

ARM64 Cheatsheet

Генерални регистри

X0-X30: 64-битни регистри
W0-W30: Доњих 32 бита X регистра
XZR/WZR: Нулти регистар (увек враћа 0)
SP: Стек показивач
X30: Линк регистар (LR)

Приступ меморији

- **LDR X0, [X1]:** Учитај 64-битну вредност са адресе из X1 у регистар X0
- **STR W0, [X1]:** Сачувај 32-битну вредност W0 на адресу у X1
- **LDR(S)B/LDR(S)H:** Учитај 8/16-битну вредност виши битови се постављају на 0(знак)
- **STRB/STRH:** Учитај 8/16-битну вредност виши битови се постављају на 0(знак)
- **LDP/STP:** Учитај/сачувај пар регистара (најчешће са стеком)
- **Адресирање:**
 - **LDR X0, [X1, #8]:** Учитај са адресе $X1 + 8$
 - **LDR X0, [X1, X2, LSL 2]:** Учитај са адресе $X1 + X2 \ll 2$
 - **LDR X0, [SP], #8:** Учитај и повећај SP за 8
 - **LDR X0, [SP, #-8]!:** Учитај са адресе $SP - 8$ и смањи SP за 8

Аритметика и битовске операције

- **ADD X0, X1, X2:** $X0 = X1 + X2$
- **SUB X0, X1, X2:** $X0 = X1 - X2$
- **MUL X0, X1, X2:** $X0 = X1 * X2$
- **MUL X0, X1, X2:** $X0 = X1 * X2$
- **SDIV/UDIV:** Означено/неозначено дељење
- **MSUB X0, X1, X2, X3:** $X0 = X3 - (X1 * X2)$
- **AND, ORR, EOR:** Битовске операције
- **LSL, LSR, ASR, ROR:** Померања/ротације

Заставице (NZCV у PSTATE)

- **N:** Негативно
- **Z:** Нула
- **C:** Пренос (unsigned overflow)
- **V:** Прелив (signed overflow)

Постављају:

- **ADDS, SUBS - ADD, SUB + заставице**
- **CMP, TST**

Скокови и гранање

- | | | |
|---|----------------------------------|------------|
| • B label: Безусловни скок | (Z==0) | |
| • BL label: Позив функције | • B.LT, B.GT, B.LE, B.GE: | Означени |
| • RET: Повратак (у X30 (LR)) | скокови | |
| • B.EQ label: Скок ако је једнако (Z==1) | • B.LO, B.HI, B.LS, B.HS: | Неозначени |
| • B.NE label: Скок ако није једнако | скокови | |

Условно извршавање

- **CSEL X0, X1, X2, EQ:** $X0 = X1$ ако $Z==1$ иначе $X0 = X2$
- **CSINC X0, X1, X2, EQ:** $X0 = X1 + 1$ ако $Z==1$ иначе $X0 = X2$
- **CSINV X0, X1, XZR, EQ:** $X0 = \sim X1$ ако $Z==1$ иначе $X0 = 0$

Конвенција за позив функција

- **X0-X7**: Првих осам аргумената, остали аргументи се прослеђују преко стека
- **X0**: Резултат функциј
- **X8**: Системски позив (индиректни)
- **X0-X15**: Чувају се пре позива функције (caller-saved)
- **X16, X17**: Регистри које линкер користи за позив функција X16=IP0, X17=IP1
- **X18**: Платформски регистар (Platform Register), не користи се у корисничком коду
- **X19-X30**: Чувају се на почетку функције (callee-saved)
- **Стек**: 16-бајтно поравнање пре позива функције

Почетак функције:

STP X29, X30, [SP, #-16]!	STP FP, LR, [SP, #-16]!
MOV X29, SP	MOV FP, SP
SUB SP, SP, #<size>	SUB SP, SP, #<size>

Крај функције:

MOV SP, X29	MOV SP, FP
LDP X29, X30, [SP], #16	LDP FP, LR, [SP], #16
RET	RET

Компилација и дебаговање

Компилација:

```
aarch64-linux-gnu-g++ file1.c file2.s -g -o program
```

Покретање дебагера:

Емулација (QEMU терминал)

```
qemu-aarch64 -g 12345 ./program
```

unos podataka u program

Дебагер (GDB терминал)

```
gdb-multiarch ./program
set architecture aarch64
target remote localhost:12345
break ime_funkcije
continue
```

Инструкције дебагера:

- run: Покрени програм
- break: Обележава место за заустављање извршавања
 - break ime_funkcije: заустављање на почетку функције ime_funkcije
 - break 0x400123: заустављање на адреси 0x400123
 - break file.cpp:10: заустављање на линији 10 у фајлу file.cpp
- next: Изврши следећу инструкцију, не улази у функције
- step: Изврши следећу инструкцију, улази у функције
- continue: Настави извршавање до следеће тачке прекида
- print X0: Прикажи вредност регистра X0
- backtrace: Прикажи стек позива функција
- quit: Изађи из дебагера
- tui reg general: Прикажи регистре и код у TUI режиму

Већина инструкција може да се користи навођењем само првих неколико слова, на пример b уместо break, c уместо continue, итд.