARM汇编指令：stlxr

Store

- 【已解决】ARM汇编指令：LDUR

crifan.org, 使用[署名4.0国际\(CC BY 4.0\)协议](#)发布 all right reserved, powered by Gitbook最后更新：
2022-10-27 22:01:21

比较

TODO:

- 【已解决】ARM汇编指令：CMP、CMN
- 【已解决】ARM汇编指令：FCMP
- 【已解决】ARM汇编指令：TBZ和TBNZ

crifan.org, 使用[署名4.0国际\(CC BY 4.0\)协议](#)发布 all right reserved, powered by Gitbook最后更新：
2022-10-27 21:58:41

分支跳转

- 【已解决】ARM汇编指令：跳转指令 分支指令
 - 举例
 - 【整理】iOS逆向调试心得：Block的invoke函数调用
 - 常用 blr 跳转
- 【已解决】ARM汇编指令：br
- 【已解决】ARM汇编指令：bl
- 【已解决】ARM汇编指令：CBZ、CBNZ

crifan.org, 使用[署名4.0国际\(CC BY 4.0\)协议](#)发布 all right reserved, powered by Gitbook最后更新：
2022-06-15 08:47:10

条件选择

- 【已解决】ARM汇编指令：CSEL

crifan.org, 使用[署名4.0国际\(CC BY 4.0\)协议](#)发布 all right reserved, powered by Gitbook最后更新：
2022-06-12 18:21:47

寻址

- 【已解决】ARM汇编指令：ADR、ADRP

crifan.org, 使用[署名4.0国际\(CC BY 4.0\)协议](#)发布 all right reserved, powered by Gitbook最后更新：
2022-06-12 18:25:48

算数运算

TODO:

- 【已解决】ARM汇编指令：FMUL

crifan.org, 使用[署名4.0国际\(CC BY 4.0\)协议](#)发布 all right reserved, powered by Gitbook最后更新：
2022-10-27 21:59:55

逻辑运算

TODO:

- 【已解决】ARM汇编指令：AND
- 【已解决】ARM指令中ASR算术右移和LSR逻辑右移的区别

crifan.org, 使用[署名4.0国际\(CC BY 4.0\)协议](#)发布 all right reserved, powered by Gitbook最后更新：
2022-10-27 22:00:26

SVC系统调用

- SVC 0x80
 - 【已解决】32位ARM中SVC指令编码格式定义
 - 【记录】在线网站ARM汇编指令编码转换二进制值
 - 【未解决】IDA搜ARM汇编指令svc 0x80二进制值01 10 00 D4为何会搜出DCB 1
 - 【已解决】ARM汇编代码指令SVC 0x80转换成二进制编码值
 - 【已解决】ARM二进制汇编指令编码值01 10 00 D4和SVC指令编码
 - 【已解决】ARM64中为何SVC 0x80指令集编码二进制值是011000D4
 - 【未解决】IDA中如何查看汇编代码指令对应的二进制编码值
 -
 - 【整理】syscall内核系统调用和svc 0x80相关基础知识

crifan.org, 使用[署名4.0国际\(CC BY 4.0\)协议](#)发布 all right reserved, powered by Gitbook最后更新：
2022-06-15 09:03:02

其他

- 【整理】ARM中的DCB指令
- 【已解决】ARM汇编指令：FCVTZS
-

crifan.org, 使用[署名4.0国际\(CC BY 4.0\)协议](#)发布 all right reserved, powered by Gitbook最后更新：
2022-10-27 21:59:52

ARM常见用法

crifan.org, 使用[署名4.0国际\(CC BY 4.0\)协议](#)发布 all right reserved, powered by Gitbook最后更新：
2022-06-12 14:53:51

函数调用

- 【待整理】ARM64 函数调用 栈操作 栈帧
- 【未解决】ARM汇编指令：STP开辟栈空间即入栈和出栈相关

crifan.org, 使用[署名4.0国际\(CC BY 4.0\)协议](#)发布 all right reserved, powered by Gitbook最后更新：
2022-06-15 08:51:21

跳转指令

TODO:

【已解决】ARM汇编指令：TBZ和TBNZ

crifan.org, 使用[署名4.0国际\(CC BY 4.0\)协议](#)发布 all right reserved, powered by Gitbook最后更新：
2022-10-27 21:55:02

条件执行

- 【已解决】ARM汇编指令通用逻辑：条件执行Conditional execution
- 【基本解决】ARM指令中Carry的C的标志位的含义
- 【已解决】ARM指令AND如何影响Carry的C标志位

crifan.org, 使用[署名4.0国际\(CC BY 4.0\)协议](#)发布 all right reserved, powered by Gitbook最后更新：
2022-06-15 08:53:25

Pre-index和Post-index

- 【已解决】ARM指令中的LDP指令的Pre-index和Post-index
- 【已解决】ARM汇编指令STP后面的感叹号的含义

crifan.org, 使用[署名4.0国际\(CC BY 4.0\)协议](#)发布 all right reserved, powered by Gitbook最后更新：
2022-06-15 08:41:17

flexible second operand

- 【已解决】ARM汇编指令通用逻辑：Operand2灵活的第二个操作数flexible second operand

crifan.org, 使用[署名4.0国际\(CC BY 4.0\)协议](#)发布 all right reserved, powered by Gitbook最后更新：
2022-06-15 08:52:32

附录

下面列出相关参考资料。

crifan.org, 使用[署名4.0国际\(CC BY 4.0\)协议](#)发布 all right reserved, powered by Gitbook最后更新：
2022-03-17 20:39:28

X86汇编

与 ARM 相对应的 x86 架构

对应的也有各种内容，包括：

X86 调用规范 = 调用约定 = Calling Convention

- x86 Registers=X86寄存器
-
- Stack during Subroutine Call = 子函数调用时的堆栈

◦

crifan.org, 使用[署名4.0国际\(CC BY 4.0\)协议](#)发布 all right reserved, powered by Gitbook最后更新：
2022-06-12 18:17:07

参考资料

- [Guide to x86 Assembly \(virginia.edu\)](#)
- [How does the ARM architecture differ from x86? - Stack Overflow](#)
- [ARM \(groupoid.github.io\)](#)
-

crifan.org, 使用[署名4.0国际\(CC BY 4.0\)协议](#)发布 all right reserved, powered by Gitbook最后更新：
2022-10-27 22:09:40