New Disks for Old Heavy Iron

Dr. John G. Zabolitzky

www.cray-cyber.org

Gesellschaft für historische Rechenanlagen e.V.
Mainframe Computer 196x-198x

- Rotating Mass Storage (RMS) as premier long-term and high-volume storage medium
- Initially 38” platters, quickly moving to 14”
- Magnetic Disk Drive with movable arm dominates the industry (fixed-head drums quickly phasing out)
- removable stack of platters or fixed stack
RMS parameters

- typically, few Mbyte to fraction Gbyte
- typically, 20 msec average access time
- typically, few Mbyte/sec transfer rate
- typically, 1-10 controllers and 2-100 spindles
- typically, 1-20 platters/spindle, = 2-40 heads
RMS topology

• Mainframe connects to I/O channel
• I/O channel connects to controller
  – possibly two redundant channels/controllers
• controller connects to spindles
  – several (~8) spindles/controller
  – two controllers/spindle
RMS topology

IOU

CPU

Channel

Unit Cables

Controller

Spindles = Units
A typical Installation
Live Museum Operations

• High power consumption (xx kVA)
• Reduced times of operation
  – few hours per week
• Temperature cycles per week
  – many °C / hour twice a week
  – typically, 10°C => 28°C => 10°C
• Mechanical instability (20°*14”*10^-5=70μ)
Disk Emulation

- Replace controller by modern (DSP) SBC
- Create HW channel interface as peripheral
- Create IDE interface as peripheral
- Program controller functions as code executing on SBC
- easily implement maximum performance 16 old spindles on single modern IDE drive (16 * 500 Mbyte = 8 Gbyte)
Advantages and Disadvantages

• Very High Reliability
• Small Volume
• Low Power Consumption
• 16 cheap spindles
• ..... But cannot simultaneously reposition 16 logical spindles as original controller can !!
CDC Cyber 170 Channel

Figure 1-2. Output Pulse Characteristics

MAX VOLTAGE SWING = ± 2.7V p-p
MIN VOLTAGE SWING = ± 2.1V p-p

Measured at output pin of TR, looking into a 75 Ω impedance.
## Cyber 170 Channel

### Table 1-1. Data Channel Coaxial Cable Lines

<table>
<thead>
<tr>
<th>Input Cable</th>
<th>PIN</th>
<th>Color Code</th>
<th>Output Cable</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data bit 0</td>
<td>A</td>
<td>90</td>
<td>Data bit 0</td>
</tr>
<tr>
<td>Data bit 1</td>
<td>B</td>
<td>91</td>
<td>Data bit 1</td>
</tr>
<tr>
<td>Data bit 2</td>
<td>C</td>
<td>92</td>
<td>Data bit 2</td>
</tr>
<tr>
<td>Data bit 3</td>
<td>D</td>
<td>93</td>
<td>Data bit 3</td>
</tr>
<tr>
<td>Data bit 4</td>
<td>E</td>
<td>94</td>
<td>Data bit 4</td>
</tr>
<tr>
<td>Data bit 5</td>
<td>F</td>
<td>95</td>
<td>Data bit 5</td>
</tr>
<tr>
<td>Data bit 6</td>
<td>H</td>
<td>96</td>
<td>Data bit 6</td>
</tr>
<tr>
<td>Data bit 7</td>
<td>J</td>
<td>97</td>
<td>Data bit 7</td>
</tr>
<tr>
<td>Data bit 8</td>
<td>K</td>
<td>98</td>
<td>Data bit 8</td>
</tr>
<tr>
<td>Data bit 9</td>
<td>L</td>
<td>99</td>
<td>Data bit 9</td>
</tr>
<tr>
<td>Data bit 10</td>
<td>M</td>
<td>900</td>
<td>Data bit 10</td>
</tr>
<tr>
<td>Data bit 11</td>
<td>N</td>
<td>901</td>
<td>Data bit 11</td>
</tr>
<tr>
<td>Active</td>
<td>P</td>
<td>902</td>
<td>Active</td>
</tr>
<tr>
<td>Inactive</td>
<td>R</td>
<td>903</td>
<td>Inactive</td>
</tr>
<tr>
<td>Full</td>
<td>S</td>
<td>904</td>
<td>Full</td>
</tr>
<tr>
<td>Empty</td>
<td>T</td>
<td>905</td>
<td>Empty</td>
</tr>
<tr>
<td>Clock (10 MHz)</td>
<td>U</td>
<td>906</td>
<td>Function</td>
</tr>
<tr>
<td>Clock (1 MHz)</td>
<td>V</td>
<td>907</td>
<td>Master Clear</td>
</tr>
<tr>
<td>Input Data Parity</td>
<td>W</td>
<td>908</td>
<td>Output Data Parity</td>
</tr>
</tbody>
</table>
# CDC 7155 Disk Controller

## Table 3-1. Subsystem Functions

<table>
<thead>
<tr>
<th>Octal Code</th>
<th>Function</th>
<th>Words Output</th>
<th>Words Input</th>
<th>General Status Required</th>
<th>Octal Code</th>
<th>Function</th>
<th>Words Output</th>
<th>Words Input</th>
<th>General Status Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>0000</td>
<td>Connect ①</td>
<td>1</td>
<td>Yes</td>
<td>Yes</td>
<td>0027</td>
<td>Read checkword gap sector</td>
<td>Yes</td>
<td></td>
<td>Yes</td>
</tr>
<tr>
<td>0001</td>
<td>Seek, 1:1 interface ①</td>
<td>4</td>
<td>Yes</td>
<td></td>
<td>0030</td>
<td>Read factory data</td>
<td>Yes</td>
<td></td>
<td>Yes</td>
</tr>
<tr>
<td>0002</td>
<td>Seek, 2:1 interface ②</td>
<td>4</td>
<td>Yes</td>
<td></td>
<td>0031</td>
<td>Read utility map</td>
<td>Yes</td>
<td></td>
<td>Yes</td>
</tr>
<tr>
<td>0004</td>
<td>Read ⑥</td>
<td>Yes</td>
<td></td>
<td></td>
<td>0032</td>
<td>Block transfer buffer read</td>
<td>Yes</td>
<td></td>
<td>Yes</td>
</tr>
<tr>
<td>0005</td>
<td>Write ⑥</td>
<td>Yes</td>
<td></td>
<td></td>
<td>0033</td>
<td>Block transfer buffer write ①</td>
<td>Yes</td>
<td>322</td>
<td>Yes</td>
</tr>
<tr>
<td>0006</td>
<td>Write verify ⑥</td>
<td>Yes</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Yes</td>
<td></td>
<td>Yes</td>
</tr>
<tr>
<td>0007</td>
<td>Read checkword</td>
<td>Yes</td>
<td></td>
<td></td>
<td>0034</td>
<td>Read protected sector</td>
<td>Yes</td>
<td></td>
<td>Yes</td>
</tr>
<tr>
<td>0010</td>
<td>Operation complete</td>
<td>No</td>
<td></td>
<td></td>
<td>0035</td>
<td>Write last sector</td>
<td>Yes</td>
<td>⑩</td>
<td>Yes</td>
</tr>
<tr>
<td>0011</td>
<td>Disable drive reserve</td>
<td>No</td>
<td></td>
<td></td>
<td>0036</td>
<td>Write verify last sector</td>
<td>Yes</td>
<td>⑩</td>
<td>Yes</td>
</tr>
<tr>
<td>0012</td>
<td>General status</td>
<td>1</td>
<td>No</td>
<td></td>
<td>0037</td>
<td>Write protected sector</td>
<td>Yes</td>
<td></td>
<td>Yes</td>
</tr>
<tr>
<td>0013</td>
<td>Detailed status</td>
<td>12⑥</td>
<td>No</td>
<td></td>
<td></td>
<td></td>
<td>Yes</td>
<td></td>
<td>Yes</td>
</tr>
<tr>
<td>0014</td>
<td>Continue ⑥, ④</td>
<td>Yes</td>
<td></td>
<td></td>
<td>0040</td>
<td>Read short</td>
<td>Yes</td>
<td>319⑤</td>
<td>Yes</td>
</tr>
<tr>
<td>0015</td>
<td>Drop seeks</td>
<td>0041</td>
<td></td>
<td></td>
<td></td>
<td>Select strobe and offset ①</td>
<td>Yes</td>
<td></td>
<td>Yes</td>
</tr>
<tr>
<td>0016</td>
<td>Format pack ①</td>
<td>Yes</td>
<td></td>
<td></td>
<td>0042</td>
<td>Clear connected access</td>
<td>No</td>
<td></td>
<td>No</td>
</tr>
<tr>
<td>0017</td>
<td>Return drive address</td>
<td>3</td>
<td>No</td>
<td></td>
<td>0043</td>
<td>Buffer read</td>
<td>Yes</td>
<td></td>
<td>Yes</td>
</tr>
<tr>
<td>0020</td>
<td>Drive release</td>
<td>3</td>
<td>No</td>
<td></td>
<td>0044</td>
<td>Buffer write</td>
<td>Yes</td>
<td></td>
<td>Yes</td>
</tr>
<tr>
<td>0021</td>
<td>Return cylinder address</td>
<td>1</td>
<td>No</td>
<td></td>
<td>0045</td>
<td>Write buffer to disk</td>
<td>Yes</td>
<td></td>
<td>Yes</td>
</tr>
<tr>
<td>0022</td>
<td>Set/clear flaw ①</td>
<td>Yes</td>
<td></td>
<td></td>
<td>0047</td>
<td>Scan cylinder addresses</td>
<td>Yes</td>
<td></td>
<td>Yes</td>
</tr>
<tr>
<td>0023</td>
<td>Extended detailed status</td>
<td>20⑩</td>
<td>No</td>
<td></td>
<td>0048</td>
<td></td>
<td>Yes</td>
<td></td>
<td>Yes</td>
</tr>
<tr>
<td>0024</td>
<td>Read gap sector</td>
<td>Yes</td>
<td></td>
<td></td>
<td>0050</td>
<td>Output on processor channel ①</td>
<td>Yes</td>
<td></td>
<td>Yes</td>
</tr>
<tr>
<td>0025</td>
<td>Write gap sector</td>
<td>Yes</td>
<td></td>
<td></td>
<td>0051</td>
<td>Execute control word sequence ①</td>
<td>Yes</td>
<td></td>
<td>Yes</td>
</tr>
<tr>
<td>0026</td>
<td>Write verify gap sector</td>
<td>Yes</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Yes</td>
<td></td>
<td>Yes</td>
</tr>
</tbody>
</table>
Emulator Hardware

Cyber 170 Channel

Channel Interface

Daughter Card

12

EIDE Interface

16

Bus Extension

SDRAM

TI DSP

TMS320C6711

FEPROM

Development Station (PC)

TI Development Board (cheap complete SBC + development toolkit)
Emulator Hardware
switch (function_code)
{
   case 00001:
      if (getpar(4)) break;
      if (param[0] & 01000) error (00001, 9); /* SEEK 1:1 */
      if ((param[0] & 00040)) error (00001, 5); /* no large sector */
      reserved[drive=(param[0] & 017)] |= 1; /* no 844 drive */
      reserved[drive] &= ~4; /* reserve drive */
      reserved[drive] |= 2; /* clear 2:1 */
      interlace=1; /* set 1:1 */
      cylinder[drive]=param[1];
      track[drive]=param[2];
      sector[drive]=param[3];
      if (x=seek_sector(drive)) error(DISK_ERR, x);
      /* store position */
      break;
   case 00004:
      if (sector[drive] > 31) error (function_code, 1); /* READ */
      if (track[drive] > 39) error (function_code, 2);
      if (cylinder[drive] > 842) error (function_code, 3);
      if (x=read_sector(buffer)) error(DISK_ERR, x);
      /* from current drive */
      putpar(buffer, 322);
      if ((sector[drive] += interlace) > 31)
         { sector[drive] = 32; ++track[drive]; }
      break;
   case 00005:
      if (getpar(322)) break; /* WRITE */
      if (sector[drive] > 31) error (function_code, 1);
      if (track[drive] > 39) error (function_code, 2);
      if (cylinder[drive] > 842) error (function_code, 3);
      if (x=write_sector(param)) error(DISK_ERR, x);
      /* to current drive */
      if ((sector[drive] += interlace) > 31)
         { sector[drive] = 32; ++track[drive]; }
      break;
Channel Receiver Detail

Adjustable Threshold

RS422 receiver
(SN75ALS195)

Signal in

TTL out

R=Z

Termination + High Pass

Diagram showing signal levels and circuit components.
Channel Transmitter Detail

Use analog switches

$t(on) = 5$ nsec
$t(off) = 2.5$ nsec

$\text{clock}=25$ nsec $\implies 22.5$ nsec pulses

Signal out

Terminated into 75 Ohms
Digital PLL for Clock

- Channel clock = 10 MHz ~ 100 nsec
- Channel pulse = 20 ... 25 nsec
- multiply channel clock by 4, using digital Phase-Locked Loop (PLL)
- have channel synchronous 40 MHz clock ~ 25 nsec ==> can make channel pulses