

NAME

cpu_raw_data_t – Contains just the raw CPUID data.

SYNOPSIS

```
#include <libcpuid.h>
```

Data Fields

```
uint32_t basic_cpuid [MAX_CPUID_LEVEL][NUM_REGS]
uint32_t ext_cpuid [MAX_EXT_CPUID_LEVEL][NUM_REGS]
uint32_t intel_fn4 [MAX_INTELFN4_LEVEL][NUM_REGS]
uint32_t intel_fn11 [MAX_INTELFN11_LEVEL][NUM_REGS]
uint32_t intel_fn12h [MAX_INTELFN12H_LEVEL][NUM_REGS]
uint32_t intel_fn14h [MAX_INTELFN14H_LEVEL][NUM_REGS]
uint32_t amd_fn8000001dh [MAX_AMDFN8000001DH_LEVEL][NUM_REGS]
uint32_t amd_fn80000026h [MAX_AMDFN80000026H_LEVEL][NUM_REGS]
uint64_t arm_midr
uint64_t arm_mpidr
uint64_t arm_revidr
uint32_t arm_id_afr [MAX_ARM_ID_AFR_REGS]
uint32_t arm_id_dfr [MAX_ARM_ID_DFR_REGS]
uint32_t arm_id_isar [MAX_ARM_ID_ISAR_REGS]
uint32_t arm_id_mmfr [MAX_ARM_ID_MMFR_REGS]
uint32_t arm_id_pfr [MAX_ARM_ID_PFR_REGS]
uint64_t arm_id_aa64afr [MAX_ARM_ID_AA64AFR_REGS]
uint64_t arm_id_aa64dfr [MAX_ARM_ID_AA64DFR_REGS]
uint64_t arm_id_aa64fpfr [MAX_ARM_ID_AA64FPFR_REGS]
uint64_t arm_id_aa64isar [MAX_ARM_ID_AA64ISAR_REGS]
uint64_t arm_id_aa64mmfr [MAX_ARM_ID_AA64MMFR_REGS]
uint64_t arm_id_aa64pfr [MAX_ARM_ID_AA64PFR_REGS]
uint64_t arm_id_aa64smfr [MAX_ARM_ID_AA64SMFR_REGS]
uint64_t arm_id_aa64zfr [MAX_ARM_ID_AA64ZFR_REGS]
```

Detailed Description

Contains just the raw CPUID data.

This contains only the most basic CPU data, required to do identification and feature recognition. Every processor should be identifiable using this data only.

Field Documentation

```
uint32_t cpu_raw_data_t::amd_fn8000001dh[MAX_AMDFN8000001DH_LEVEL][NUM_REGS]
    when the CPU is AMD and supports leaf 8000001Dh (topology information for the DC) this stores the
    result of CPUID with eax = 8000001Dh and ecx = 0, 1, 2...

uint32_t cpu_raw_data_t::amd_fn80000026h[MAX_AMDFN80000026H_LEVEL][NUM_REGS]
    when the CPU is AMD and supports leaf 80000026h (Extended CPU Topology leaf) this stores the result of
    CPUID with eax = 80000026h and ecx = 0, 1, 2...

uint64_t cpu_raw_data_t::arm_id_aa64afr[MAX_ARM_ID_AA64AFR_REGS]
    when then CPU is ARM-based and supports ID_AA64AFR* (AArch64 Auxiliary Feature Register)

uint64_t cpu_raw_data_t::arm_id_aa64dfr[MAX_ARM_ID_AA64DFR_REGS]
    when then CPU is ARM-based and supports ID_AA64DFR* (AArch64 Debug Feature Register)

uint64_t cpu_raw_data_t::arm_id_aa64fpfr[MAX_ARM_ID_AA64FPFR_REGS]
    when then CPU is ARM-based and supports ID_AA64FPFR* (Floating-point Feature Register)

uint64_t cpu_raw_data_t::arm_id_aa64isar[MAX_ARM_ID_AA64ISAR_REGS]
    when then CPU is ARM-based and supports D_AA64ISAR* (AArch64 Instruction Set Attribute Register)
```

uint64_t cpu_raw_data_t::arm_id_aa64mmfr[MAX_ARM_ID_AA64MMFR_REGS]
 when then CPU is ARM-based and supports ID_AA64MMFR* (AArch64 Memory Model Feature Register)

uint64_t cpu_raw_data_t::arm_id_aa64pfr[MAX_ARM_ID_AA64PFR_REGS]
 when then CPU is ARM-based and supports ID_AA64PFR* (AArch64 Processor Feature Register)

uint64_t cpu_raw_data_t::arm_id_aa64smfr[MAX_ARM_ID_AA64SMFR_REGS]
 when then CPU is ARM-based and supports ID_AA64SMFR* (AArch64 SME Feature ID Register)

uint64_t cpu_raw_data_t::arm_id_aa64zfr[MAX_ARM_ID_AA64ZFR_REGS]
 when then CPU is ARM-based and supports ID_AA64ZFR* (SVE Feature ID register)

uint32_t cpu_raw_data_t::arm_id_afr[MAX_ARM_ID_AFR_REGS]
 when then CPU is ARM-based and supports ID_AFR* (AArch32 Auxiliary Feature Register)

uint32_t cpu_raw_data_t::arm_id_dfr[MAX_ARM_ID_DFR_REGS]
 when then CPU is ARM-based and supports ID_DFR* (AArch32 Debug Feature Register)

uint32_t cpu_raw_data_t::arm_id_isar[MAX_ARM_ID_ISAR_REGS]
 when then CPU is ARM-based and supports D_ISAR* (AArch32 Instruction Set Attribute Register)

uint32_t cpu_raw_data_t::arm_id_mmfr[MAX_ARM_ID_MMFR_REGS]
 when then CPU is ARM-based and supports ID_MMFR* (AArch32 Memory Model Feature Register)

uint32_t cpu_raw_data_t::arm_id_pfr[MAX_ARM_ID_PFR_REGS]
 when then CPU is ARM-based and supports ID_PFR* (AArch32 Processor Feature Register)

uint64_t cpu_raw_data_t::arm_midr
 when then CPU is ARM-based and supports MIDR (Main ID Register)

uint64_t cpu_raw_data_t::arm_mpidr
 when then CPU is ARM-based and supports MPIDR (Multiprocessor Affinity Register)

uint64_t cpu_raw_data_t::arm_revidr
 when then CPU is ARM-based and supports REVIDR (Revision ID Register)

uint32_t cpu_raw_data_t::basic_cpuid[MAX_CPUID_LEVEL][NUM_REGS]
 contains results of CPUID for eax = 0, 1, ...

uint32_t cpu_raw_data_t::ext_cpuid[MAX_EXT_CPUID_LEVEL][NUM_REGS]
 contains results of CPUID for eax = 0x80000000, 0x80000001, ...

uint32_t cpu_raw_data_t::intel_fn11[MAX_INTELFN11_LEVEL][NUM_REGS]
 when the CPU is intel and it supports leaf 0Bh (Extended Topology enumeration leaf), this stores the result of CPUID with eax = 11 and ecx = 0, 1, 2...

uint32_t cpu_raw_data_t::intel_fn12h[MAX_INTELFN12H_LEVEL][NUM_REGS]
 when the CPU is intel and supports leaf 12h (SGX enumeration leaf), this stores the result of CPUID with eax = 0x12 and ecx = 0, 1, 2...

uint32_t cpu_raw_data_t::intel_fn14h[MAX_INTELFN14H_LEVEL][NUM_REGS]
 when the CPU is intel and supports leaf 14h (Intel Processor Trace capabilities leaf). this stores the result of CPUID with eax = 0x12 and ecx = 0, 1, 2...

uint32_t cpu_raw_data_t::intel_fn4[MAX_INTELFN4_LEVEL][NUM_REGS]
 when the CPU is intel and it supports deterministic cache information: this contains the results of CPUID for eax = 4 and ecx = 0, 1, ...

Author

Generated automatically by Doxygen for libcuid from the source code.