SERIAL BUS PROTOCOL FUNDAMENTALS

GERALD MARAZAS
IBM, BOCA RATON, FLORIDA
(C407) 982-4423 VOICE
(C407) 982-0067 FAX
MARAZAS@BCRUMP2.VNET.IBM.COM

NEXT SBP EDITOR'S MEETING
→ JAN 27 & 28, 1993 (WED. & THURS)
→ 9 AM - 5 PM EACH DAY
HOSTED BY SCOTT SMYERS
APPLE COMPUTER
CUPERTINO, CALIFORNIA
SERIAL BUS PROTOCOL

1. Elements required in:
   - the target
   - the initiator

2. The command block chain

3. The tap protocol for command delivery

4. The "log-in" procedure

5. Request for tap slots

6. Dedicated tap slots vs pool tap slots

7. The "sign-in" procedure

8. Asynchronous events

9. First failure register

10. Status delivery

11. Isochronous
Serial Bus Protocol

A target has:
- Normal FIFO
- Urgent FIFO
- ACA FIFO

A FIFO has tap slots

Configuration ROM
- Base address of the FIFO

FIFO address has the structure

```
| INIT-ID | TYPE |
```

Initiator-ID = 8 bits, assigned by the target

Working storage to hold command blocks

Optionally— Isochronous control reg. first failure register
SERIAL BUS PROTOCOL

1. AN INITIATOR HAS:
2. STATUS FIFO
3. COMMAND BLOCK CHAINS
4. Optionally an I SOCHRONOUS CONTROL REGISTER
THE INITIATOR MAY DIRECT MULTIPLE CHAINS TO THE SAME TARGET

COMMAND BLOCKS IN THE SAME CHAIN:
- MAY INCLUDE DIFFERENT QUEUE TYPES
- MAY BE DIRECTED TO DIFFERENT LUNA AT SAME TARGET
- INDICATE FETCH POLICY

EACH CHAIN REQUIRES 1 TAP SLOT

A CHAIN MAY GO TO THE URGENT FIFO
SERIAL BUS PROTOCOL

COMMAND DELIVERY PROTOCOL

INITIATOR

SEND TAP PACKET
(MUST HAVE INIT-ID)
DEDICATED TAP SLOT = ?

TARGET

STORE TAP PACKET INTO TAP SLOT

REQUEST FIRST COMMAND BLOCK

MORE = ?
END SUB CHAIN = ?
LINKED = ?
HEAD OF QUEUE = ?

REQUEST NEXT COMMAND BLOCK

REQUEST LAST COMMAND BLOCK

LATEST TIME TO RELEASE TAP SLOT
**Serial Bus Protocol**

**The Log-in Procedure**

**Initiator**

Send log-in request tap message (use init. id = 0)

Supply log-in cmd blk

Record init. id

**Target**

Accept message into fifo

Request log-in command block

Assign 8-bit id to initiator

Send data packet with init. id

Note: The same initiator is likely to have a different init. id from each target
SERIAL BUS PROTOCOL

REQUEST FOR TAP SLOTS

INITIATOR

SEND TAP SLOT
REQUEST MSG TO FIFO
USE INIT ED

TARGET

ACCEPT MESSAGE INTO FIFO

REQUEST TAP SLOT REQUEST COMMAND BLOCK

SUPPLY TAP SLOT REQUEST COMMAND BLOCK

GATHER UP TAP SLOTS TO BE DEDICATED TO THE INITIATOR

SEND DATA PACKET

USE ALLOCATION OF DEDICATED TAP SLOTS TO MANAGE SENDING OF TAP PACKETS

OLD # SLOTS
NEW # SLOTS

7
<table>
<thead>
<tr>
<th>INITIATOR-A</th>
<th>TARGET</th>
<th>DEDICATED TO INIT-A</th>
<th>POOL SLOTS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Send Tap</td>
<td>Must Accept</td>
<td>2</td>
<td>4</td>
</tr>
<tr>
<td>Send Tap</td>
<td>Must Accept</td>
<td>0</td>
<td>4</td>
</tr>
<tr>
<td>Send Tap</td>
<td>May Reject</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td>Send Tap</td>
<td>May Reject</td>
<td>0</td>
<td>2</td>
</tr>
<tr>
<td>Send Tap</td>
<td>Did Reject</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

Try to send Tap again may accept
**SIGN-IN PROCEDURE**

**INITIATOR**
- Send Sign-In Request Tap Message
  - Use INIT-ID

**TARGET**
- Accept Message into FIFO
  - Request Sign-In Command Block
  - Record AEN:
    - Status FIFO
    - Sense Buffer
      - ...
  - AEN Occurs
    - Send Status to AEN Status FIFO
    - Send Sense Data to Sense Buf
  - Retire Status FIFO
  - Retire Sense Buf
Serial Bus Protocol

Asynchronous Events

- **Init-A**
  - Yes:
    - Target

- **Init-B**
  - Yes:
    - Target

- **Init-C**
  - No:
    - A
    - B

**A** and **B** sign-in for AEN

- **A** does not

**AEN Occurs**

- Notify @ A
- Retire @ A
- Notify @ B
- Retire @ B

**Sign-In from A**

**Only A makes Sign-In**
**Serial Bus Protocol**

**First Failure Register**, FF-REG

**Target**

**FF-REG Unlocked**

**Initiator-A**

Supply Commands

**Failure Occurs**

Record in FF-REG

**Unlock Request to FF-CNTL**

**Time Out**

Read Request to FF-REG

**Unlock Request to FF-CNTL**

Supply Contents of FF-REG

**Target Unlocks FF-REG**

94 (M)
SERIAL BUS PROTOCOL

STATUS DELIVERY

1. Each command block contains:
   - Status FIFO
   - Sense Buffer

2. Initiator may use unique status FIFO in each command block.

3. Initiator may use same status FIFO for several command blocks.

4. Write operation to status FIFO may or may not cause
   an interrupt to the initiator.

5. Initiator must receive status block from the target.

6. Initiator may tell target not to send a status block.

95(12)
ISOCHRONOUS

- PORT-SETUP COMMAND BLOCK (NO CDB)
  LOTS OF PARMS

- PORT-APPEND COMMAND BLOCK (CARRIES A CDB)
  → ENABLES ISO XFER, DOES NOT CAUSE IT!

- WRITE OPERATION TO ISOCHRONOUS CONTROL REGISTER
  START ISO XFER
  PAUSE ISO XFER

- PORT-TEAR-DOWN COMMAND BLOCK (NO CDB)