目录

最流行汇编语言：ARM

- 最新版本：`v0.3`
- 更新时间：`20221020`

简介

整理最流行的汇编语言ARM的基础知识，寄存器架构，常用汇编指令等通用知识。

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

版权说明

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

ARM概述

- 【整理】ARM汇编指令和架构

ARMv8-A has a 64-bit architecture called AArch64

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

ARM寄存器

寄存器架构

TODO：

【整理】ARM汇编基础知识：寄存器命名叫法 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

- 。

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

IP

- 【已解决】ARM中的R12寄存器IP

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

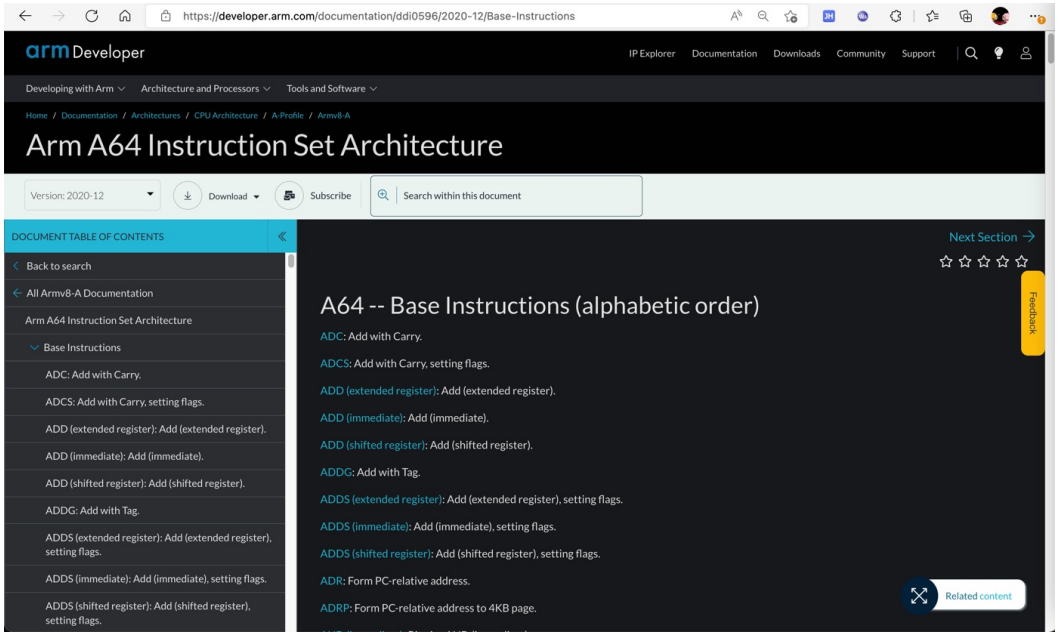
其他寄存器

- 【已解决】ARM汇编寄存器：TPIDRRO_EL0
- 【已解决】ARM寄存器：q0 q1

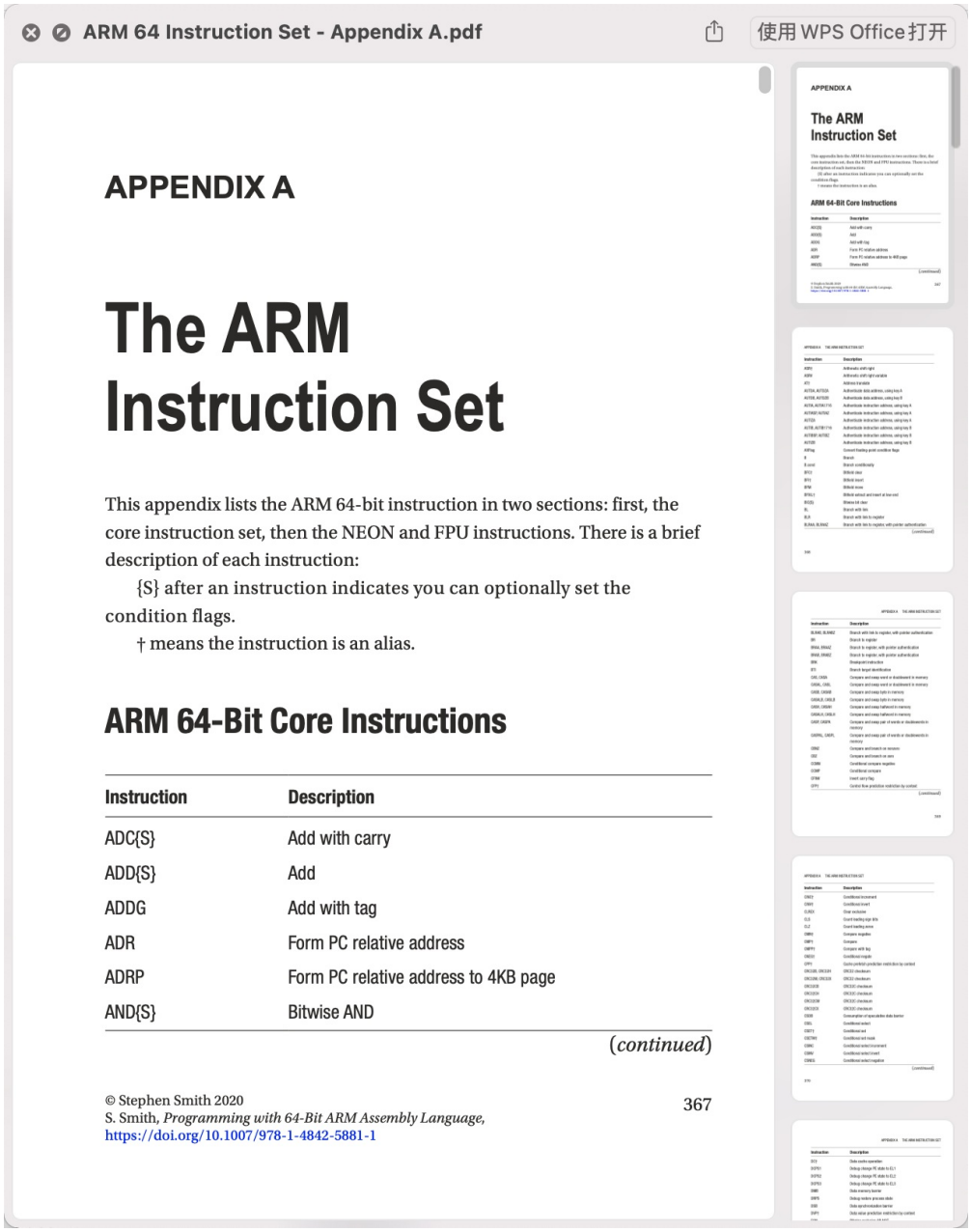
ARM汇编指令

汇编指令概述

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)
  - 官网的专业介绍
    - Arm A64 Instruction Set Architecture
    - 

  - 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



- 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 regis
ter).
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).
EOR (immediate): Bitwise Exclusive OR (immediate).
EOR (shifted register): Bitwise Exclusive OR (shifted register).
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.
LDAXR: 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.
RETAA, 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.
RMIF: 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.
SBCS: 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.
SMNEGL: 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, L
DADDA, LDADDAL, LDADDL.
STADDB, STADDLB: Atomic add on byte in memory, without return: an alias of LDADDB, LDADDAB, LDA
DDALB, 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 LD
CLR, LDCLRA, LDCLRAL, LDCLRL.
STCLRB, STCLRLB: Atomic bit clear on byte in memory, without return: an alias of LDCLRB, LDCLRA
B, LDCLRALB, LDCLRLB.
STCLRH, STCLRLH: Atomic bit clear on halfword in memory, without return: an alias of LDCLRH, LD
CLRAH, 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, LDE
ORAB, 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.
STSMAX, STSMAXL: Atomic signed maximum on word or doubleword in memory, without return: an alia
s of LDSMAX, LDSMAXA, LDSMAXAL, LDSMAXL.
STSMAXB, STSMAXLB: Atomic signed maximum on byte in memory, without return: an alias of LDSMAXB
, LDSMAXAB, LDSMAXALB, LDSMAXLB.
STSMAXH, STSMAXLH: Atomic signed maximum on halfword in memory, without return: an alias of LDS
MAXH, LDSMAXAH, LDSMAXALH, LDSMAXLH.
STSMIN, STSMINL: Atomic signed minimum on word or doubleword in memory, without return: an alia
s of LDSMIN, LDSMINA, LDSMINAL, LDSMINL.
STSMINB, STSMINLB: Atomic signed minimum on byte in memory, without return: an alias of LDSMINB
, LDSMINAB, LDSMINALB, LDSMINLB.
STSMINH, 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 al
ias of LDUMAX, LDUMAXA, LDUMAXAL, LDUMAXL.
STUMAXB, STUMAXLB: Atomic unsigned maximum on byte in memory, without return: an alias of LDUMA
XB, LDUMAXAB, LDUMAXALB, LDUMAXLB.
STUMAXH, STUMAXLH: Atomic unsigned maximum on halfword in memory, without return: an alias of L
DUMAXH, LDUMAXAH, LDUMAXALH, LDUMAXLH.
STUMIN, STUMINL: Atomic unsigned minimum on word or doubleword in memory, without return: an al
ias 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 L
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, XPACLRI: Strip Pointer Authentication Code.
YIELD: YIELD.
```

常用汇编指令

赋值

MOV系列

- 【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

内存操作

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

Load

Store

比较

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

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

条件选择

- 【已解决】ARM汇编指令：CSEL

寻址

- 【已解决】ARM汇编指令：ADR、ADRP

逻辑运算

- 【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

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

其他

- 【整理】ARM中的DCB指令

ARM常见用法

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

跳转指令

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

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

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

附录

下面列出相关参考资料。

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

。

参考资料

- Guide to x86 Assembly (virginia.edu)
- How does the ARM architecture differ from x86? - Stack Overflow
-