ERIC J. BOWDEN
MATT E. RICKS
IRENE K. THOMPSON

PROPOSAL

COMPUTER ENGINEERING FINAL PROJECT: DIGITAL VIDEO RECORDER

REVISION 5.0E

REVISED ON APRIL 28, 2006

TABLE OF CONTENTS

LIST OF FIGURES AND TABLES

INTRODUCTION AND MOTIVATION

In today's world using a tape as a storage device has become obsolete. VCRs have made way for TIVO. For our project we would like to build a Digital Video Recorder. This device would record television programs onto a hard disk drive and allow the user to watch the program at their convenience. Specifically, our device would allow the user to pause live TV. With this feature, you would never miss part of your favorite show again due to interruptions. In addition to pausing live television the user would be able to fast-forward and rewind within a 15 minute timeframe.

We believe this project would be a great way to combine both hardware and software components. It would allow us to explore analog signal processing, analog-to-digital conversion and digital storage and recall of encoded data in a real-time environment.

FUNCTIONAL SPECIFICATION

BASELINE FUNCTIONALITY SPECIFICATION

The device to be designed and implemented is a Digital Video Recorder (DVR) which will serve as video capture unit for live television signals. There are three necessary features that must be operational by the completion of the baseline implementation. The DVR must support:

- Receiving a standard, analog NTSC cable television signal and converting it into a digital audio and video stream.
- Storing 15 minutes of digitally converted data to some storage medium for future recall.
- Convert the 15 minutes of digital data into an NTSC analog television signal on demand using standard user video manipulation techniques such as fast-forward, rewind, pause and play.

BASELINE IMPLEMENTATION SPECIFICATION

The DVR functionality will be implemented as a PCI add-on card for consumer PCs. This naturally creates two processing domains (on-card versus on-PC) with three critical interfaces and several sub-interfaces. The first interface is a user-accessible hardware interface for inputting and outputting the AV data. The second interface consists of the PCI communication bus between the host process on the PC and the microcontroller on the board. The final interface is a software implemented UI on the host PC that provides control over the DVR.

EXTENDED FUNCTIONALITY SPECIFICATIONS

If time permits, we would like to expand on the aforementioned features of our recorder. The following are extensions of the *Baseline Functionality Specification* that represent desired, but not critical, enhancements to functionality.

- Commercial detection and eradication.
- Increased storage capacity, preferably up to an hour's worth of live television.
- Embedded interactive television menus for DVR control
- Have the ability to store entire broadcast segments for recall in addition to live television.
- Have the ability to instruct the DVR to record a television program on a given channel, at a given time and duration.

NOTES ABOUT COMMERCIAL DETECTION

Commercial detection may be achieved via two potential routes. First, commercials are separated from desired program by a few frames of black, thus the processor would need to detect these frames in order to know when to start and stop recording. Some heuristics may need to be done in order to account for noise in the black frames. An alternate method would be to use the first 21 lines of the broadcast (which is not visible to consumers) to distinguish commercials from prime-time broadcasts.

PROJECT SCHEDULING

MILESTONES

There are four major milestones (with one optional milestone) for this project.

MILESTONE 1: AUDIO/ VIDEO SIGNAL DECOMPOSITION AND ENCODING/DECODING

Signal capturing should be done first. Without a digitally encoded AV signal there would be no way to store the data for future recall. Similarly, a digitally encoded signal is only useful if it can be decoded into a faithful reproduction of the original analog signal. During this phase the encoded signal would be immediately decoded through the designed hardware, and sent to a television set to ensure there are no hang-ups in the encoding/decoding stages.

MILESTONE 2: STORING DATA

Once the analog-to-digital encoding and decoding scheme has been defined the initial digital processing software may be designed. The goal for this milestone is the shipment and storage of the digitally encoded A/V data. Reaching this point represents the largest development hurdle as it involves integrating the onboard microcontroller with the host

application via the PCI-kernel interface. Once this milestone is reached the system should be able to send an arbitrary amount of A/V to the host for management and storage through the host application. The success indicator will be the ability to play the stored MPEG2 compressed data on Windows Media Player.

MILESTONE 3: DATA PLAYBACK

After achieving the ability to store data, the system should be able to recall data for playback on a television set. This milestone represents the reverse of milestone 2's goal. The main difference is that the kernel module is in charge of bus-mastering (as opposed to the DVR PCI card). The success criterion for this milestone is the ability to send pre-stored MPEG2 data (e.g., data stored by milestone 2) back to the PCI card for analog encoding and playback on a TV.

MILESTONE 4: SIMULTANEOUS SEND AND RECEIVE (FEATURE COMPLETE)

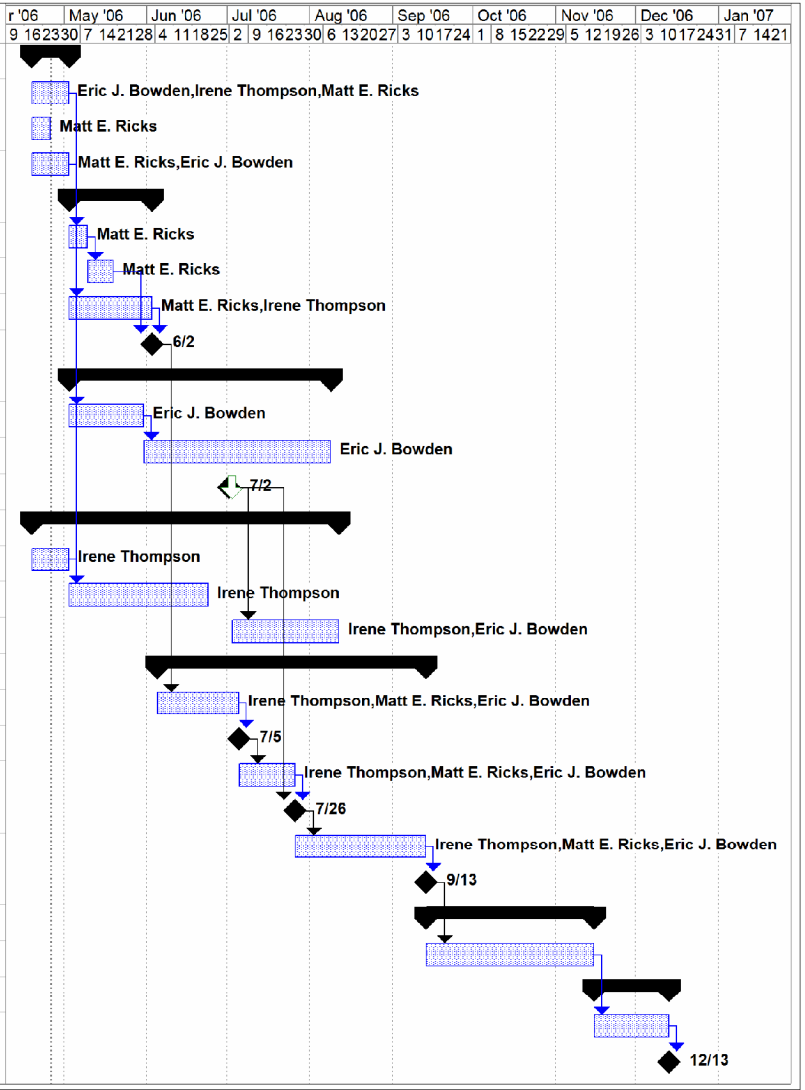
Once dedicated data streaming can occur in either direction, the system should be able to perform data storage and recall simultaneously. As a DVR, this represents the ability to time-delay live television (for example, a 15-minute pause). This milestone corresponds with the introduction of a transaction scheduler for the PCI card and host PC. The user interface should permit pausing, playing, fast-forwarding and rewinding, in addition to channel changing (if we are attached to a TV tuner instead of a cable box). At this point the system would be baseline feature complete.

MILESTONE 5: IMPLEMENT EXTENDED FEATURES

Once the baseline features have been hammered out the extended features (like commercial detection, scheduling program recording, and embedded menus) may be implemented.

TIMELINE

ID	Task Name	Start	Finish
1	**Scoping Phase**	**Wed 4/19/06**	**Tue 5/2/06**
2	Finalize Project Specification	Wed 4/19/06	Tue 5/2/06
3	Finalize Component Selection	Wed 4/19/06	Tue 4/25/06
4	Order Materials for First Sprint	Wed 4/19/06	Tue 5/2/06
5	**Hardware Design**	**Wed 5/3/06**	**Fri 6/2/06**
6	Layout and Design PCI Board	Wed 5/3/06	Tue 5/9/06
7	Fabricate PCI Board	Wed 5/10/06	Fri 5/19/06
8	Prepare board verification code	Wed 5/3/06	Fri 6/2/06
9	Milestone 1: Immediate encode and decode	Fri 6/2/06	Fri 6/2/06
10	**Kernel Software Design**	**Wed 5/3/06**	**Tue 8/8/06**
11	Research PCI Interfacing with OSR Online Training Kit	Wed 5/3/06	Tue 5/30/06
12	Kernel Module Development	Wed 5/31/06	Tue 8/8/06
13	Finalize kernel-host API	Sun 7/2/06	Sun 7/2/06
14	**Host PC Software User Interface**	**Wed 4/19/06**	**Fri 8/11/06**
15	Research Java Media Framework	Wed 4/19/06	Tue 5/2/06
16	Implement Host-side GUI	Wed 5/3/06	Fri 6/23/06
17	Implement Host-side I/O	Mon 7/3/06	Fri 8/11/06
18	**Microcontroller Software**	**Mon 6/5/06**	**Wed 9/13/06**
19	Develop M2 (storage stub) code	Mon 6/5/06	Wed 7/5/06
20	Milestone 2: Send data from card to PC for storage	Wed 7/5/06	Wed 7/5/06
21	Develop M3 (receptor) code	Wed 7/5/06	Wed 7/26/06
22	Milestone 3: Receive pre-stored data from PC to display on T	Wed 7/26/06	Wed 7/26/06
23	Develop M4 (simultaneous scheduler) code	Wed 7/26/06	Wed 9/13/06
24	Milestone 4: Simultaneous send and receive (Feature compl	Wed 9/13/06	Wed 9/13/06
25	**Debug time**	**Wed 9/13/06**	**Wed 11/15/06**
26	If not feature complete, use this time to address issues	Wed 9/13/06	Wed 11/15/06
27	**Extended functionality**	**Wed 11/15/06**	**Wed 12/13/06**
28	Develop extensions	Wed 11/15/06	Wed 12/13/06
29	Milestone 5: Extended functionality features	Wed 12/13/06	Wed 12/13/06



Project: SeniorProject_GANT.mpp
Date: Wed 4/26/06

Task | Split | Progress | Milestone | Summary | Project Summary | External Tasks | External Milestone | Deadline

Page 1

Figure 1 - Project Gantt chart

TASKING GUIDELINES

Matt Ricks will lead the analog-to-digital encoding and decoding, hardware development.

Irene Thompson will lead the signal subsystem processing and onboard software development.

Eric Bowden will head up the analog and digital integration and the software development for the PCI interface on the host system.

As may be seen in the Gantt chart, the resource lead also represents the component lead for that particular aspect of development.

RISK MANAGEMENT

POTENTIAL RISKS

1) The necessity of writing a kernel module (device driver) for Windows as a PC is the primary user interface. The difficulty level of writing the driver may be much greater than anticipated. We are investigating the purchase of PCI training kits (see osronline.com 9054 DMA Learning Kit) to help mitigate this risk. If the kernel module proves too time consuming we will opt to have a hard drive directly hooked up to the PCI card for storage purposes. The PCI card would still be connected, if only for power and clock supplies.

2) Choosing fast enough hardware such that we can process the data without any notice of signal loss to the user. The signal the user sees may need to be delayed by a few seconds in order to do the necessary processing. According to Al Davis, if we get in a bind we may drop a few video frames without a noticeable deterioration in picture quality.

3) If we implement commercial detection – there are risks of not being able to adequately characterize commercial broadcasts (e.g., noise in black frames, encrypted scan lines, etc.)

4) Cost; we can't afford to make mistakes as far as PCI layout and manufacturing are concerned. Optimally, the device will fit on a single PCI card (utilizing both sides) – we're trying to avoid a multi-layered board.

BILL OF MATERIALS

Part Description	*Manufacturer*	*Vendors*	*Cost*
4"x6" 2 Layer PCB	PCB123.com		$53
PIC16F870 Microcontroller	Microchip	Digikey, Mouser	$3
PCI 9054-AC50PI F	PLX Technology	SemicondutorStore.com	$32
SAA6752HS MPEG Encoder	Philips	Digikey	$30
ADV7180 Video Decoder	Analog devices		$5
ADV7170 Video Encoder	Analog devices		$5
		Total	$128

Table 1 - Bill of materials

Presently all parts are immediately available with no noted lead times.

CONTACT INFORMATION

PCB123
13626 S Freeman Rd
Mulino Oregon 97042 USA
Phone: (503) 829-9108
(800) 228-8198
Fax: (503) 829-6657
http://www.pcb123.com

Analog Devices, Inc
Phone: (800) 262-5643
http://www.analog.com

Microchip
Corporate Headquarters
Microchip Technology Inc.
2355 West Chandler Blvd.
Chandler, Arizona, USA 85224-6199
Phone: (480) 792-7200
http://www.microchip.com

SemiconductorStore.com
Richard Judy, Customer Support
Phone: (877) 466-9722
http://www.semiconductorestore.com
http://www.plxtech.com

Digikey and Mouser are general electronics suppliers, and are not presently supplying us with a rare or difficult-to-find part.

IMPLEMENTATION DETAILS

HARDWARE DESIGN AND INTERFACE

The first stage of AV processing occurs on the PCI card, where a user plugs in a coaxial cable carrying an NTSC television signal. This signal may come from a cable box or a terrestrial antenna. If the input is supplied by antenna, the on-board microcontroller will feed the signal through an analog tuner chip to isolate the desired channel. From this point, the NTSC signal proceeds to the encoding and compressing phase.

NTSC video signals arrive as a continuous stream of horizontal lines for each video frame. Each TV frame has 525 horizontal lines. The incoming signal is digitized into 367,000 pixels using an ADV7180 Video Decoder from Analog Devices. The pixels are subsequently fed into a SAA6752HS MPEG-2 encoder from Philips. The SAA6752HS does the encoding in a single chip. The MPEG packets are then sent to the microcontroller, which buffers a frame's worth of data and then ships the frame to the host PC via the PCI bus. The reverse encoding process is similar. The signal returned from the PC host application is decompressed with an MPEG decoder chip and converted into a video signal using the ADV7170 Video Encoder. The video signal is then streamed through a channel 3/4 modulator for TV viewing.

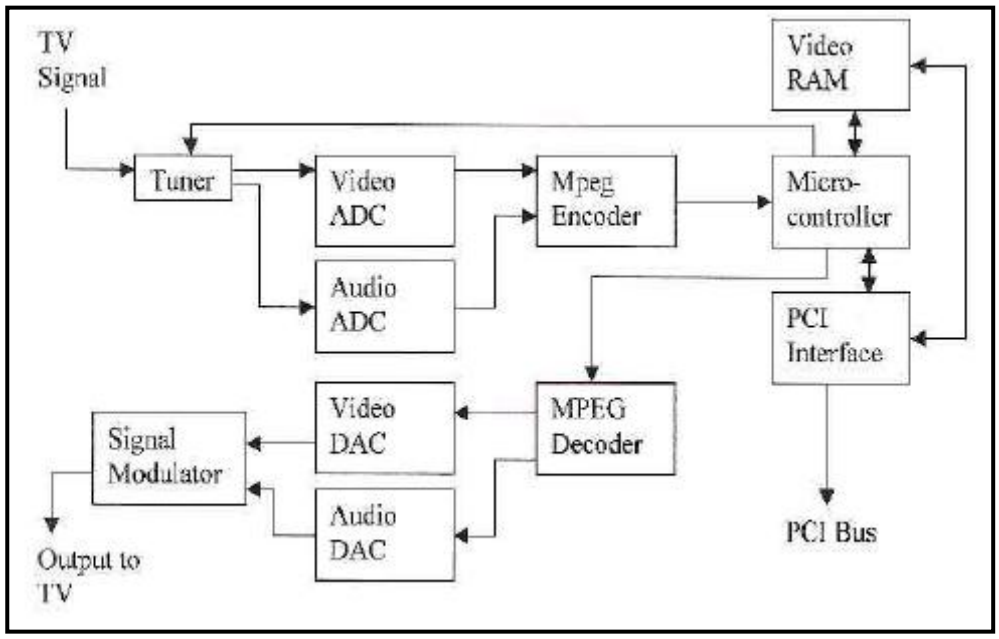


Figure 2 - Hardware block diagram

PCI HARDWARE INTERFACE

Once a single frame's worth of AV data has been encoded and buffered it must be sent over the PCI bus to the host for assembly and processing. This represents a significant challenge and risk as no one on the team has had experience programming for a PCI bus. Thus we have elected to use a PCI interface chip to reduce the problem to one of software rather than hardware. The chip in question is the PLX PCI 9054-AC50PI F, a PCI I/O accelerator chip provided by PLX Technology.

This chip provides several key interfaces that facilitate communication (on the hardware timing and signaling level) between the microcontroller and the PC host. First, it provides 32-bit Direct Master FIFOs to the microcontroller that permit data transmission as a bus master – thus instead of waiting for the PC host software to query for updated data, the PCI card will be pro-active in pushing data out for storage. Once again, this interface chip takes care of the PCI transmission mechanics so that the problem becomes one of software.

Secondly, the PCI 9054 provides two mission-critical direct memory access (DMA) engines that permit the host PC software to operate completely oblivious to the PCI card. As far as the host software is concerned, it is just writing to another memory location, not much different from regular memory (albeit, a little slower). Under the hood, however, this data gets pushed through the PCI interface directly to a VRAM module (alternatively, a dual-ported SDRAM module) that the onboard microcontroller drives continuously for data being streamed to the DAC and mixer for display on the television set.

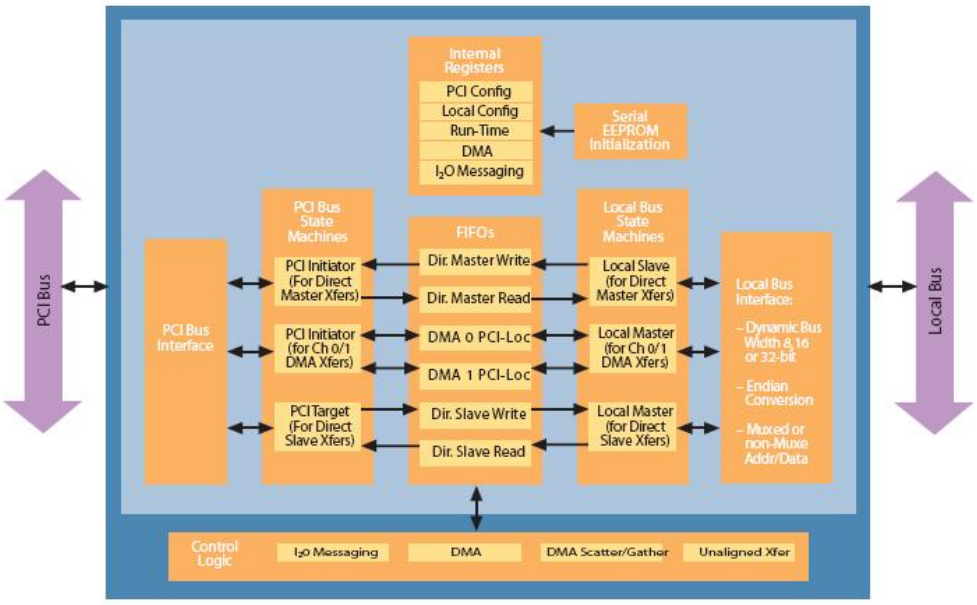


Figure 3 - PCI 9054 internal block diagram; from PLX marketing material

PCI SOFTWARE INTERFACE

The interface to the PCI 9054 from the host computer will be achieved by creating a device driver (kernel module) for Windows XP. This driver will control access to the PCI card for the host application. Specifically, the driver will instantiate interrupt service routines and handlers with the operating system that will permit both the PCI interface chip and the host application to communicate with each other via interrupts.

Developing the PCI interface driver represents the single largest risk for this project, as experience in the matter is minimal. Various channels are being investigated at this time to mitigate this risk, including the purchase of PLX Technology's Rapid Deployment Kit for the PCI 9054 ($499) and/or the purchase of osronline.com's 9054 DMA Learning Kit ($199). If the kernel driver becomes untenable then "Plan B" calls for connecting a SATA hard drive directly to the PCI card. This would necessitate writing a hard disk device driver, but would keep everything locally on the PCI card. The PCI card would still be plugged into the host PCI, if only for power and clock supplies.

SOFTWARE DESIGN AND INTERFACES

The software module has two functions. First, it needs to provide the user with an interface to control the television broadcast. Second, the module will deal with data flow control. Implementation will be done using Java.

USER INTERFACE

The user will control the DVR through a GUI interface. This interface will include an MPEG player which displays the frames as they are being sent back to the PCI card. This video player will be built using the Java Media Framework (JMF). This is a Java API that enables audio and video to be added to Java applications. This package makes it possible to playback several media formats, including MPEG.

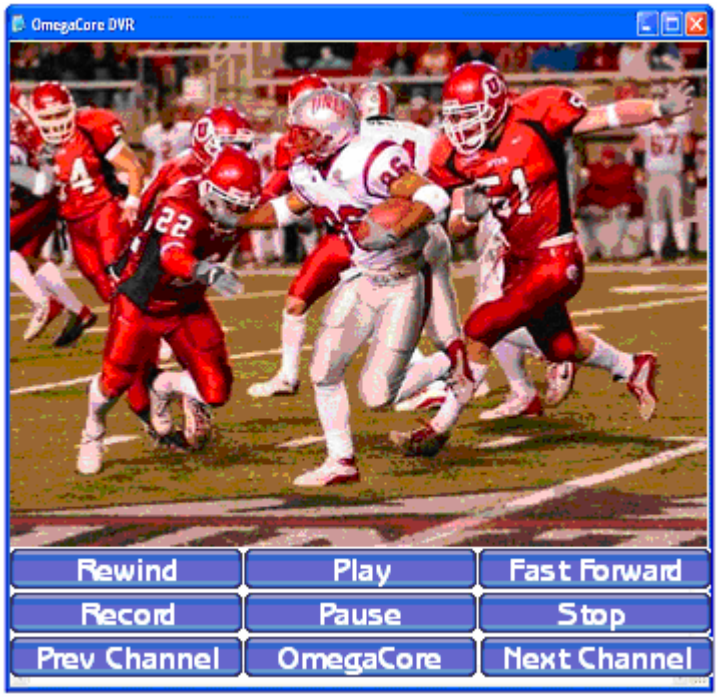


Figure 4- Initial GUI Concept

DATA FLOW CONTROL

The software application receives frames from the PCI card through a kernel module. It is responsible to store each frame on the hard drive. This module is also responsible for extracting frames from the hard drive and sending them to the PCI card. The PCI card in turn will send these frames on to the television set. The implementation strategy views the hard drive as a circular memory device. The program will keep track of two pointers, which point at memory locations on the hard drive. One pointer keeps track of where the next frame that is received from the PCI card will be stored (broadcast pointer). A second pointer keeps track of the location of the next frame to be sent to the PCI card (user pointer). Different actions requested by the user will result in different reactions by the program.

Play: The broadcast pointer and user pointer move up.

Pause: The broadcast pointer keeps moving up since new data continuously needs to get stored. The user pointer does not move and keeps pointing at the same S frame. This way live television will get stored, but the program the user sees will get paused.

Rewind: The broadcast pointer moves up. The user pointer moves backwards along the S frames. It skips over several frames at once.

Fast Forward: The broadcast pointer moves up. The user pointer moves up in larger steps, pointing at the S frames only.

Previous/Next Channel: The module will send a message through the kernel to the microcontroller that sets the frequency of the tuner on the PCI board.

MICROCONTROLLER SOFTWARE

The microcontroller facilitates the communication between several parts on the PCI card. First, it writes the data it receives from the MPEG Encoder to VRAM. This data eventually is transferred from the VRAM to the PCI bus and stored on the hard drive. Second, the microcontroller receives data from the PCI interface through a VRAM module. This data is sent to the MPEG Decoder to be displayed on the television set.

Finally, the microcontroller is responsible for selecting which frequency the tuner selects for viewing. The microcontroller receives the desired frequency from the Software module via the PCI interface.

REFERENCES

Abbott, Doug. 2004. PCI Bus Demystified, Second Edition. Newnes.

Ball, Stuart R. 2004. Analog Interfacing to Embedded Microprocessors, Second Edition. Newnes.

Kovsky, Steve. High-Tech Toys for Your TV, Secrets of TiVo, Xbox, ReplayTV, UltimateTV and More. Que.

Watkinson, John. 2004. The Mpeg Handbook. Focal Press.

Whitaker, Jerry. 2001. Television Receivers. McGraw-Hill.

Whitaker, Jerry. 2003. Video and Television Engineering, Fourth Edition. McGraw-Hill.