From: John A. Gallant
Digital Equipment Corp.
Open Systems Group
Nashua NH

To: John Lehner X3T9.2 Committee Chairman

Date: April 16, 1991
Subject: Changes and additions for the Common Access Method Doc, X3T9.2/90-186

1) Changing offsets for Peripheral Driver Pointer and Callback on Completion.

I would like to recommend that the Peripheral Driver Pointer field be swapped with the Callback on Completion field. This places the "hits" from the pointer fields down below some "flags" fields in the other CCBs. This is a big help to the Peripheral Device drivers that use their allocated CCBs.

9.1 CAM Control block to Request I/O

<table>
<thead>
<tr>
<th>Table 9-1 SCSI I/O REQUEST CCB</th>
</tr>
</thead>
<tbody>
<tr>
<td>Size/Dir</td>
</tr>
<tr>
<td>1</td>
</tr>
<tr>
<td>4</td>
</tr>
<tr>
<td>4</td>
</tr>
<tr>
<td>4</td>
</tr>
<tr>
<td>4</td>
</tr>
<tr>
<td>4</td>
</tr>
<tr>
<td>4</td>
</tr>
<tr>
<td>4</td>
</tr>
<tr>
<td>n</td>
</tr>
</tbody>
</table>

2) Adding a field for the Auto sense transfer residual count.

There is currently a field for the residual data transfer in the SCSI 10 CCB. I would like to recommend that a single byte field for the residual count for the autosenso transfer be added. Like the residual data transfer field this will be a two complement form number for the number of autosenso bytes that were not transferred. I have also added the word "Data" to the residual data transfer field to differentiate them.

<table>
<thead>
<tr>
<th>Table 9-1 SCSI I/O REQUEST CCB</th>
</tr>
</thead>
<tbody>
<tr>
<td>Size/Dir</td>
</tr>
<tr>
<td>1</td>
</tr>
<tr>
<td>4</td>
</tr>
<tr>
<td>4</td>
</tr>
<tr>
<td>4</td>
</tr>
<tr>
<td>4</td>
</tr>
<tr>
<td>4</td>
</tr>
<tr>
<td>4</td>
</tr>
<tr>
<td>4</td>
</tr>
<tr>
<td>n</td>
</tr>
</tbody>
</table>

The following changes will be needed in the description sections.

9.1.19 Data Transfer Residual Length

This field contains the difference in two complement form of the number of data bytes transferred by the HBA compared with the number of bytes requested by the CCB.

9.1.23 Sense Info Residual Length

This field contains the difference in two complement form of the number of sense bytes transferred by the HBA compared with the number of sense bytes requested in the Sense Info Buffer Length.

3) I would like to present a new C include file to replace the current one in Annex C. This include file represents the CAM document at Rev. 2.3.

/* cam.h

/* This file contains the definitions and data structures for the CAM Subsystem interface. This contents of this file should match what data structures and constants that are specified in the CAM document, X3T9.2/90-186 Rev 2.3 that is produced by the HCA and the SCSI-2 committee.

Modification History

<table>
<thead>
<tr>
<th>Version</th>
<th>Date</th>
<th>Who</th>
<th>Reason</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.00</td>
<td>03/28/90</td>
<td>Jag</td>
<td>Creation date. Just to start the process.</td>
</tr>
<tr>
<td>1.01</td>
<td>05/23/90</td>
<td>Jag</td>
<td>Modified for Rev. 1.7 April 1990. Minor mods, added common prefix to all the structure fields. Added target ID and LUN to the header.</td>
</tr>
<tr>
<td>1.02</td>
<td>06/05/90</td>
<td>Jag</td>
<td>Updated to Rev. 2.1 October 1990.</td>
</tr>
<tr>
<td>1.03</td>
<td>10/18/90</td>
<td>Jag</td>
<td>Updated to Rev. 2.2 November 1990. Changed uchar, short, long to u[char, short, long] for more SVID/BSD consistency. Replaced &quot;int&quot; in the CAM EDIT ENTRY structure. Added SUCCESS/FAILDUE to the OSD sect. Moved SIM_ENTRY to the main area. Made the async field in EDT a pointer. Forcs long word alignment to the SIM private data space in the IO CCB. Updated to Rev. 2.3 January 1991. Added reserved fields, Target mode and HBA Engine defines.</td>
</tr>
<tr>
<td>1.04</td>
<td>11/05/90</td>
<td>Jag</td>
<td></td>
</tr>
<tr>
<td>1.05</td>
<td>11/07/90</td>
<td>Jag</td>
<td></td>
</tr>
<tr>
<td>1.06</td>
<td>12/10/90</td>
<td>Jag</td>
<td></td>
</tr>
<tr>
<td>1.07</td>
<td>03/22/91</td>
<td>Jag</td>
<td></td>
</tr>
<tr>
<td>1.08</td>
<td>04/24/91</td>
<td>Jag</td>
<td></td>
</tr>
</tbody>
</table>
element in the S/G list a char pointer.

Fixed CCB_PATHING, removed the u_long

*cam_featureFlags.*

/*
 *
 /* Defines for the XPT function codes, Table 3-2 in the CAM spec. */
 /* Common function commands, 0x00 - 0xff */
 #define XPT CDB 0x00 /* Execute CDB */
 #define XPT_SCSI IO 0x01 /* Execute the requested SCSI IO */
 #define XPT_DEV_TYPE 0x02 /* Get the device type information */
 #define XPT_PATH TNG 0x03 /* Path Inquiry */
 #define XPT_HALF_BNM 0x04 /* Release the SIM queue that is frozen */
 #define XPT_SAVED CDB 0x05 /* Set Async callback parameters */
 #define XPT_BNDEV 0x06 /* Set the device type information */
 * /

/* XPT SCSI control functions, 0x10 - 0x1f */
#define XPT_ABORT 0x10 /* Abort the selected CDB */
#define XPT_RESET BUS 0x11 /* Reset the SCSI bus */
#define XPT_RESET DEV 0x12 /* Reset the SCSI device, BDR */
#define XPT_TERM IO 0x13 /* Terminate the I/O process */

/* HBA engine commands, 0x20 - 0x2f */
#define XPT_ENG INQ 0x20 /* HBA engine inquiry */
#define XPT_ENG EXEC 0x21 /* HBA execute engine request */
#define XPT BNLM 0x23 /* Enable LUN, Target mode support */
#define XPT_TARG TIE 0x31 /* Execute the target ID request */

/* XPT FUNC */
#define XPT FUNC 0x2f /* TEMPLATE */
#define XPT VUNITIQUE 0x80 /* All the rest are vendor unique commands */

/* General allocation length defines for the CDB structures. */
#define LOCDBLEN 12 /* Space for the CDB bytes/pointer */
#define VUNITIQUE 14 /* Vendor Unique */
#define SIM I D 16 /* ASCII string len for SIM ID */
#define SIM I D 16 /* ASCII string len for HBA ID */
#define SIM PRIV 50 /* Length of SIM private data area */

/* Structure definitions for the CAM control blocks, CDBs for the subsystem. */

/* Common CDB header definition. */
typedef struct cdb_header {
  u_char *cDB_hdr; /* The address of this CDB */
  u_char *cDB_cdb_len; /* Length of the entire CDB */
  u_char *cDB_func_code; /* XPT function code */
  u_char *cDB_status; /* Returned CAM subsystem status */
  u_char *cDB_savev; /* Saved CAM subsystem status */
  u_char *cDB_path ID; /* Path ID for the request */
  u_char *cDB_target_dev; /* Target device ID */
  u_char *cDB_target LUN; /* Target LUN number */
  u_char *cDB_management FLGS; /* Flags for operation of the subsystem */
} CDB_HEADER;

/* Common XPT functions. */

/* Union definition for the CDB space in the SCSI I/O request CDB */
typedef union cdb_un

u_char *cDB_hdr;

/* Pointer to the CDB bytes to send */
/* Area for the CDB to send */
/* CDB_LEN; */

/* Get device type CDB */
typedef struct cdb_getDev {
  CDB_HEADER cDB hdr;
  char *cDB_targ_LUN; /* Parity device type from the TLM */
} CDB_GETDEV;

/* Path inquiry CDB */
typedef struct cdb_pathInq {
  CDB_HEADER cDB hdr;
  u_char *cDB_func_vnum; /* Version number for the SIM/HBA */
  u_char *cDB_HBA_IngRef; /* Flags for target mode support */
  u_char *cDB_HBA Mac; /* Misc HBA feature flags */
  u_char *cDB_HBA Mac; /* HBA engine count */
  u_char *cDB_HBA Mac; /* Vendor unique capabilities */
  u_char *cDB_HBA Mac; /* Size of SIM private data area */
  u_char *cDB_HBA Mac; /* Event cap. for Async Callback */
  u_char *cDB_HBA Mac; /* Highest path ID in the subsystem */
  u_char *cDB_HBA Mac; /* ID of the HBA on the SCSI bus */
  u_char *cDB_HBA Mac; /* Reserved field, for alignment */
  u_char *cDB_HBA Mac; /* Vendor ID of the SIM */
  u_char *cDB_HBA Mac; /* Vendor ID of the HBA */
  u_char *cDB_HBA Mac; /* For the OBD specific area */
} CDB_PATHING;

/* Release SIM Queue CDB */
typedef struct ccb_relSim {
  CDB_HEADER cDB hdr;
  CDB_REL_System;
} CDB_REL_System;

/* SCSI I/O Request CDB */
typedef struct ccb_scsmemo {
  CDB_HEADER cDB hdr;
  u_char *cDB_func_vnum; /* CDB header */
  u_char *cDB_command; /* CDB command */
  u_char *cDB data out; /* Data to be written */
  u_char *cDB data in; /* Data to be read */
  u_char *cDB path ID; /* Path ID for the request */
  u_char *cDB target dev; /* Target device ID */
  u_char *cDB target LUN; /* Target LUN number */
  u_char *cDB_management FLGS; /* Flags for operation of the subsystem */
} CDB_HEADER;

/* Common XPT functions. */

/* Union definition for the CDB space in the SCSI I/O request CDB */
typedef union cdb_un

/* Unix OSD defines and data structures. */

#define INQLEN 36  /* Inquiry string length to store. */
#define CAM_SUCCESS 0 /* For signaling general success */
#define CAM_FAILURE 1  /* For signaling general failure */
#define CAM_FALSE 0  /* General purpose flag value */
#define CAM_TRUE 1  /* General purpose flag value */
#define XPT_CCB_INVALID -1 /* for signaling a bad CCB to free */

/* General union for kernel space allocation. Contains all the possible CCB structures. This union should never be used for manipulating CCB's it's only use is for the allocation and deallocation of raw CCB space. */

typedef union ccb_size_union
{
    CCB_SCSI10 cso; /* Please keep this first, for debug/print */
    CCB_DETDEV cdp;
    CCB_PATHINFO cpi;
    CCB_RELSIM crs;
    CCB фингкc фес;
    CCB_RELSIM cbs;
    CCB_RELSIM cdb;
    CCB_ABORT cib;
    CCB_RESETBUS crb;
    CCB_RESETDEV crud;
    CCB_TERMIO cti;
    CCB_EN LUN cel;
    CCB_EN NSW cel;
    CCB_EN EXEC ces;
} CCB_SIZE_UNION;

/* The typedef for the Async callback information. This structure is used to store the supplied info from the Set Async Callback CCB, in the EDT table in a linked list structure. */

typedef struct async_info
{
    struct async_info *cam_async_next; /* pointer to the next structure */
    u_long cam_event_enable; /* Event enables for Callback resp */
    void (*cam_async_func)(); /* Async Callback function address */
    u_long cam_async_len; /* Length of "information" buffer */
    u_char *cam_async_ptr; /* Address for the "information" */
} ASYNC_INFO;

/* The CAM EDT table contains the device information for all the devices, SCSI ID and LUN, for all the SCSI buses in the system. The table contains a CAM_EDT_ENTRY structure for each device on the bus. */

typedef struct cam_edt_entry
{
    long cam_tln_found; /* Flag for the existence of the target/LUN */
    ASYNC_INFO *cam_info; /* Async callback list info for this B/L */
    u_long cam_owner_tag; /* Tag for the peripheral driver's ownership */
    char cam_c1q_data[INQLEN]; /* storage for the inquiry data */
} CAM_EDT_ENTRY;

John A. Gallant
Senior Software Engineer - Ultrix Engineering Group
jag@decvax.dec.com

Digital Equipment Corp. (603) 991-2472
"A beautiful theory, killed by a nasty, ugly, little fact.
" Thomas H. Huxley