

```

/* SPDX-License-Identifier: GPL-2.0 */
#ifndef ALPHA_THREAD_INFO_H
#define ALPHA_THREAD_INFO_H

#ifdef KERNEL

#ifndef ASSEMBLY
#include <asm/processor.h>
#include <asm/types.h>
#include <asm/hwrpb.h>
#include <asm/sysinfo.h>
#endif

#ifndef ASSEMBLY
struct thread_info {
    struct pcb_struct    pcb;           /* palcode state */

    struct task_struct  *task;         /* main task structure */
    unsigned int         flags;         /* low level flags */
    unsigned int         ieee_state;    /* see fpu.h */

    mm_segment_t        addr_limit;    /* thread address space */
    unsigned             cpu;          /* current CPU */
    int                 preempt_count; /* 0 => preemptable, <0 =>
BUG */
    unsigned int         status;        /* thread-synchronous
flags */

    int bpt_nsaved;
    unsigned long bpt_addr[2];         /* breakpoint handling */
    unsigned int bpt_insn[2];
};

/*
 * Macros/functions for gaining access to the thread information
 * structure.
 */
#define INIT_THREAD_INFO(tsk)
{
    .task           = &tsk,
    .addr_limit      = KERNEL_DS,
    .preempt_count   = INIT_PREEMPT_COUNT,
}

/* How to get the thread information struct from C. */
register struct thread_info *__current_thread_info __asm ("8");
#define current_thread_info() __current_thread_info

#endif /* ASSEMBLY */

/* Thread information allocation. */
#define THREAD_SIZE_ORDER 1
#define THREAD_SIZE (2*PAGE_SIZE)

/*
 * Thread information flags:
 * - these are process state flags and used from assembly

```

```

* - pending work-to-be-done flags come first and must be assigned to
be
*   within bits 0 to 7 to fit in and immediate operand.
*
* TIF_SYSCALL_TRACE is known to be 0 via blbs.
*/
#define TIF_SYSCALL_TRACE 0      /* syscall trace active */
#define TIF_NOTIFY_RESUME 1    /* callback before returning to user
*/
#define TIF_SIGPENDING 2      /* signal pending */
#define TIF_NEED_RESCHED 3    /* rescheduling necessary */
#define TIF_SYSCALL_AUDIT 4    /* syscall audit active */
#define TIF_NOTIFY_SIGNAL 5    /* signal notifications exist */
#define TIF_DIE_IF_KERNEL 9    /* dik recursion lock */
#define TIF_MEMDIE 13         /* is terminating due to OOM killer
*/
#define TIF_POLLING_NRFLAG 14  /* idle is polling for
TIF_NEED_RESCHED */

#define TIF_SYSCALL_TRACE (1<<TIF_SYSCALL_TRACE)
#define TIF_SIGPENDING (1<<TIF_SIGPENDING)
#define TIF_NEED_RESCHED (1<<TIF_NEED_RESCHED)
#define TIF_NOTIFY_RESUME (1<<TIF_NOTIFY_RESUME)
#define TIF_SYSCALL_AUDIT (1<<TIF_SYSCALL_AUDIT)
#define TIF_NOTIFY_SIGNAL (1<<TIF_NOTIFY_SIGNAL)
#define TIF_POLLING_NRFLAG (1<<TIF_POLLING_NRFLAG)

/* Work to do on interrupt/exception return. */
#define TIF_WORK_MASK      (TIF_SIGPENDING | TIF_NEED_RESCHED
| \
TIF_NOTIFY_RESUME)

```