

# ARM64 ELF hwcaps

This document describes the usage and semantics of the arm64 ELF hwcaps.

## 1. Introduction

Some hardware or software features are only available on some CPU implementations, and/or with certain kernel configurations, but have no architected discovery mechanism available to userspace code at EL0. The kernel exposes the presence of these features to userspace through a set of flags called hwcaps, exposed in the auxilliary vector.

Userspace software can test for features by acquiring the AT\_HWCAP or AT\_HWCAP2 entry of the auxiliary vector, and testing whether the relevant flags are set, e.g.:

```
bool floating_point_is_present(void)
{
    unsigned long hwcaps = getauxval(AT_HWCAP);
    if (hwcaps & HWCAP_FP)
        return true;

    return false;
}
```

Where software relies on a feature described by a hwcap, it should check the relevant hwcap flag to verify that the feature is present before attempting to make use of the feature.

Features cannot be probed reliably through other means. When a feature is not available, attempting to use it may result in unpredictable behaviour, and is not guaranteed to result in any reliable indication that the feature is unavailable, such as a SIGILL.

## 2. Interpretation of hwcaps

The majority of hwcaps are intended to indicate the presence of features which are described by architected ID registers inaccessible to userspace code at EL0. These hwcaps are defined in terms of ID register fields, and should be interpreted with reference to the definition of these fields in the ARM Architecture Reference Manual (ARM ARM).

Such hwcaps are described below in the form:

Functionality implied by `idreg.field == val`.

Such hwcaps indicate the availability of functionality that the ARM ARM defines as being present when `idreg.field` has value `val`, but do not indicate that `idreg.field` is precisely equal to `val`, nor do they indicate the absence of functionality implied by other values of `idreg.field`.

Other hwcaps may indicate the presence of features which cannot be described by ID registers alone. These may be described without reference to ID registers, and may refer to other documentation.

## 3. The hwcaps exposed in AT\_HWCAP

### HWCAP\_FP

Functionality implied by `ID_AA64PFR0_EL1.FP == 0b0000`.

### HWCAP\_ASIMD

Functionality implied by `ID_AA64PFR0_EL1.AdvSIMD == 0b0000`.

### HWCAP\_EVTSTRM

The generic timer is configured to generate events at a frequency of approximately 10KHz.

### HWCAP\_AES

Functionality implied by `ID_AA64ISAR0_EL1.AES == 0b0001`.

### HWCAP\_PMULL

Functionality implied by `ID_AA64ISAR0_EL1.AES == 0b0010`.

### HWCAP\_SHA1

Functionality implied by `ID_AA64ISAR0_EL1.SHA1 == 0b0001`.

### HWCAP\_SHA2

Functionality implied by `ID_AA64ISAR0_EL1.SHA2 == 0b0001`.

### HWCAP\_CRC32

Functionality implied by ID\_AA64ISAR0\_EL1.CRC32 == 0b0001.

#### HWCAP\_ATOMICS

Functionality implied by ID\_AA64ISAR0\_EL1.Atomic == 0b0010.

#### HWCAP\_FPHP

Functionality implied by ID\_AA64PFR0\_EL1.FP == 0b0001.

#### HWCAP\_ASIMDHP

Functionality implied by ID\_AA64PFR0\_EL1.AdvSIMD == 0b0001.

#### HWCAP\_CPUID

EL0 access to certain ID registers is available, to the extent described by Documentation/arm64/cpu-feature-registers.rst.

These ID registers may imply the availability of features.

#### HWCAP\_ASIMDRDM

Functionality implied by ID\_AA64ISAR0\_EL1.RDM == 0b0001.

#### HWCAP\_JSCVT

Functionality implied by ID\_AA64ISAR1\_EL1.JSCVT == 0b0001.

#### HWCAP\_FCMA

Functionality implied by ID\_AA64ISAR1\_EL1.FCMA == 0b0001.

#### HWCAP\_LRCPC

Functionality implied by ID\_AA64ISAR1\_EL1.LRCPC == 0b0001.

#### HWCAP\_DCPOP

Functionality implied by ID\_AA64ISAR1\_EL1.DPB == 0b0001.

#### HWCAP\_SHA3

Functionality implied by ID\_AA64ISAR0\_EL1.SHA3 == 0b0001.

#### HWCAP\_SM3

Functionality implied by ID\_AA64ISAR0\_EL1.SM3 == 0b0001.

#### HWCAP\_SM4

Functionality implied by ID\_AA64ISAR0\_EL1.SM4 == 0b0001.

#### HWCAP\_ASIMDDP

Functionality implied by ID\_AA64ISAR0\_EL1.DP == 0b0001.

#### HWCAP\_SHA512

Functionality implied by ID\_AA64ISAR0\_EL1.SHA2 == 0b0010.

#### HWCAP\_SVE

Functionality implied by ID\_AA64PFR0\_EL1.SVE == 0b0001.

#### HWCAP\_ASIMDFHM

Functionality implied by ID\_AA64ISAR0\_EL1.FHM == 0b0001.

#### HWCAP\_DIT

Functionality implied by ID\_AA64PFR0\_EL1.DIT == 0b0001.

#### HWCAP\_USCAT

Functionality implied by ID\_AA64MMFR2\_EL1.AT == 0b0001.

#### HWCAP\_ILRCPC

Functionality implied by ID\_AA64ISAR1\_EL1.LRCPC == 0b0010.

#### HWCAP\_FLAGM

Functionality implied by ID\_AA64ISAR0\_EL1.TS == 0b0001.

#### HWCAP\_SSBS

Functionality implied by ID\_AA64PFR1\_EL1.SSBS == 0b0010.

#### HWCAP\_SB

Functionality implied by ID\_AA64ISAR1\_EL1.SB == 0b0001.

#### HWCAP\_PACA

Functionality implied by ID\_AA64ISAR1\_EL1.APA == 0b0001 or ID\_AA64ISAR1\_EL1.API == 0b0001, as described by Documentation/arm64/pointer-authentication.rst.

#### HWCAP2\_PACG

Functionality implied by ID\_AA64ISAR1\_EL1.GPA == 0b0001 or ID\_AA64ISAR1\_EL1.GPI == 0b0001, as described by Documentation/arm64/pointer-authentication.rst.

#### HWCAP2\_DCPODP

Functionality implied by ID\_AA64ISAR1\_EL1.DPB == 0b0010.

#### HWCAP2\_SVE2

Functionality implied by ID\_AA64ZFR0\_EL1.SVEVer == 0b0001.

#### HWCAP2\_SVEAES

Functionality implied by ID\_AA64ZFR0\_EL1.AES == 0b0001.

#### HWCAP2\_SVEPMULL

Functionality implied by ID\_AA64ZFR0\_EL1.AES == 0b0010.

#### HWCAP2\_SVEBITPERM

Functionality implied by ID\_AA64ZFR0\_EL1.BitPerm == 0b0001.

#### HWCAP2\_SVSHA3

Functionality implied by ID\_AA64ZFR0\_EL1.SHA3 == 0b0001.

#### HWCAP2\_SVESM4

Functionality implied by ID\_AA64ZFR0\_EL1.SM4 == 0b0001.

#### HWCAP2\_FLAGM2

Functionality implied by ID\_AA64ISAR0\_EL1.TS == 0b0010.

#### HWCAP2\_FRINT

Functionality implied by ID\_AA64ISAR1\_EL1.FRINTTS == 0b0001.

#### HWCAP2\_SVEI8MM

Functionality implied by ID\_AA64ZFR0\_EL1.I8MM == 0b0001.

#### HWCAP2\_SVEF32MM

Functionality implied by ID\_AA64ZFR0\_EL1.F32MM == 0b0001.

#### HWCAP2\_SVEF64MM

Functionality implied by ID\_AA64ZFR0\_EL1.F64MM == 0b0001.

#### HWCAP2\_SVEBF16

Functionality implied by ID\_AA64ZFR0\_EL1.BF16 == 0b0001.

#### HWCAP2\_I8MM

Functionality implied by ID\_AA64ISAR1\_EL1.I8MM == 0b0001.

#### HWCAP2\_BF16

Functionality implied by ID\_AA64ISAR1\_EL1.BF16 == 0b0001.

#### HWCAP2\_DGH

Functionality implied by ID\_AA64ISAR1\_EL1.DGH == 0b0001.

#### HWCAP2\_RNG

Functionality implied by ID\_AA64ISAR0\_EL1.RNDR == 0b0001.

#### HWCAP2\_BTI

Functionality implied by ID\_AA64PFR0\_EL1.BT == 0b0001.

#### HWCAP2\_MTE

Functionality implied by ID\_AA64PFR1\_EL1.MTE == 0b0010, as described by Documentation/arm64/memory-tagging-extension.rst.

#### HWCAP2\_ECV

Functionality implied by ID\_AA64MMFR0\_EL1.ECV == 0b0001.

#### HWCAP2\_AFP

Functionality implied by ID\_AA64MFR1\_EL1.AFP == 0b0001.

#### HWCAP2\_RPRES

Functionality implied by ID\_AA64ISAR2\_EL1.RPRES == 0b0001.

#### HWCAP2\_MTE3

Functionality implied by ID\_AA64PFR1\_EL1.MTE == 0b0011, as described by Documentation/arm64/memory-tagging-extension.rst.

## 4. Unused AT\_HWCAP bits

For interoperation with userspace, the kernel guarantees that bits 62 and 63 of AT\_HWCAP will always be returned as 0.