```
#ifndef _AHA1740_H

/* $Id$
 *
 * Header file for the adaptec 1740 driver for Linux
 *
 * With minor revisions 3/31/93
 * Written and (C) 1992,1993 Brad McLean.  See aha1740.c
 * for more info
 *
 */

#include <linux/types.h>
#include <linux/kdev_t.h>

/*
 * Eisa Enhanced mode operation - slot locating and addressing
 *
 * I don't have an EISA Spec to know these ranges, so I Just took my
 * machine's specifications.  Adjust to fit.
 */
#define MINEISA 1
#define MAXEISA 8   /* I just saw an ad, and bumped this from 6 to 8 */

#define SLOTBASE(x)     ((x << 12)+ 0xc80 )
#define BASE            (base)

/* EISA configuration registers & values */
#define HID0    (base + 0x0)
#define HID1    (base + 0x1)
#define HID2    (base + 0x2)
#define HID3    (base + 0x3)
#define EBCNTRL (base + 0x4)
#define PORTADR (base + 0x40)
#define BIOSADR (base + 0x41)
#define INTDEF  (base + 0x42)
#define SCSIDEF (base + 0x43)
#define BUSDEF  (base + 0x44)
#define RESV0   (base + 0x45)
#define RESV1   (base + 0x46)
#define RESV2   (base + 0x47)

#define HID_MFG "ADP"
#define HID_PRD 0
#define HID_REV 2
#define EBCNTRL_VALUE 1
#define PORTADDR_ENH 0x80
/* READ */
#define G2INTST (BASE + 0x56)
#define G2STAT  (BASE + 0x57)
#define MBOXIN0 (BASE + 0x58)
#define MBOXIN1 (BASE + 0x59)
#define MBOXIN2 (BASE + 0x5a)
#define MBOXIN3 (BASE + 0x5b)
#define G2STAT2 (BASE + 0x5c)

#define G2INTST_MASK            0xf0    /* isolate the status */
#define G2INTST_CCBGOOD         0x10    /* CCB Completed */
#define G2INTST_CCBRETRY        0x50    /* CCB Completed with a retry */
#define G2INTST_HARDFAIL        0x70    /* Adapter Hardware Failure */
```

```
#define G2INTST_CMDGOOD         0xa0    /* Immediate command success */
#define G2INTST_CCBERROR        0xc0    /* CCB Completed with error */
#define G2INTST_ASNEVENT        0xd0    /* Asynchronous Event Notification */
#define G2INTST_CMDERROR        0xe0    /* Immediate command error */

#define G2STAT_MBXOUT   4       /* Mailbox Out Empty Bit */
#define G2STAT_INTPEND  2       /* Interrupt Pending Bit */
#define G2STAT_BUSY     1       /* Busy Bit (attention pending) */

#define G2STAT2_READY   0       /* Host Ready Bit */

/* WRITE (and ReadBack) */
#define MBOXOUT0        (BASE + 0x50)
#define MBOXOUT1        (BASE + 0x51)
#define MBOXOUT2        (BASE + 0x52)
#define MBOXOUT3        (BASE + 0x53)
#define ATTN            (BASE + 0x54)
#define G2CNTRL         (BASE + 0x55)

#define ATTN_IMMED      0x10    /* Immediate Command */
#define ATTN_START      0x40    /* Start CCB */
#define ATTN_ABORT      0x50    /* Abort CCB */

#define G2CNTRL_HRST    0x80            /* Hard Reset */
#define G2CNTRL_IRST    0x40            /* Clear EISA Interrupt */
#define G2CNTRL_HRDY    0x20            /* Sets HOST ready */

/* This is used with scatter-gather */
struct aha1740_chain {
  u32  dataptr;         /* Location of data */
  u32  datalen;         /* Size of this part of chain */
};

/* These belong in scsi.h */
#define any2scsi(up, p)                         \
(up)[0] = (((unsigned long)(p)) >> 16)  ;       \
(up)[1] = (((unsigned long)(p)) >> 8);          \
(up)[2] = ((unsigned long)(p));

#define scsi2int(up) ( (((long)*(up)) << 16) + \
                      (((long)(up)[1]) << 8) + ((long)(up)[2]) )

#define xany2scsi(up, p)        \
(up)[0] = ((long)(p)) >> 24;    \
(up)[1] = ((long)(p)) >> 16;    \
(up)[2] = ((long)(p)) >> 8;     \
(up)[3] = ((long)(p));

#define xscsi2int(up) ( (((long)(up)[0]) << 24) + (((long)(up)[1]) << 16) \
                      + (((long)(up)[2]) <<  8) +  ((long)(up)[3]) )

#define MAX_CDB 12
#define MAX_SENSE 14
#define MAX_STATUS 32

struct ecb {                    /* Enhanced Control Block 6.1 */
  u16   cmdw;                   /* Command Word */
  /* Flag Word 1 */
  u16           cne:1,          /* Control Block Chaining */
        :6,     di:1,           /* Disable Interrupt */
```

```
        :2,     ses:1,          /* Suppress Underrun error */
        :1,     sg:1,           /* Scatter/Gather */
        :1,     dsb:1,          /* Disable Status Block */
                ars:1;          /* Automatic Request Sense */
  /* Flag Word 2 */
  u16           lun:3,          /* Logical Unit */
                tag:1,          /* Tagged Queuing */
                tt:2,           /* Tag Type */
                nd:1,           /* No Disconnect */
        :1,     dat:1,          /* Data transfer - check direction */
                dir:1,          /* Direction of transfer 1 = datain */
                st:1,           /* Suppress Transfer */
                chk:1,          /* Calculate Checksum */
        :2,     rec:1,  :1;     /* Error Recovery */
  u16   nil0;                   /* nothing */
  u32   dataptr;                /* Data or Scatter List ptr */
  u32   datalen;                /* Data or Scatter List len */
  u32   statusptr;              /* Status Block ptr */
  u32   linkptr;                /* Chain Address */
  u32   nil1;                   /* nothing */
  u32   senseptr;               /* Sense Info Pointer */
  u8    senselen;               /* Sense Length */
  u8    cdblen;                 /* CDB Length */
  u16   datacheck;              /* Data checksum */
  u8    cdb[MAX_CDB];           /* CDB area */
/* Hardware defined portion ends here, rest is driver defined */
  u8    sense[MAX_SENSE];       /* Sense area */
  u8    status[MAX_STATUS];     /* Status area */
  Scsi_Cmnd *SCpnt;             /* Link to the SCSI Command Block */
  void (*done)(Scsi_Cmnd *);    /* Completion Function */
};

#define AHA1740CMD_NOP   0x00   /* No OP */
#define AHA1740CMD_INIT  0x01   /* Initiator SCSI Command */
#define AHA1740CMD_DIAG  0x05   /* Run Diagnostic Command */
#define AHA1740CMD_SCSI  0x06   /* Initialize SCSI */
#define AHA1740CMD_SENSE 0x08   /* Read Sense Information */
#define AHA1740CMD_DOWN  0x09   /* Download Firmware (yeah, I bet!) */
#define AHA1740CMD_RINQ  0x0a   /* Read Host Adapter Inquiry Data */
#define AHA1740CMD_TARG  0x10   /* Target SCSI Command */

int aha1740_detect(Scsi_Host_Template *);
int aha1740_command(Scsi_Cmnd *);
int aha1740_queuecommand(Scsi_Cmnd *, void (*done)(Scsi_Cmnd *));
int aha1740_abort(Scsi_Cmnd *);
int aha1740_reset(Scsi_Cmnd *);
int aha1740_biosparam(Disk *, kdev_t, int*);

#define AHA1740_ECBS 32
#define AHA1740_SCATTER 16

#ifndef NULL
#define NULL 0
#endif


#define AHA1740 {NULL,  NULL,                           \
                   NULL,                                \
                   NULL,                                \
                   "Adaptec 174x (EISA)",               \
```

```
                    aha1740_detect,                        \
                    NULL,                                  \
                    NULL,                                  \
                    aha1740_command,                       \
                    aha1740_queuecommand,                  \
                    aha1740_abort,                         \
                    aha1740_reset,                         \
                    NULL,                                  \
                    aha1740_biosparam,                     \
                    AHA1740_ECBS,                          \
                    7,                                     \
                    AHA1740_SCATTER,                       \
                    1,                                     \
                    0,                                     \
                    0,                                     \
                    ENABLE_CLUSTERING}

#endif
```