Linux Device Drivers: Case Study of a Storage Controller

Manolis Marazakis
FORTH-ICS (CARV)
IOP348-based I/O Controller
Programmable I/O Controller

Continuous Data Protection:
Versioning (snapshots), Migration, Encryption, Data Integrity
Misc. Storage Devices

SSD: 16+ GBytes
READ: 110–270 MB/sec
WRITE: 70-160 MB/sec
6000 – 10000 random IOPS
< 2 Watts

HDD: 500+ GBytes
READ/WRITE: 80-125 MB/sec
120-230 random IOPS
8.5-13.5 Watts

- Acceleration of commercial I/O-intensive workloads
- Reduction of power consumption
IOP348 – Boot Loader

RedBoot> fis list
Name        FLASH addr  Mem addr    Length      Entry point
(reserved)  0xF0000000  0xF0000000  0x00200000  0x00000000
RedBoot     0xF0200000  0xF0200000  0x00040000  0x00000000
LinuxKernel 0xF0240000  0x01008000  0x00260000  0x01008000
FIS directory 0xF07E0000  0xF07E0000  0x0001F000  0x00000000
RedBoot config  0xF07FF000  0xF07FF000  0x00001000  0x00000000

RedBoot> fis load LinuxKernel
RedBoot> exec -c "console=ttyS0,115200 root=/dev/sda1" 0x1008000
Using base address 0x01008000 and length 0x00241f10
Disk Interface Technologies (I)

**ATA**: Advanced Technology Attachment

**SATA**: Serial ATA

**SCSI**: Small Computer System Interface

**FC**: Fibre Channel
## Disk Interface Technologies (II)

<table>
<thead>
<tr>
<th>Performance</th>
<th>Parallel ATA</th>
<th>Parallel SCSI</th>
<th>Fibre Channel</th>
<th>SATA</th>
<th>Serial Attached SCSI</th>
</tr>
</thead>
<tbody>
<tr>
<td>Maximum Speed</td>
<td>100 MB/s</td>
<td>320 MB/s</td>
<td>4.2 Gb/s (400 MB/s)</td>
<td>3.0 Gb/s (300 MB/s)</td>
<td>3.0 Gb/s (300 MB/s)</td>
</tr>
<tr>
<td>Topology</td>
<td>Shared bus master/slave</td>
<td>Shared bus</td>
<td>Arbitrated loop/switched fabric</td>
<td>Point-to-point</td>
<td>Point-to-point</td>
</tr>
<tr>
<td>Number of Devices</td>
<td>2</td>
<td>15</td>
<td>1,000s</td>
<td>up to 15</td>
<td>100s</td>
</tr>
</tbody>
</table>
Example Storage Area Network Configuration

- Node 1 and Node 2 are connected to Switches.
- Controller module and Expansion tray.
- Dual controllers and Dual Fabric with 2Gb Dual channel HBAs.

Diagram showing various components of a Storage Area Network configuration.
RAID: Redundant Array of Independent *(Inexpensive)* Disks

- Striping for parallel data transfer & load balancing
- Redundancy for failure protection
- Error-correcting codes
RAID Levels

RAID Level 0: Non-redundant

RAID Level 1: Mirroring

RAID Level 2: Byte-Interleaved, ECC

RAID Level 3: Byte-Interleaved, Parity

RAID Level 4: Block-Interleaved, Parity

RAID Level 5: Block-Interleaved, Distributed Parity
SCSI (Small Computer System Interface)
### SCSI: Command Sets

<table>
<thead>
<tr>
<th>Command name</th>
<th>OpCode</th>
<th>Command Support</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>FORMAT UNIT</td>
<td>04h</td>
<td>O</td>
<td>RBC</td>
</tr>
<tr>
<td>INQUIRY</td>
<td>12h</td>
<td>M</td>
<td>SPC-2'</td>
</tr>
<tr>
<td>MODE SELECT(6)</td>
<td>15h</td>
<td>M</td>
<td>SPC-2'</td>
</tr>
<tr>
<td>MODE SENSE(6)</td>
<td>1Ah</td>
<td>M</td>
<td>SPC-2'</td>
</tr>
<tr>
<td>PERSISTENT RESERVE IN</td>
<td>5Eh</td>
<td>O</td>
<td>SPC-2'</td>
</tr>
<tr>
<td>PERSISTENT RESERVE OUT</td>
<td>5Fh</td>
<td>O</td>
<td>SPC-2'</td>
</tr>
<tr>
<td>PREVENT(ALLOW MEDIUM REMOVAL</td>
<td>1Eh</td>
<td>N/A</td>
<td>SPC-2'</td>
</tr>
<tr>
<td>READ (10)</td>
<td>26h</td>
<td>M</td>
<td>RBC</td>
</tr>
<tr>
<td>READ CAPACITY</td>
<td>25h</td>
<td>M</td>
<td>RBC</td>
</tr>
<tr>
<td>RELEASE(6)</td>
<td>17h</td>
<td>O</td>
<td>SPC-2'</td>
</tr>
<tr>
<td>REQUEST SENSE</td>
<td>03h</td>
<td>O</td>
<td>SPC-2'</td>
</tr>
<tr>
<td>RESERVE(6)</td>
<td>16h</td>
<td>O</td>
<td>SPC-2'</td>
</tr>
<tr>
<td>START STOP UNIT</td>
<td>18h</td>
<td>M</td>
<td>RBC</td>
</tr>
<tr>
<td>SYNCHRONIZE CACHE</td>
<td>35h</td>
<td>O</td>
<td>RBC</td>
</tr>
<tr>
<td>TEST UNIT READY</td>
<td>00h</td>
<td>M</td>
<td>SPC-2'</td>
</tr>
<tr>
<td>VERIFY (10)</td>
<td>2Fh</td>
<td>M</td>
<td>RBC</td>
</tr>
<tr>
<td>WRITE (10)</td>
<td>2Ah</td>
<td>M</td>
<td>RBC</td>
</tr>
<tr>
<td>WRITE BUFFER</td>
<td>3Bh</td>
<td>M</td>
<td>SPC-2'</td>
</tr>
</tbody>
</table>

[ RBC - reduced block command set ]
SAS: Serial Attached SCSI

- Desktop PC / Workstation
- Mainstream Servers / Direct Attached Storage
- Storage Area Network

- PC Chipset
- SAS HBA
- Server chipset with SAS support, SAS HBA card, or SAS RAID card to drives or JBODs
- FC HBA to external storage

- SATA
  - Desktop-class disk drives (single port)
  - Compatible physical layer between SATA and SATA

- SAS
  - Enterprise-class disk drives (single or dual port)
  - Similar HDD ASIC and firmware design between SAS and FC

- FC
  - Enterprise-class disk drives (dual port)
  - Configuration
  - Flexibility & Simplicity
  - One host design accommodates SATA for
  - Low Cost Bulk Storage or
  - SAS for Performance & Reliability in Mission Critical Apps
SAS Host-Based Adapters

**Direct attcn** = Number of drives limited to number of ports in the HBA

**Expander attcn** = More drives than HBA ports

- SAS HBA (initiator)
  - SAS physical link
  - SATA physical links
  - SAS physical link
  - SAS physical link(s)

- SAS Drive
- SATA Drive

- SSP (Serial SCSI Protocol) used to communicate with SAS drives
- SATA (Serial ATA) used to communicate with SATA drives over SATA physical links

- Expander(s)
- SAS physical links
- SATA physical links

- STP (Serial ATA Tunneling Protocol) used over SAS physical links when communicating with SATA drives through expanders

- SAS Drives
- SATA Drives
SAS: Read Command Sequence

**SSP initiator port**

- Send SCSI Command ()
- Command Complete Received ()
- time

**SSP target port**

- COMMAND frame
- DATA frame
- RESPONSE frame
- time

- SCSI Command Received ()
- Send Data In ()
- Data In Delivered ()
- SSP target port sends DATA frames until all read data has been transferred
- Send Command Complete Complete ()
Chipset
(Intel/ICH)
Chipset (AMD/NVIDIA)
Chipset (Intel/MCH)
Host-Based Adapter

CPU1

CPU2

Front Side Bus

Memory Controller Hub

I/O Controller Hub

USB PCI IDE

PCI-E or Proprietary

DDR

PCIE

Intel® 413808 HBA

PCI-E Slot

PCI-E Slot

PCI-E Slot

SAS Network

Disks

Onboard NIC
Intel 81348 Features (I)

- **Host Interface:**
  - PCI-X _or_ PCI-Express
  - 2-function PCI device: (ATU + MU, TPMI)

- **Intel XScale Processor (ARM v5tel)**
  - 2 cores, running at 1.2GHz
  - I-cache, D-cache per core (32KB each, 4-way)
  - Unified L2 cache (512KB, 8-way)
  - Inter-Processor Messaging Unit

- **Internal Busses (North, South):**
  - 128-bits wide, running at 400 MHz
  - Internal Bus System Controller: internal address bus arbitration, internal data bus arbitration, framing Address bus cycles, framing Data bus cycles, shared address & data paths

- **DDR memory controller**

- **Timers:**
  - 2 programmable timers per processor, 1 watchdog timer per processor

- **I2C Bus Interface, 2 UART’s, 16 GPIO, Peripheral Bus Interface (PBI), Performance Monitoring Unit (PMI)**
XScale Microarchitecture

Unified L2 Cache (optional)
- No L2 / 256KB / 512KB
- 8 way set associative
- write-back / write-allocate
- lockable by line
- supports coherency with other ASSP blocks
- portions may be used as SRAM
- accepts writes from other ASSP blocks

Instruction Cache
- 32KB
- 4 way set associative
- lockable by line

Data Cache
- 32KB
- 4 way set associative
- Low-Locality Reference
- lockable by line
- write-back or write-through
- supports coherency
- hit-under-miss

Memory Management
- 32 entry Instruction TLB
- 32 entry Data TLB
- Lockable by entry

Branch Target Buffer
- Branch prediction
- 128 entries

Multiply / Accumulate
- single clock throughput (16*32)
- 2 way 16-bit SIMD
- 40-bit accumulator

Software Debug
- hardware breakpoints
- Debug SRAM
- software trace buffer

Performance Monitoring

Power Management

JTAG
PCI Express: Switch + Links
PCI Express: Layers

1. Config/OS
   - PCI PnP Model (int, enum, config)
   - No OS Impact

2. S/W
   -PCI Software/Driver Model

3. Transaction
   - Packet-based Protocol

4. Data Link
   - Data Integrity

5. Physical
   - Point-to-point, serial, differential, hot-plug configurable width, inter-op form factors
   - Future speeds and encoding techniques only impact the Physical layer
3 Application DMA Channels (ADMA)
  – Dual-ported: South Bus – SDRAM
  – Support L2 cache coherence

Address Translation Unit (ATU)
  – Allow PCI Tx’s direct access to local DDR SDRAM
  – Programmable registers to control address translation

Messaging Unit (MU)
  – Data transfers between PCI system & 81348
    • Message passing, interrupt generation
  – Interrupts to notify each system when new data arrives

FSENG block: 8 SATA/SAS engines
IOP348: Functional Blocks
Address Translation Unit (ATU)
The MU is accessed by an external PCI agent via ATU.

- Message registers
- Doorbell registers
- Circular queues
- Index registers
MU: inbound & outbound queues

**Inbound:**

<table>
<thead>
<tr>
<th>Queue Mnemonic</th>
<th>Queue Name</th>
<th>Queue Size (Bytes)</th>
</tr>
</thead>
<tbody>
<tr>
<td>IWQ</td>
<td>Inbound Write Data Queue</td>
<td>4 KBytes (4*1KB)</td>
</tr>
<tr>
<td>IWADQ</td>
<td>Inbound Write Address Queue</td>
<td>4 Transaction Addresses</td>
</tr>
<tr>
<td>IRQ</td>
<td>Inbound Read Data Queue</td>
<td>4 KBytes (4*1KB)</td>
</tr>
<tr>
<td>IDWQ</td>
<td>Inbound Delayed Write address/data Queue</td>
<td>1 Transaction</td>
</tr>
<tr>
<td>ITQ</td>
<td>Inbound Transaction Queue</td>
<td>8 Addresses/Commands</td>
</tr>
</tbody>
</table>

**Outbound:**

<table>
<thead>
<tr>
<th>Queue Mnemonic</th>
<th>Queue Name</th>
<th>Queue Size (Bytes)</th>
</tr>
</thead>
<tbody>
<tr>
<td>OQW</td>
<td>Outbound Write Data Queue</td>
<td>4 KBytes (4*1024B)</td>
</tr>
<tr>
<td>OQADQ</td>
<td>Outbound Write Address Queue</td>
<td>4 Transaction Addresses</td>
</tr>
<tr>
<td>ORQ</td>
<td>Outbound Read Data Queue</td>
<td>2 or 4 KBytes (4<em>512B or 4</em>1024B)</td>
</tr>
<tr>
<td>OTQ</td>
<td>Outbound Transaction Queue</td>
<td>8 Addresses/Commands</td>
</tr>
</tbody>
</table>
Application DMA Channel (ADMA)

API: chain of Descriptors: (SRC, DST, byte-count, control-bits) + link to next descriptor
PCI Express card running with Linux

Drivers make the card appear as a SCSI block device

Take advantage of virtualized stack to provide hot updates

Any Linux Distribution available
IOP348 - Controller Glue

- Windows or Linux Host
  - IssueCxt
  - Completion
  - SCSI Stack
  - Queue Module
  - PCIe x8
  - DMA
  - SCSI to Block I/O
  - Buffer Mgmt

- Virtualization Framework
- Standard Block Modules e.g. RAID

- IOP
  - Concurrency
  - High rates
  - Critical data path

- Linux
  - Standard APIs with control command extensions

Note: The diagram illustrates the flow of data and control commands through the system, emphasizing the critical data path and high rates of operation.