To: X3T9.2 Committee Membership X3T9.2/93-082r0

From: Bill Dallas <dallas@zk3.dec.com>

Date: Mon, 10 May 1993 12:13:26 -0400

Subject: Minutes of CAM-2 Study Group Meeting (04/20/93)

Agenda
1.0 Opening Remarks
2.0 Attendance and Introductions
3.0 Call for items to be placed on the current agenda
4.0 Status of Action Items
5.0 SYNC negotiation and 16 and 32 wide transfers.
6.0 Proposed changes to CAM-1 Rev 3.0
7.0 Defining CCB’s and function codes
8.0 Review of Action Items
9.0 Next Meeting Items

1.0Opening Remarks

Bill Dallas, Editor of the CAM-2 Study Group, called the meeting to order at 5:00 p.m., Tuesday April 20, 1993.

It was noted that the meeting was poorly attended, and a copy of the attendance list was circulated for attendance and corrections. The customary procedure of people attending introducing themselves was dispensed with due the three people attending all knew each other.

2.0Attendance

The following people attended the meeting:

Name	Organization	Telephone
Mr. Scott Smyers	Apple Computer	(408) 974-7057
Mr. Charles Monia	Digital Equipment Corp.	(508) 841-6757
Mr. William Dallas	Digital Equipment Corp.	(603) 881-2508

3 People Present

3.0Call for items to be placed on the current agenda

No new agenda items.

4.0Status of Action Items

An overview was given by Bill Dallas on what the adhoc group has been doing on the rewrite of section 10 of the CAM 1 document. The overview went into the functionality and behavior model of the target mode rewrite.

It was suggested that target mode support both non disconnects privilege and the Terminate I/O process message. The current model does not support that functionality. The suggestion will be presented to the adhoc group.

Adhoc Participants:
Randell JesupCommodore
Johnathan Vail Tegra
Maria Vella DEC
John Gallant DEC
Bill Dallas DEC
Rich Napolitano napcon Associates
Rich Whalen DEC

5.0 SYNC negotiation and 16 and 32 wide transfers.

Discussion on a problem with 16 and 32 wide transfers. Currently CAM controls SYNC transfers at the peripheral driver level by use of CAM Flags. There currently is no way to control 16 and 32 transfers.

It was decided to expand the definition of the CAM flags of Initiate Synchronous Transfers and Disable Synchronous Transfers. The expanded definition for Initiate Synchronous Transfers will effectively be:
Negotiate best transfer parameters (sync and wide(if supported)).
Responsibility shall be at the SIM for negotiation.

The expanded definition for Disable Synchronous Transfers will effectively be:
Negotiate worst transfer parameters (async and narrow).
Responsibility shall be at the SIM for negotiation.

6.0 Proposed changes to CAM-1 Rev 3.0

Due to a number of questions the editor has received regarding clarification of the CAM 1 document. The editor proposed changes to the CAM 1 document. These changes were approved by the working group.

The proposed changes are as follows:

****
WAS
****

TABLE 9-4 CAM STATUS

00h	Request in progress
01h	Request completed without error
02h	Request aborted by host
03h	Unable to Abort Request
04h	Request completed with error
05h	CAM Busy
06h	Invalid Request
07h	Invalid Path ID
08h	SCSI device not installed
09h	Unable to Terminate I/O Process
0Ah	Target Selection Timeout
0Bh	Command Timeout
0Ch	reserved
0Dh	Message Reject received
0Eh	SCSI Bus Reset Sent/Received
0Fh	Uncorrectable Parity Error Detected
10h	Autosense Request Sense Cmd Failed
11h	No HBA detected
12h	Data OverRun/UnderRun
13h	Unexpected Bus Free
14h	Target bus phase sequence failure
15h	CCB Length Inadequate
16h	Cannot Provide Requested Capability

17h	Bus Device Reset Sent
18h	Terminate I/O Process
19-37h	reserved
	Target Mode Only Status
38h	Invalid LUN
39h	Invalid Target ID
3Ah	Function not Implemented
3Bh	Nexus not Established
3Ch	Invalid Initiator ID
3Dh	SCSI CDB Received
3Eh	LUN Already Enabled
3Fh	SCSI bus Busy
+40H	to indicate that SIM Queue is frozen
+80h	to indicate that Autosense is valid

- 12h Data OverRun: target transferred more data bytes than peripheral driver indicated in the CCB.

****

Proposed

****

TABLE 9-4 CAM STATUS

00h	Request in progress
01h	Request completed without error
02h	Request aborted by host
03h	Unable to Abort Request
04h	Request completed with error
05h	CAM Busy
06h	Invalid Request
07h	Invalid Path ID
08h	SCSI device not installed
09h	Unable to Terminate I/O Process
0Ah	Target Selection Timeout
0Bh	Command Timeout
0Ch	reserved
0Dh	Message Reject received
0Eh	SCSI Bus Reset Sent/Received
0Fh	Uncorrectable Parity Error Detected
10h	Autosense Request Sense Cmd Failed
11h	No HBA detected
12h	Data OverRun
13h	Unexpected Bus Free
14h	Target bus phase sequence failure
15h	CCB Length Inadequate
16h	Cannot Provide Requested Capability
17h	Bus Device Reset Sent
18h	Terminate I/O Process
19-37h	reserved
	Target Mode Only Status
38h	Invalid LUN
39h	Invalid Target ID
3Ah	Function not Implemented
3Bh	Nexus not Established
3Ch	Invalid Initiator ID
3Dh	SCSI CDB Received
3Eh	LUN Already Enabled
3Fh	SCSI bus Busy
+40H	to indicate that SIM Queue is frozen
+80h	to indicate that Autosense is valid

```
+------+---------------------------------------+
```

- 12h Data OverRun: target transferred more data bytes than peripheral driver indicated in the CCB.

****
WAS
****
9.1.3.3 Byte 3 Bits

The Pointer fields are set up to have one characteristic. If a bit is set to 1 it means the pointer contains a Physical Address. If set to 0 it means the pointer contains a Virtual Address. If the SIM needs an address in a different form to that provided, it should be converted by the SIM (using OSD facilities) and stored in Private Data.

****
Proposed
****
9.1.3.3 Byte 3 Bits

The Pointer fields are set up to have one characteristic. If a bit is set to 1 it means the pointer contains a Physical Address. If set to 0 it means the pointer contains a Virtual Address. If the pointer points to a list defined by CAM, the pointers within the list shall have the same attributes. If the SIM needs an address in a different form to that provided, it should be converted by the SIM (using OSD facilities) and stored in Private Data.

****
WAS
****
11.2 Execute Engine Request (Optional)

To accommodate buffering associated with the engine, the CAM Flag SG List/Data set to 1=Engine is used to specify that the normal Data Buffer Pointer is actually a physical address in the buffer space of the engine.

There are four modes associated with engine processing established by CAM Flags:

- A Direction setting of Out is used to Encrypt or Compress the data
- A Direction setting of In is used to Decrypt or Decompress the data
- Synchronize is used in conjunction with In or Out to flush any residual bits prior to terminating engine processing.

The Execute Engine Request CCB activates the engine to perform the requested function. Some functions change the data size e.g. a compression engine reduces the size of data prior to transmission over SCSI.

****
Proposed
****

11.2 Execute Engine Request (Optional)

To accommodate buffering associated with the engine, the CAM Flag SG List/Data (bit 7) set to 1=Engine is used to specify that the normal Data Buffer Pointer is actually an address in the buffer space of the engine. The CAM Flag SG List/Data (bit 5) shall signify whether the address is Virtual or Physical.

There are four modes associated with engine processing established by CAM Flags:

- A Direction setting of Out is used to Encrypt or Compress the data
- A Direction setting of In is used to Decrypt or Decompress the data
- Synchronize is used in conjunction with In or Out to flush any residual bits prior to terminating engine processing.

The Execute Engine Request CCB activates the engine to perform the requested function. Some functions change the data size e.g. a compression engine reduces the size of data prior to transmission over SCSI.

7.0 Defining CCB’s and function codes

In the January study group, a CCB header was approved for CAM-2. The editor of CAM-2 discovered some problems with that header and proposed some changes to it. The new proposed CCB header kept all the fields but the arrangement changed. The CAM study group approved these changes.

It was also decided to define all CCB structure members types declarations as a typedef. This will allow transportability of software to machine platforms with varying word sizes. While the CCB’s size varies based on machine word size, they are of fixed size and structure member offsets are fixed for specific machine word size.

While the typedef and the rules have not been presented to the working group for approval, I have included them into the minutes for clarification.

The left hand side of the typedef is Operating system dependent and platform dependent.

```
typedef charI_8;/* 8 bits*/
typedef unsigned charU_8; /* 8 bits*/
typedef shortI_16;/* 16 bits*/
typedef unsigned shortU_16;/* 16 bits              */
typedef longI_32;/* 32 bits*/
typedef unsigned longU_32;/* 32 bits*/
```

All (x)_8 (chars) shall being on a 8 bit boundary and shall be 8 bits length.

All (x)_16 (shorts) shall being on a on a 16 bit boundary and shall be 16 bits in length.

All x_32 (longs, ints, (OSD)) shall being on a 32 bit boundary and shall be 32 bits in length.

All pointers shall begin on a machine word boundary shall be of machine word size.

All structures and unions and arrays shall begin and end on a machine word boundary. If they don’t they shall be padded out to the machine word boundary.

All CCB headers shall begin on a machine word boundary.

Due to compiler differences between machine platforms and O.S’s if the next defined member type does not align with the specified alignment then it shall be padded to force alignment.

Padding shall use the following type:

```
u_8          :8;     /* Alignment Padding            */
```

New Proposed CAM 2 CCB header

```
/* Common CCB CAM 2 header definition.
 * For 32 bit machines
 */
typedef struct ccb_header2 {
            void *my_addr;      /* The address of this CCB */
    U_16 cam_ccb_len;        /* Length of the entire CCB */
    U_8 cam_2_id;            /* XPT function code/CAM 2 signifier */
    U_8 cam_status;          /* Returned CAM subsystem status */
    U_16 cam_cam2_func_code; /* The actual CAM 2 function Code */
    U_16 cam_ccb_vers;       /* CCB version number   */
    U_8 cam_path_id[8];      /* Path ID for the request */
    U_8 cam_target_id[8];    /* Target device ID */
    U_8 cam_target_lun[8];   /* Target LUN number */
    U_32 cam_flags;          /* Flags for operation of the subsys */
}CCB_HEADER2; /* structure ends on 32 bit boundary */
```

This size of a 32 bit machine CCB_HEADER2 shall be 40 bytes.

New Proposed CAM 2 CCB header

```
/* Common CCB CAM 2 header definition.
 * For 64 bit machines
 */
typedef struct ccb_header2 {
            void *my_addr;           /* The address of this CCB */
    U_16 cam_ccb_len;        /* Length of the entire CCB */
    U_8 cam_2_id;            /* XPT function code/CAM 2 signifier */
    U_8 cam_status;          /* Returned CAM subsystem status */
    U_16 cam_cam2_func_code; /* The actual CAM 2 function Code */
    U_16 cam_ccb_vers;       /* CCB version number   */
    U_8 cam_path_id[8];      /* Path ID for the request */
    U_8 cam_target_id[8];    /* Target device ID */
    U_8 cam_target_lun[8];   /* Target LUN number */
    U_32 cam_flags;          /* Flags for operation of the subsys */
    U_8:8;     /* Alignment Padding            */
    U_8:8;     /* Alignment Padding            */
    U_8:8;     /* Alignment Padding            */
    U_8:8;     /* Alignment Padding            */
}CCB_HEADER2; /* structure ends on 64 bit boundary, forced by padding */
```

The size of a 64 bit machine CCB_HEADER2 shall be 48 bytes.

It should be noted that the fields cam_path_id[8] cam_target_id[8] cam_target_lun[8], will probably placed into unions, with defined frontend software macros to allow transportability and field transparency.

If any of person would like to place an item on the agenda for the next CAM-2 study group meeting, please contact the editor.

Bill Dallas internet dallas@wasted.enet.dec.com
voice (603) 881-2508
fax(603) 881-2257