

# New Disks for Old Heavy Iron

Dr. John G. Zabolitzky

[www.cray-cyber.org](http://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

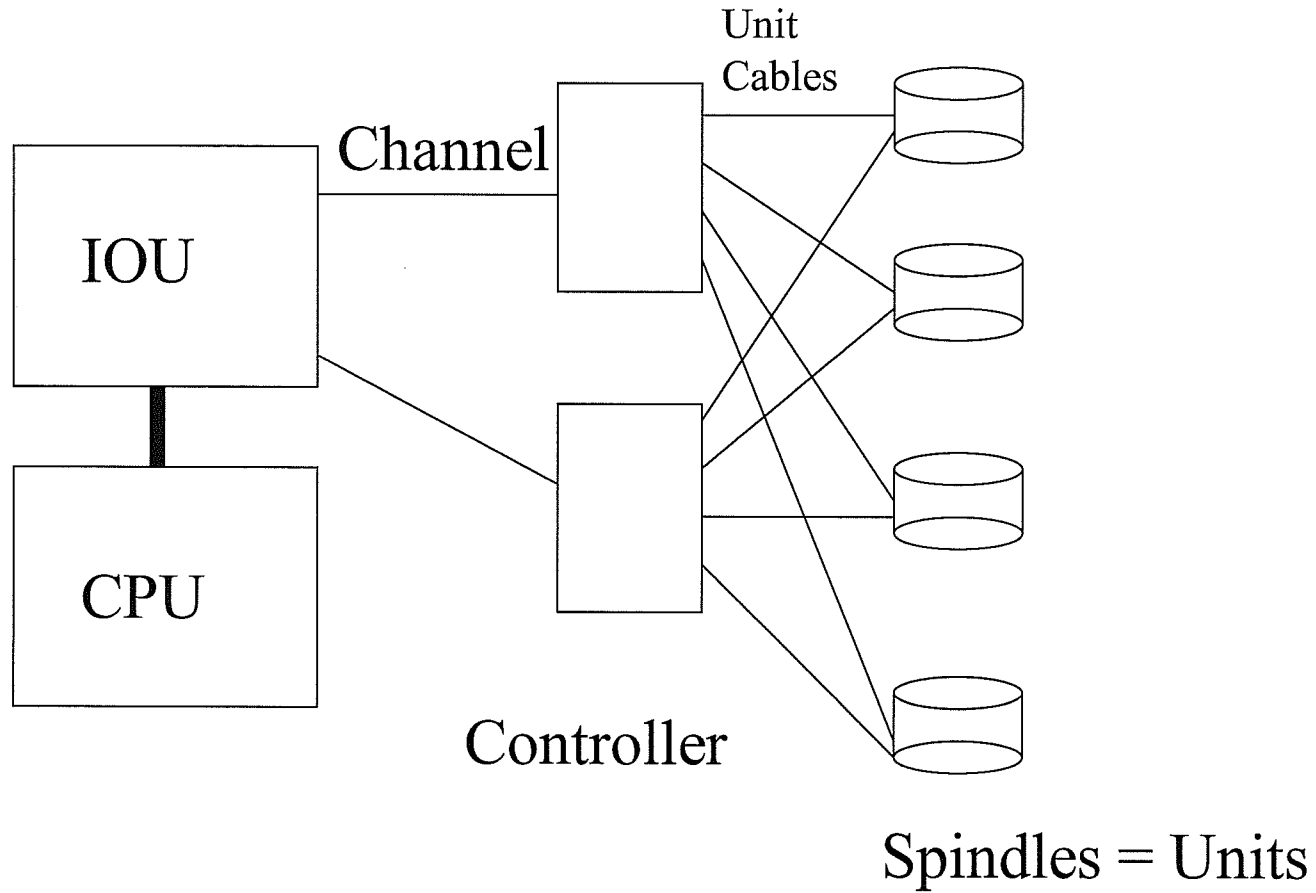
# RMS parameters

- 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

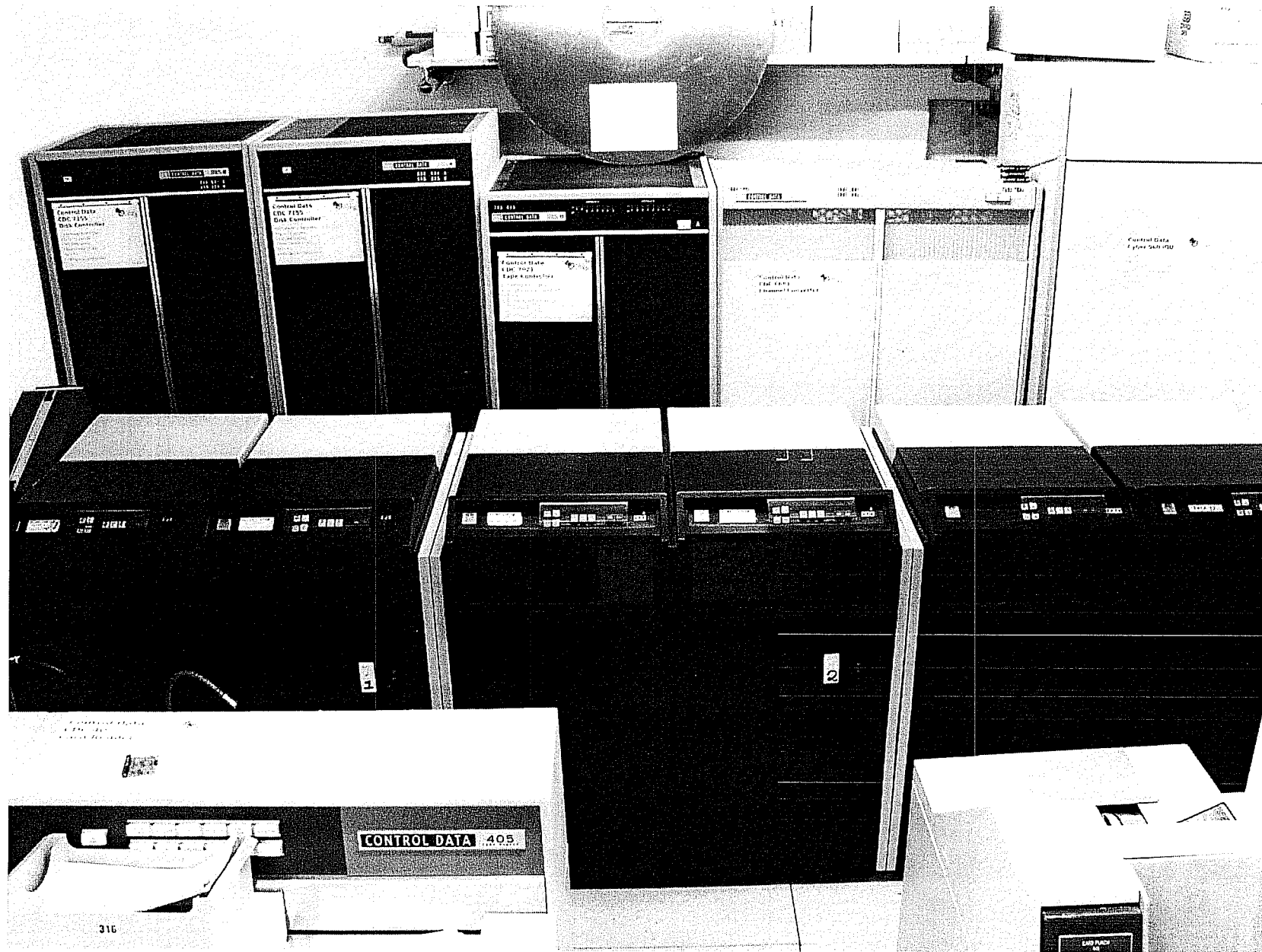
# RMS topology

- 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



# A typical Installation



CONTROLLER

DRIVES

# 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 ( $20^{\circ} * 14'' * 10^{-5} = 70\mu$ )

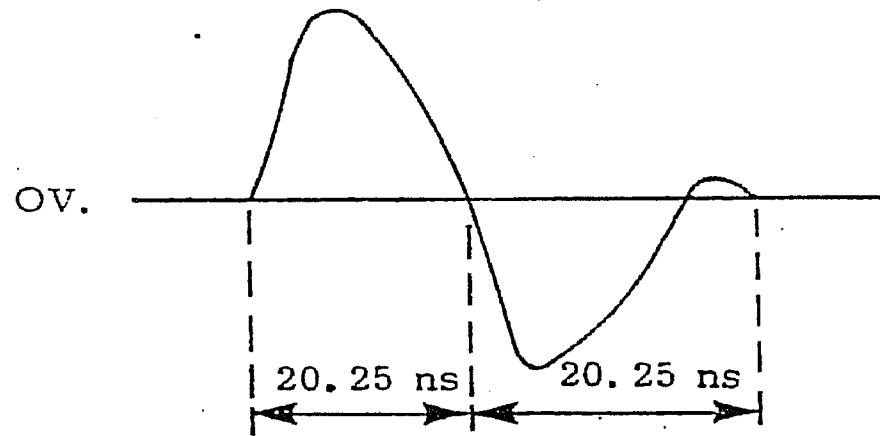
# Disk Emulation

- 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



MAX VOLTAGE SWING =  $\pm 2.7V$  p-p

MIN VOLTAGE SWING =  $\pm 2.1V$  p-p

Figure 1-2. Output Pulse Characteristics

Measured at output pin of TR. looking into a  $75\Omega$  impedance.

# Cyber 170 Channel

TABLE 1-1. DATA CHANNEL COAXIAL CABLE LINES

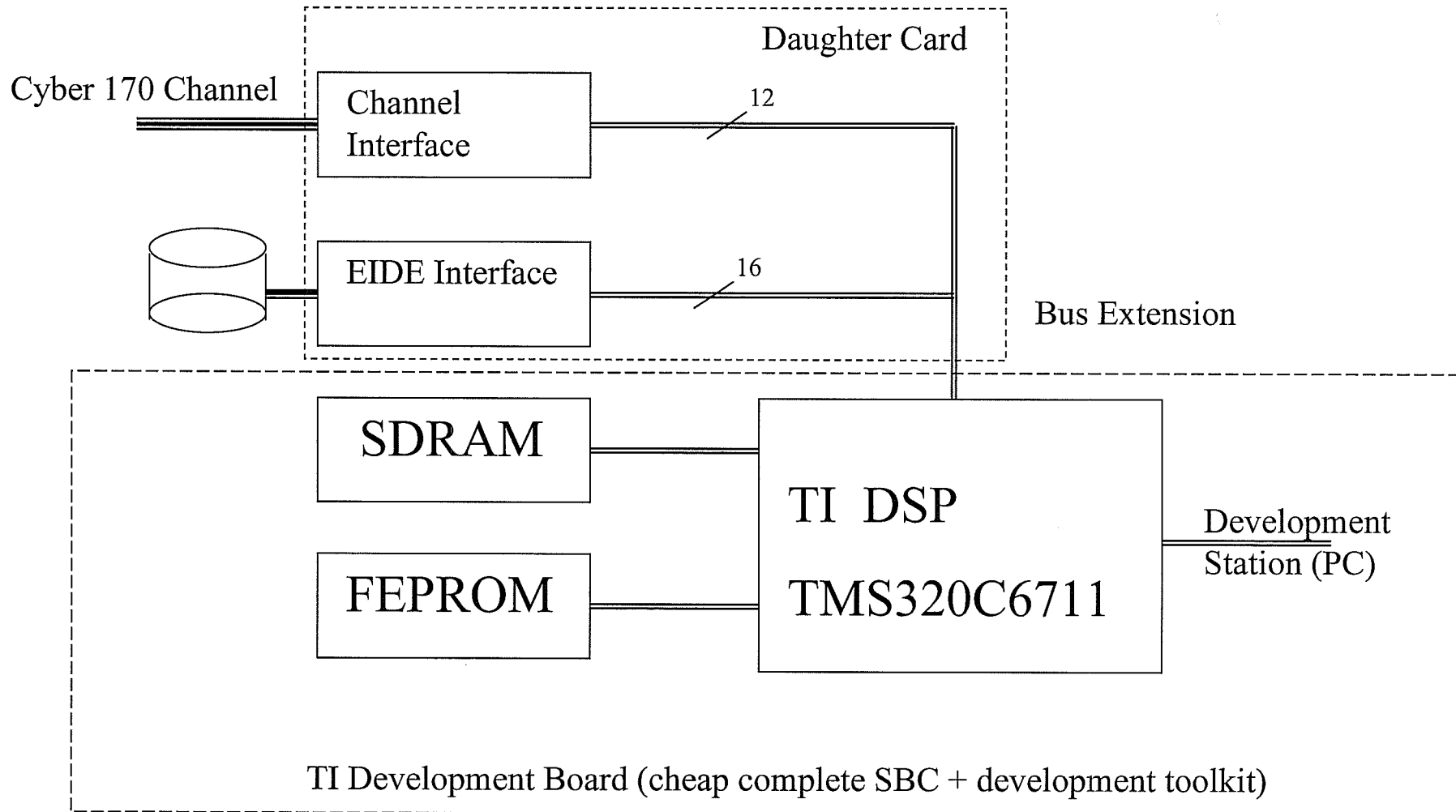
Input Cable	PIN	Color Code	Output Cable
Data bit 0	A	90	Data bit 0
Data bit 1	B	91	Data bit 1
Data bit 2	C	92	Data bit 2
Data bit 3	D	93	Data bit 3
Data bit 4	E	94	Data bit 4
Data bit 5	F	95	Data bit 5
Data bit 6	H	96	Data bit 6
Data bit 7	J	97	Data bit 7
Data bit 8	K	98	Data bit 8
Data bit 9	L	99	Data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

# CDC 7155 Disk Controller

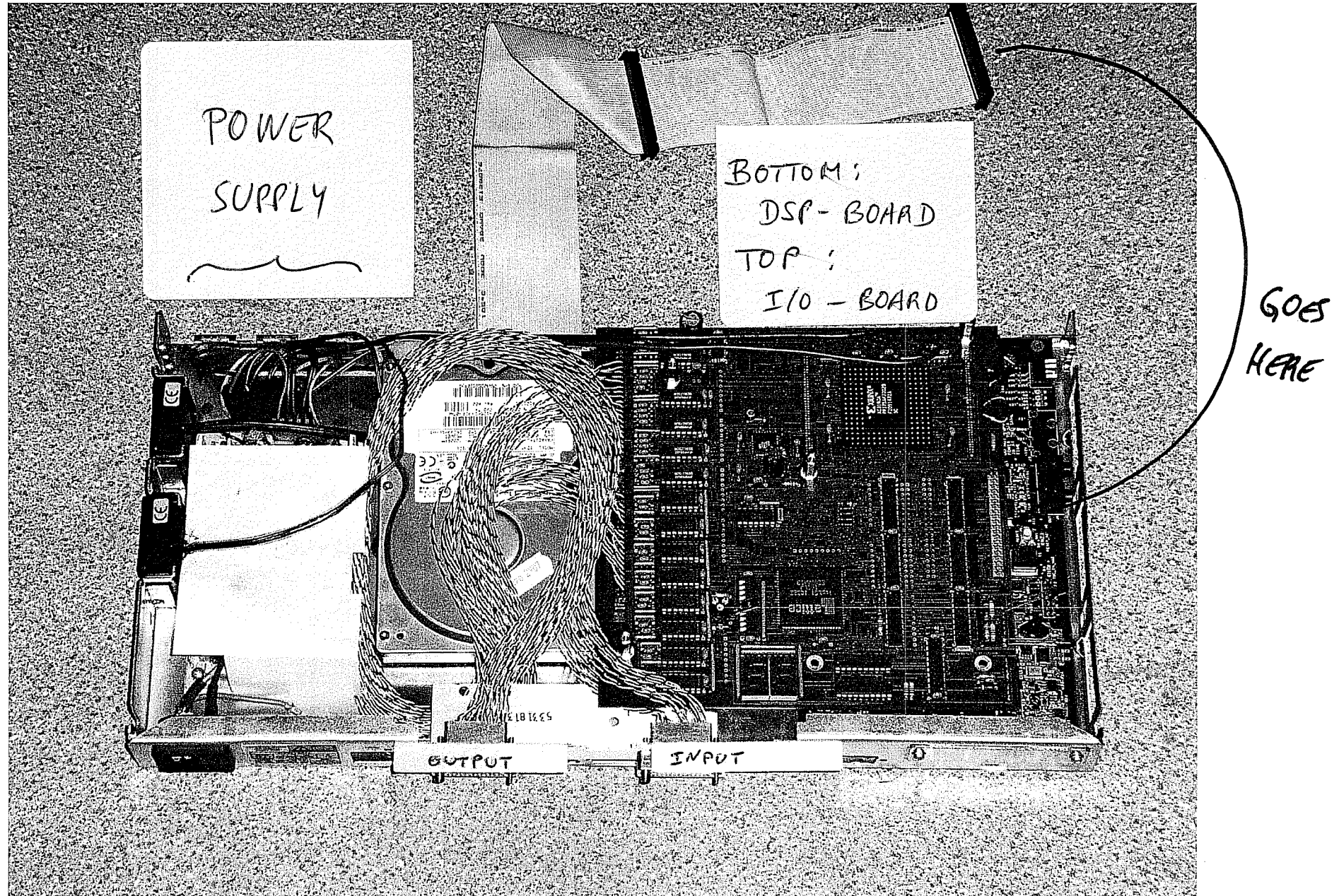
TABLE 3-1. SUBSYSTEM FUNCTIONS

Octal Code	Function	Words Output	Words Input	General Status Required	Octal Code	Function	Words Output	Words Input	General Status Required
✓ 0000	Connect ①	1		Yes	✓ 0027	Read checkword gap sector			Yes
✓ 0001	Seek, 1:1 interlace ①	4		Yes	✓ 0030	Read factory data		322	Yes
✓ 0002	Seek, 2:1 interlace ①	4		Yes	✓ 0031	Read utility map		322	Yes
✓ 0004	Read		⑥	Yes	✓ 0032	Block transfer buffer read		322	No
✓ 0005	Write	⑥		Yes ②	✓ 0033	Block transfer buffer write ①	322		Yes
✓ 0006	Write verify	⑥		Yes ②	✓ 0034	Read protected sector		322	Yes
✓ 0007	Read checkword			Yes	✓ 0035	Write last sector		⑥	Yes
✓ 0010	Operation complete			No	✓ 0036	Write verify last sector		⑥	Yes
✓ 0011	Disable drive reserve			No	✓ 0037	Write protected sector	322		Yes
✓ 0012	General status		1	No	✓ 0040	Read short		319 ⑤	Yes
✓ 0013	Detailed status		12 <sub>10</sub>	No	✓ 0041	Select strobe and offset ①		1	Yes
✓ 0014	Continue	⑥ ③	⑥ ④	Yes	✓ 0042	Clear connected access			No
✓ 0015	Drop seeks			No	✓ 0043	Buffer read		322	Yes
✓ 0016	Format pack ①	7		Yes	✓ 0044	Buffer write	322		Yes
✓ 0017	Return drive address		3	No	✓ 0046	Write buffer to disk			Yes
✓ 0020	Drive release			No	✓ 0047	Scan cylinder addresses			Yes
✓ 0021	Return cylinder address		1	No	✓ 0050	Output on processor channel ①		3	Yes
✓ 0022	Set/clear flaw ①	1		Yes	✓ 0051	Execute control word sequence ①		49	Yes
✓ 0023	Extended detailed status		20 <sub>10</sub>	No					
✓ 0024	Read gap sector		⑥	Yes					
✓ 0025	Write gap sector	⑥		Yes ②					
✓ 0026	Write verify gap sector	⑥		Yes ②					

# Emulator Hardware



# Emulator Hardware



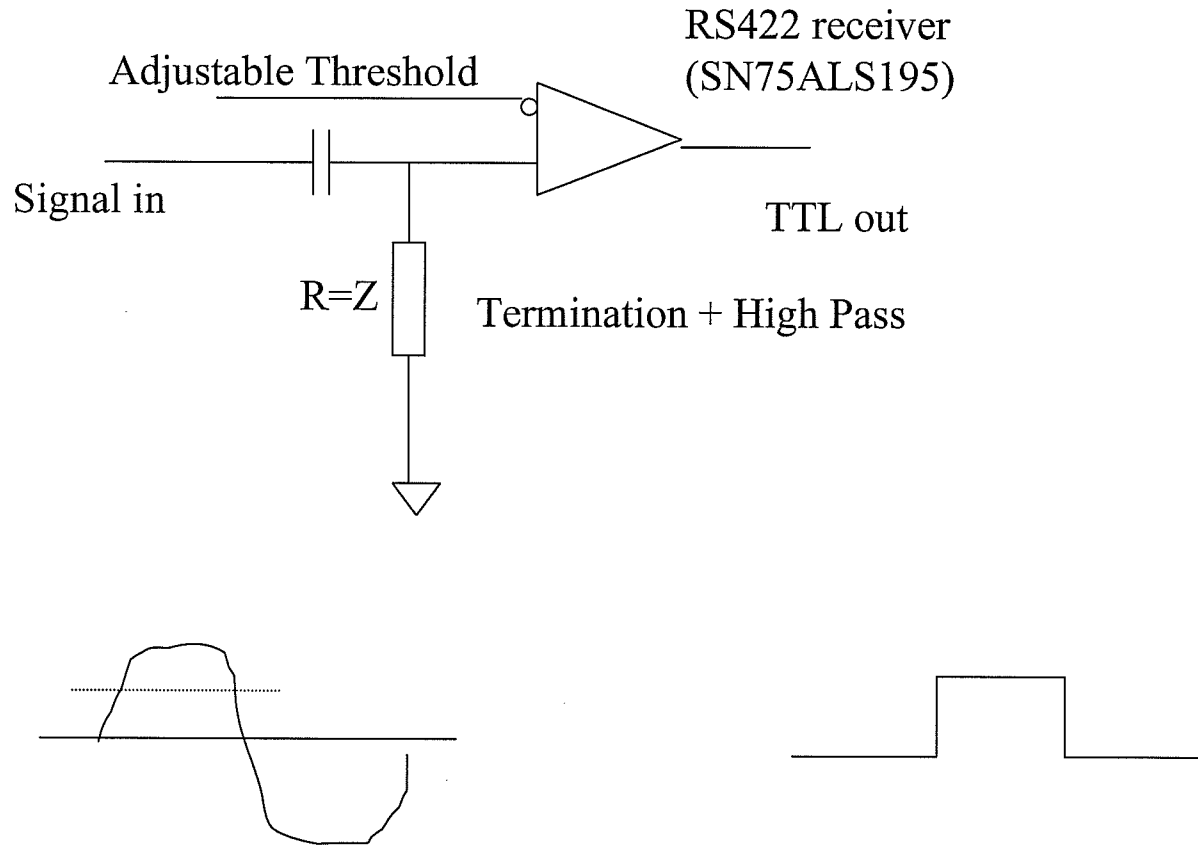
# Typical Emulation Code

```
switch (function_code)
{case 00001:    if (getpar(4)) break;                                /* SEEK 1:1 */
               if ( param[0] & 01000 ) error (00001, 9);          /* no large sector */
               if (!(param[0] & 00040)) error (00001, 5);          /* no 844 drive */
               reserved[ drive=(param[0] & 017) ] |= 1;           /* reserve drive */
               reserved[drive] &= ~4;                               /* clear 2:1 */
               reserved[drive] |= 2;                               /* set 1:1 */
               interlace=1;
               cylinder[drive]=param[1];                          /* store position */
               track[drive]=param[2];
               sector[drive]=param[3];
               if (x=seek_sector(drive)) error(DISK_ERR, x);      /* let disk work */
               break;

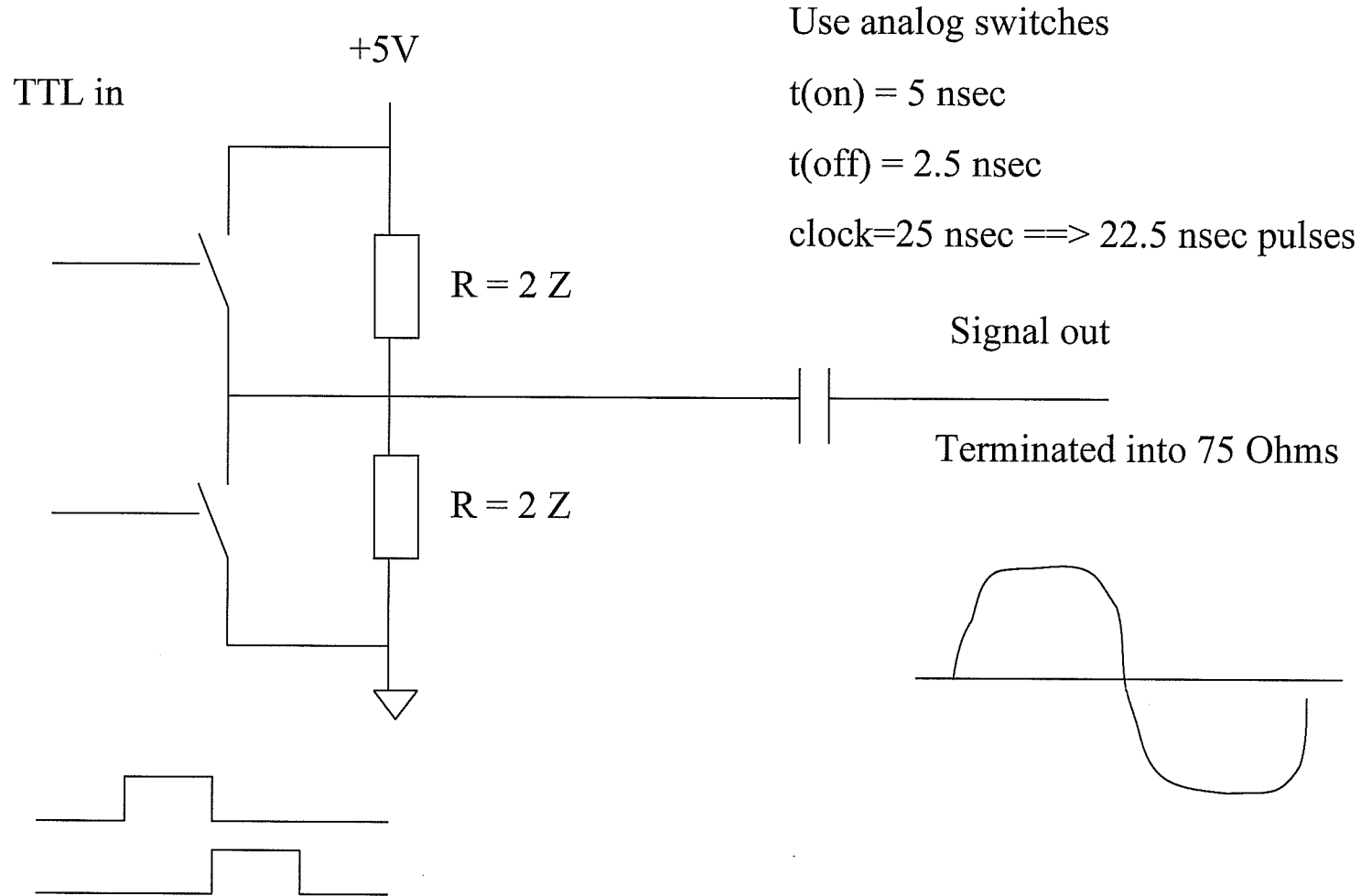
case 00004:                                          /* READ */
               if (sector[drive] > 31) error (function_code, 1);
               if (track[drive] > 39) error (function_code, 2);
               if (cylinder[drive] > 842) error (function_code, 3);
               if (x=read_sector(buffer)) 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



# Channel Transmitter Detail



# Digital PLL for Clock

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