# NAME

- PAPI_presets - PAPI predefined named events

## Synopsis

```
#include <papi.h>
```

## Description

The PAPI library names a number of predefined, or preset events. This set is a collection of events typically found in many CPUs that provide performance counters. A PAPI preset event name is mapped onto one or more of the countable native events on each hardware platform. On any particular platform, the preset can either be directly available as a single counter, derived using a combination of counters or unavailable.

The PAPI preset events can be broken loosely into several categories, as shown in the table below: PAPI Preset Event Definitions by Category:

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PAPI_BR_CN</td>
<td>Conditional branch instructions</td>
</tr>
</tbody>
</table>

## Contents

- 1 NAME
- 2 Synopsis
- 3 Description
  - 3.1 Conditional Branching
  - 3.2 Cache Requests:
  - 3.3 Conditional Store:
  - 3.4 Floating Point Operations:
  - 3.5 Instruction Counting:
  - 3.6 Cache Access:
  - 3.7 Data Access:
  - 3.8 TLB Operations:
- 4 Bugs
- 5 AUTHORS
- 6 See Also
PAPI_BR_INS  Branch instructions
PAPI_BR_MSP  Conditional branch instructions mispredicted
PAPI_BR_NTK  Conditional branch instructions not taken
PAPI_BR_PRC  Conditional branch instructions correctly predicted
PAPI_BR_TKN  Conditional branch instructions taken
PAPI_BR_UCN  Unconditional branch instructions
PAPI_BRU_IDL  Cycles branch units are idle
PAPI_BTAC_M  Branch target address cache misses

Cache Requests:

PAPI_CA_CLN  Requests for exclusive access to clean cache line
PAPI_CA_INV  Requests for cache line invalidation
PAPI_CA_ITV  Requests for cache line intervention
PAPI_CA_SHR  Requests for exclusive access to shared cache line
PAPI_CA_SNP  Requests for a snoop

Conditional Store:

PAPI_CSR_FAL  Failed store conditional instructions
PAPI_CSR_SUC  Successful store conditional instructions
PAPI_CSR_TOT  Total store conditional instructions

Floating Point Operations:

PAPI_FAD_INS  Floating point add instructions
PAPI_FDV_INS  Floating point divide instructions
PAPI_FMA_INS  FMA instructions completed
PAPI_FML_INS  Floating point multiply instructions
PAPI_FNV_INS  Floating point inverse instructions
PAPI_FP_INS  Floating point instructions
PAPI_FP_OPS  Floating point operations
PAPI_FP_STAL  Cycles the FP unit
PAPI_FPU_IDL  Cycles floating point units are idle
PAPI FSQ_INS  Floating point square root instructions

Instruction Counting:

PAPI_FUL_CCY  Cycles with maximum instructions completed
PAPI_FUL_ICY  Cycles with maximum instruction issue
PAPI_FXU_IDL  Cycles integer units are idle
PAPI_HW_INT  Hardware interrupts
PAPI_INT_INS  Integer instructions
PAPI_TOT_CYC  Total cycles
PAPI_TOT_IIS  Instructions issued
PAPI_TOT_INS  Instructions completed
PAPI_VEC_INS  Vector/SIMD instructions
**Cache Access:**

- \texttt{PAPI\_L1\_DCA} L1 data cache accesses
- \texttt{PAPI\_L1\_DCH} L1 data cache hits
- \texttt{PAPI\_L1\_DCM} L1 data cache misses
- \texttt{PAPI\_L1\_DCR} L1 data cache reads
- \texttt{PAPI\_L1\_DCW} L1 data cache writes
- \texttt{PAPI\_L1\_ICA} L1 instruction cache accesses
- \texttt{PAPI\_L1\_ICH} L1 instruction cache hits
- \texttt{PAPI\_L1\_ICM} L1 instruction cache misses
- \texttt{PAPI\_L1\_ICR} L1 instruction cache reads
- \texttt{PAPI\_L1\_ICW} L1 instruction cache writes
- \texttt{PAPI\_L1\_LDM} L1 load misses
- \texttt{PAPI\_L1\_STM} L1 store misses
- \texttt{PAPI\_L1\_TCA} L1 total cache accesses
- \texttt{PAPI\_L1\_TCH} L1 total cache hits
- \texttt{PAPI\_L1\_TCM} L1 total cache misses
- \texttt{PAPI\_L1\_TCR} L1 total cache reads
- \texttt{PAPI\_L1\_TCW} L1 total cache writes
- \texttt{PAPI\_L2\_DCA} L2 data cache accesses
- \texttt{PAPI\_L2\_DCH} L2 data cache hits
- \texttt{PAPI\_L2\_DCM} L2 data cache misses
- \texttt{PAPI\_L2\_DCR} L2 data cache reads
- \texttt{PAPI\_L2\_DCW} L2 data cache writes
- \texttt{PAPI\_L2\_ICA} L2 instruction cache accesses
- \texttt{PAPI\_L2\_ICH} L2 instruction cache hits
- \texttt{PAPI\_L2\_ICM} L2 instruction cache misses
- \texttt{PAPI\_L2\_ICR} L2 instruction cache reads
- \texttt{PAPI\_L2\_ICW} L2 instruction cache writes
- \texttt{PAPI\_L2\_LDM} L2 load misses
- \texttt{PAPI\_L2\_STM} L2 store misses
- \texttt{PAPI\_L2\_TCA} L2 total cache accesses
- \texttt{PAPI\_L2\_TCH} L2 total cache hits
- \texttt{PAPI\_L2\_TCM} L2 total cache misses
- \texttt{PAPI\_L2\_TCR} L2 total cache reads
- \texttt{PAPI\_L2\_TCW} L2 total cache writes
- \texttt{PAPI\_L3\_DCA} L3 data cache accesses
- \texttt{PAPI\_L3\_DCH} L3 Data Cache Hits
- \texttt{PAPI\_L3\_DCM} L3 data cache misses
- \texttt{PAPI\_L3\_DCR} L3 data cache reads
- \texttt{PAPI\_L3\_DCW} L3 data cache writes
- \texttt{PAPI\_L3\_ICA} L3 instruction cache accesses
- \texttt{PAPI\_L3\_ICH} L3 instruction cache hits
- \texttt{PAPI\_L3\_ICM} L3 instruction cache misses
- \texttt{PAPI\_L3\_ICR} L3 instruction cache reads
- \texttt{PAPI\_L3\_ICW} L3 instruction cache writes
- \texttt{PAPI\_L3\_LDM} L3 load misses
PAPI_L3_STM  L3 store misses
PAPI_L3_TCA  L3 total cache accesses
PAPI_L3_TCH  L3 total cache hits
PAPI_L3_TCM  L3 cache misses
PAPI_L3_TCR  L3 total cache reads
PAPI_L3_TCW  L3 total cache writes

Data Access:

PAPI_LD_INS   Load instructions
PAPI_LST_INS  Load/store instructions completed
PAPI_LSU_IDL  Cycles load/store units are idle
PAPI_MEM_RCY  Cycles Stalled Waiting for memory Reads
PAPI_MEM_SCY  Cycles Stalled Waiting for memory accesses
PAPI_MEM_WCY  Cycles Stalled Waiting for memory writes
PAPI_PRF_DM   Data prefetch cache misses
PAPI_RES_STL  Cycles stalled on any resource
PAPI_SR_INS   Store instructions
PAPI_STL_CCY  Cycles with no instructions completed
PAPI_STL_ICY  Cycles with no instruction issue
PAPI_SYC_INS  Synchronization instructions completed

TLB Operations:

PAPI_TLB_DM   Data translation lookaside buffer misses
PAPI_TLB_IM   Instruction translation lookaside buffer misses
PAPI_TLB_SD   Translation lookaside buffer shootdowns
PAPI_TLB_TL   Total translation lookaside buffer misses

**Bugs**

The exact semantics of an event counter are platform dependent. PAPI preset names are mapped onto available events in a way so as to count as similar types of events as possible on different platforms. Due to hardware implementation differences it is not necessarily possible to directly compare the counts of a particular PAPI event obtained on different hardware platforms.

**AUTHORS**

Nils Smeds <smeds@cs.utk.edu>

**See Also**

PAPI_event_code_to_name(3), PAPI_event_name_to_code(3), PAPI(3), PAPI_native(3), PAPI_enum_event(3), PAPI_get_event_info(3)


Category: PAPI Component Manual
This page was last modified on 17 October 2008, at 00:51.
This page has been accessed 308 times.

Jan 26, 2010