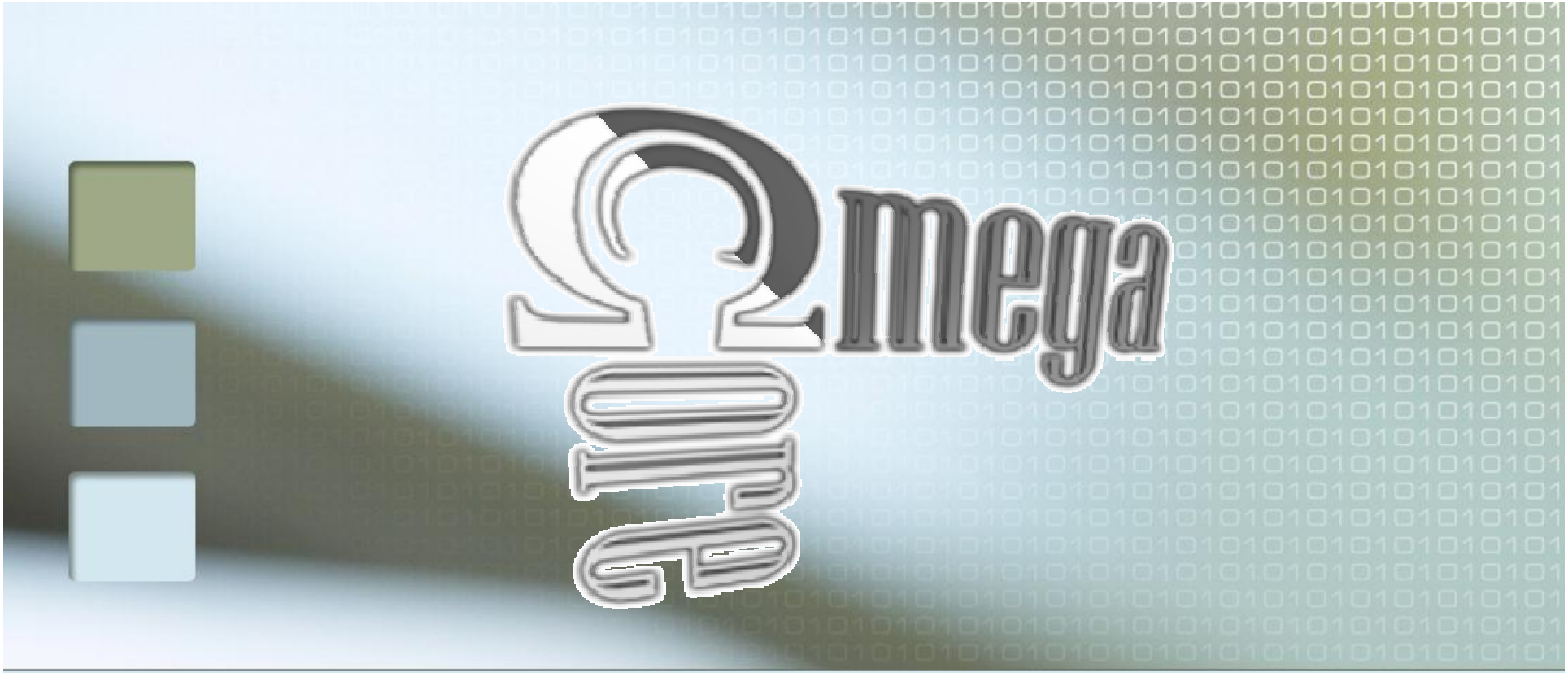
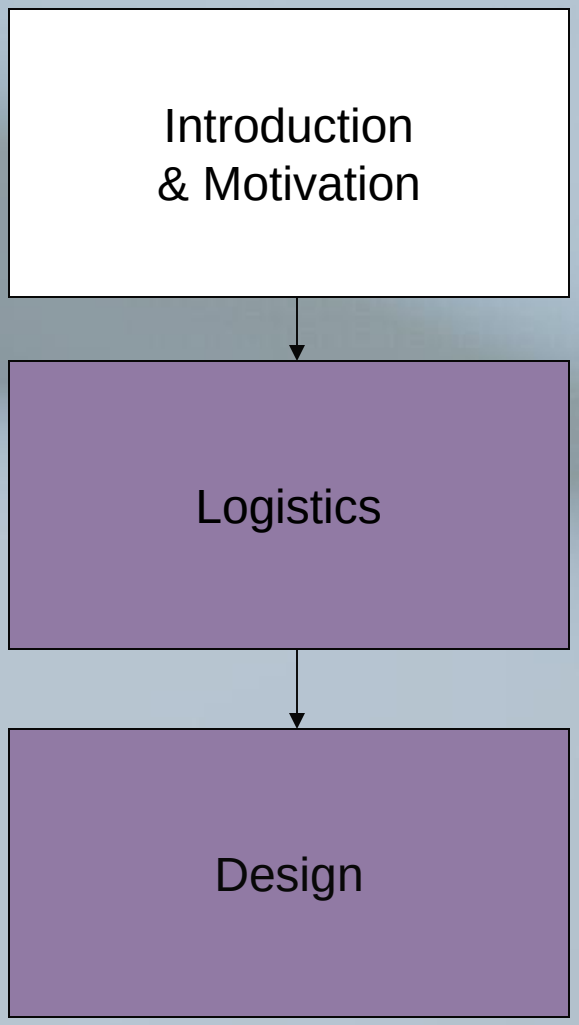


Digital Video Recorder

Eric Bowden, Matt Ricks, Irene Thompson

Presentation Roadmap



What is a DVR?

Why make a DVR?

Implementation Overview

Specifications (Baseline and Extensions)

What is a Digital Video Recorder?

A video capture unit for live television signals

Baseline Functionality

Pause Live TV

- Never miss part of a show again due to interruptions

Rewind Live TV

- For up to 15 minutes of the broadcast

Fast Forward

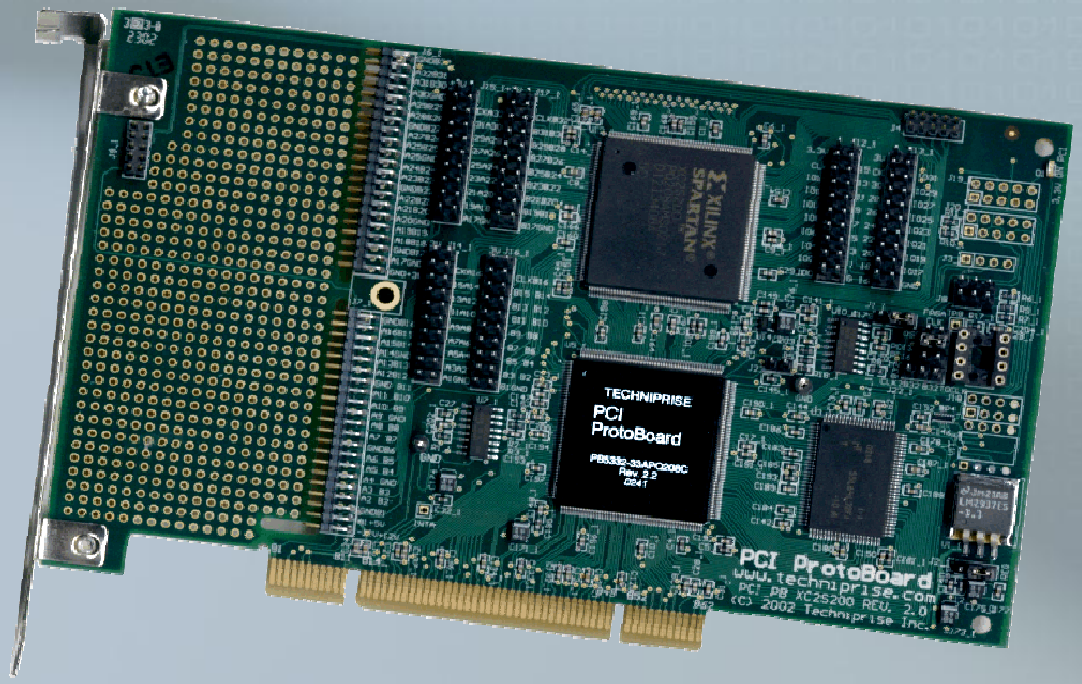
- Allows the user to skip through selected portions of the buffered broadcast

Motivation

- Currently available DVRs charge a monthly subscription fee
- Non-trivially combines both hardware and software components
- Allows us to explore analog signal processing, analog-to-digital conversion, and digital storage

DVR Implementation

We will develop a PCI 2.2 compliant add-on card for use in consumer PCs running Windows XP.



Functional Specification

The baseline device must:

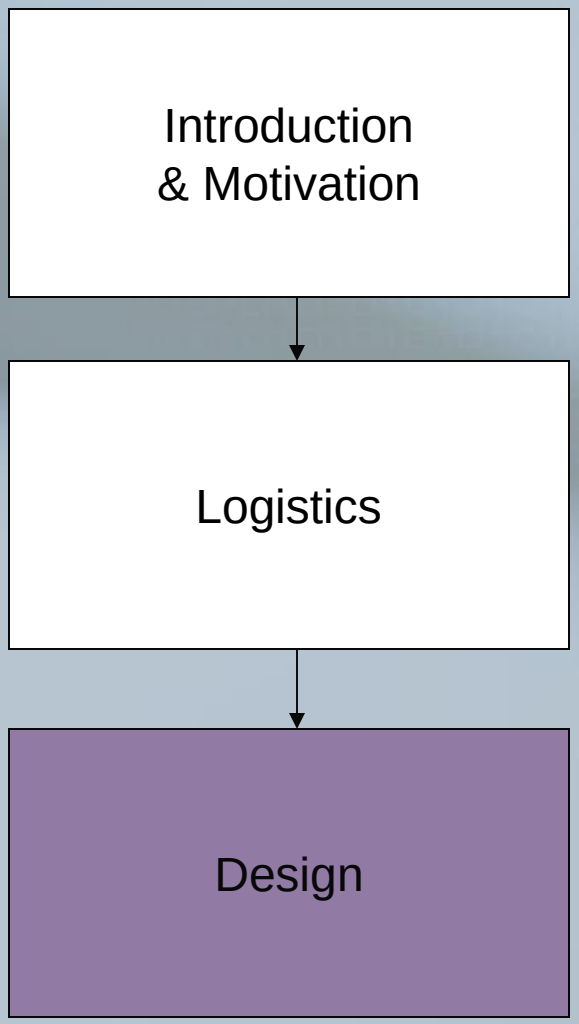
- Convert an analog, NTSC television signal (video/audio) into a digital signal
- Manipulate live television using pause, play, rewind and fast-forward
- Store, at minimum, 15 minutes of live television for recall using the aforementioned techniques
- The recalled audio/video data must be converted back into an NTSC composite signal

Extended Specification

Additional functionality, time permitting

- Commercial detection and eradication for recorded broadcasts
- Increased storage capacity
- User interface via remote control
- Embedded menus for navigation via a television
- Schedule recording a program for later viewing

Presentation Roadmap



Project Task Leads

Project Schedule

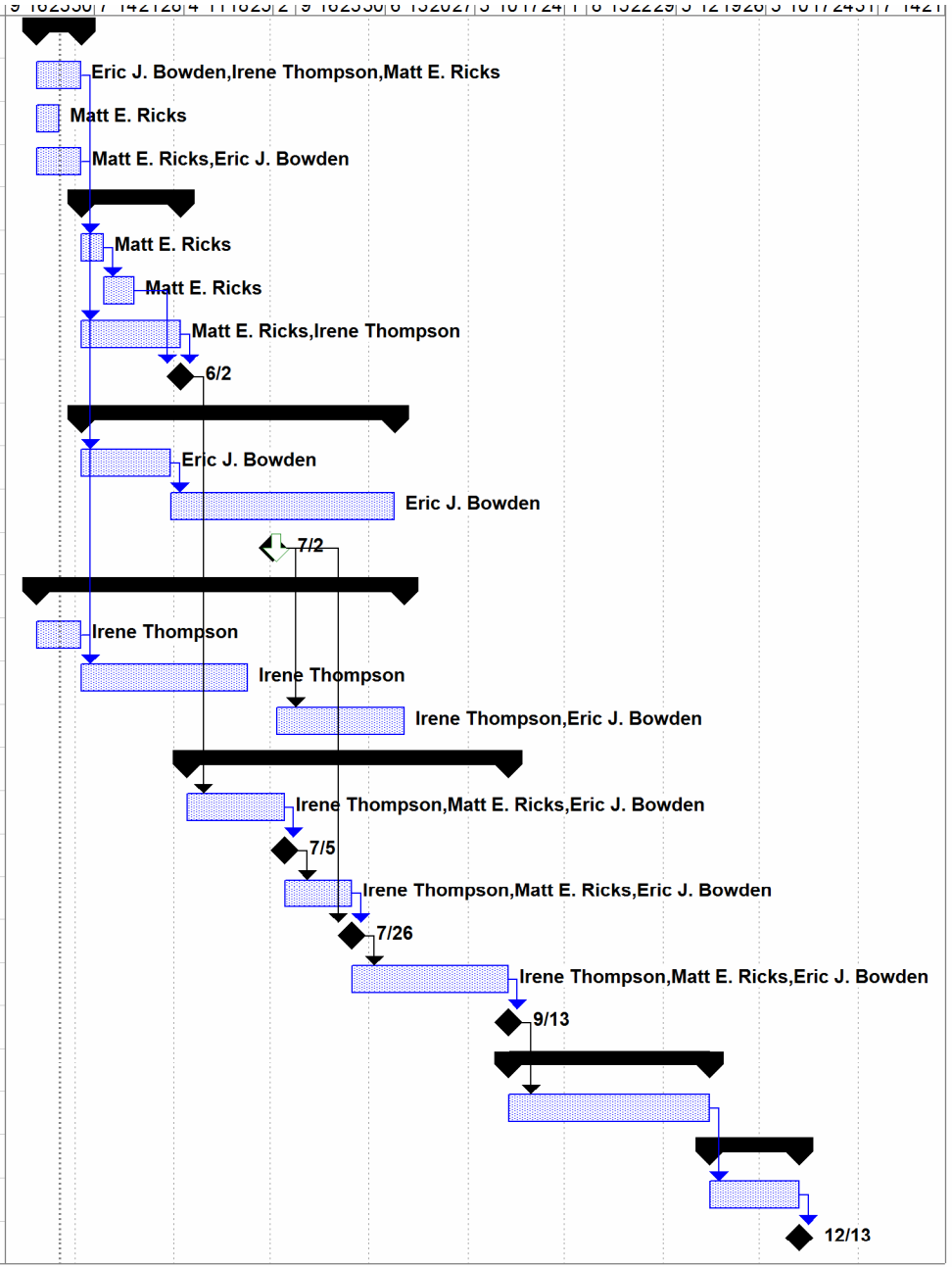
Bill of Materials

Risks

Tasking

- Matt Ricks: analog-to-digital encoding and decoding, hardware development
- Irene Thompson: signal subsystem processing and software development
- Eric Bowden: analog and digital integration and the software development for the PCI interface

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

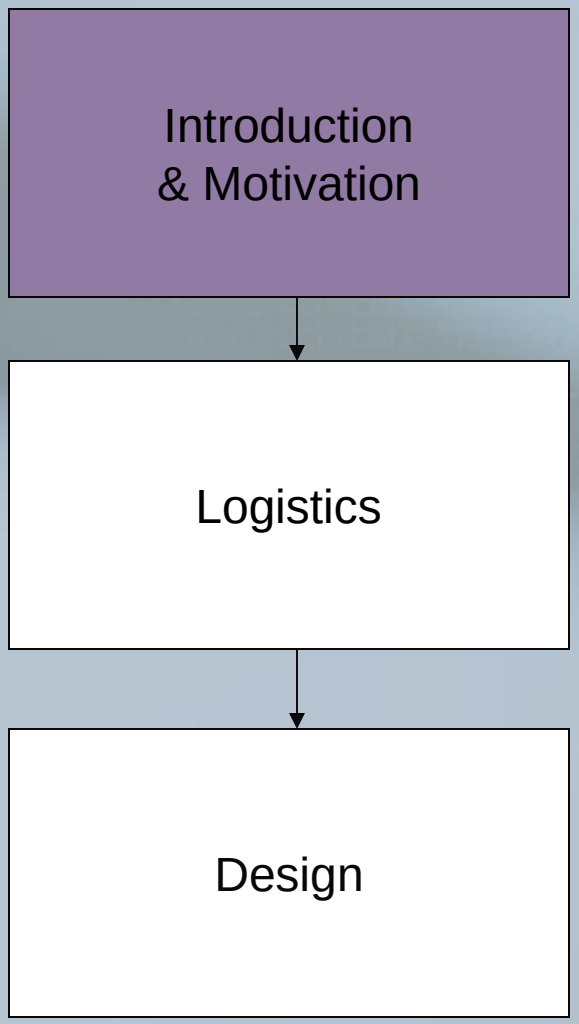
Bill of Materials

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

Risks

- Speed, not being able to store live video faster than we receive it.
- Kernel Module, difficulty factor unknown
- Needing to manufacture multiple PCI boards (unsoldering pins)
- Traces not lining up nicely = Multi-layer board
- Commercial Detection
- Compression backlog (Frame dropping)
  - Different chip or Different compression codec

Presentation Roadmap

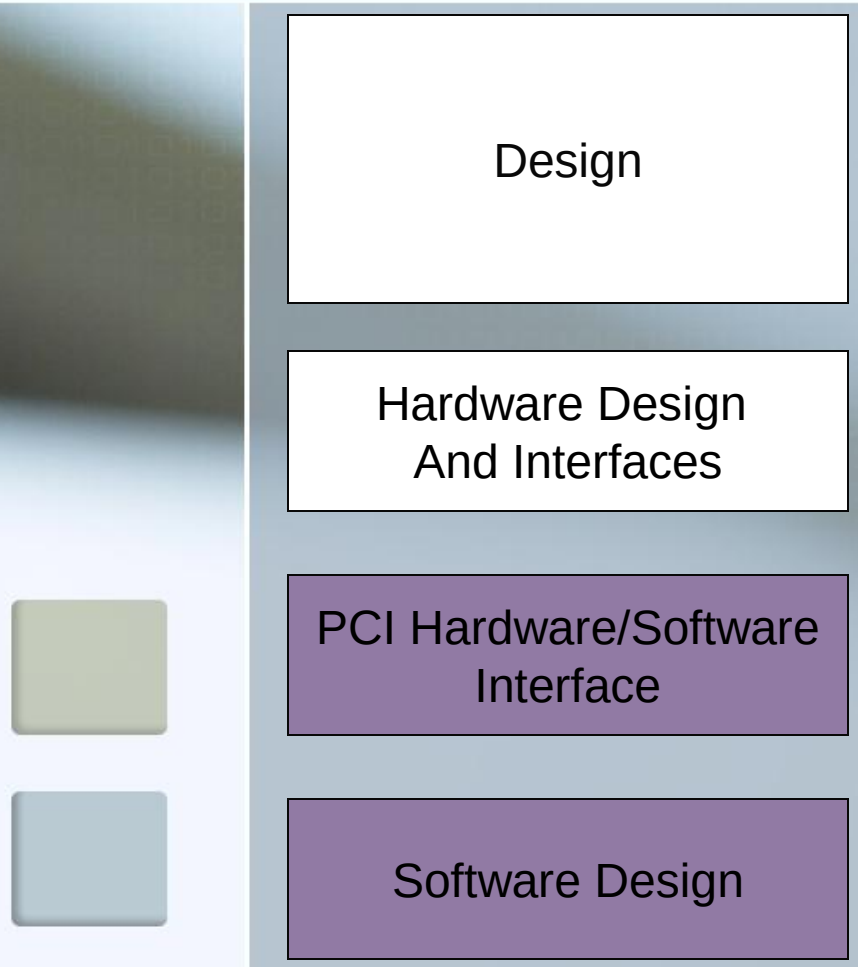


Hardware Design and Interfacing

PCI Bus Considerations

Software Design and Interfacing

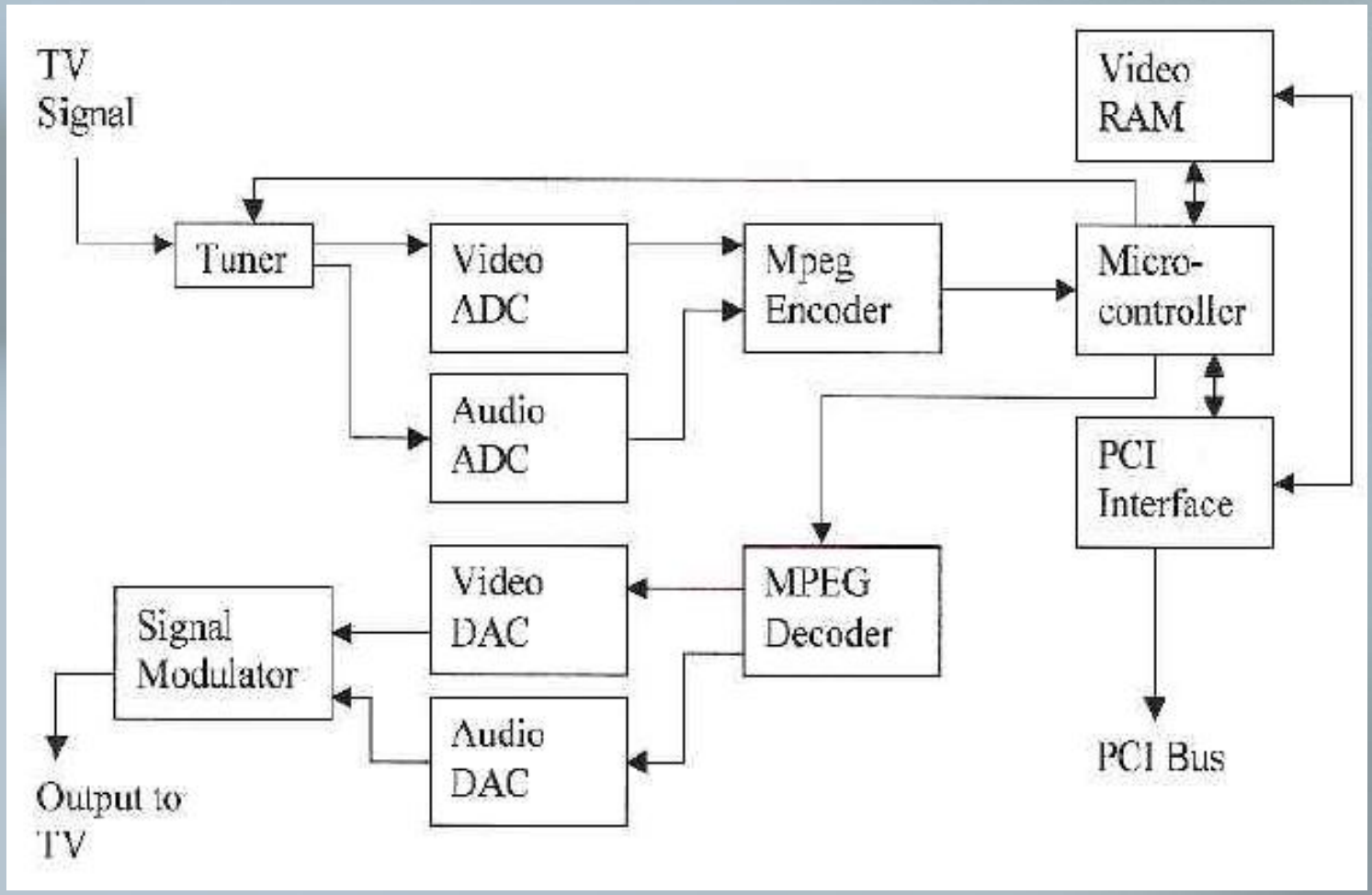
Project Design



Hardware Design

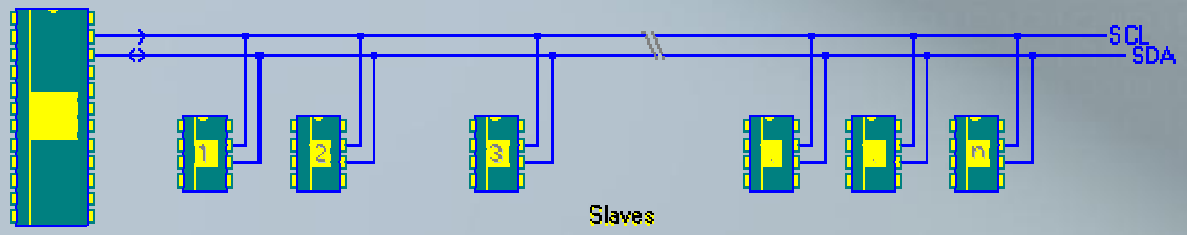
- Hardware block diagram
- I2C Bus
- NTSC Video notes
- MPEG Compression
- Microcontroller
- Remixing

DVR PCI Card Block Diagram



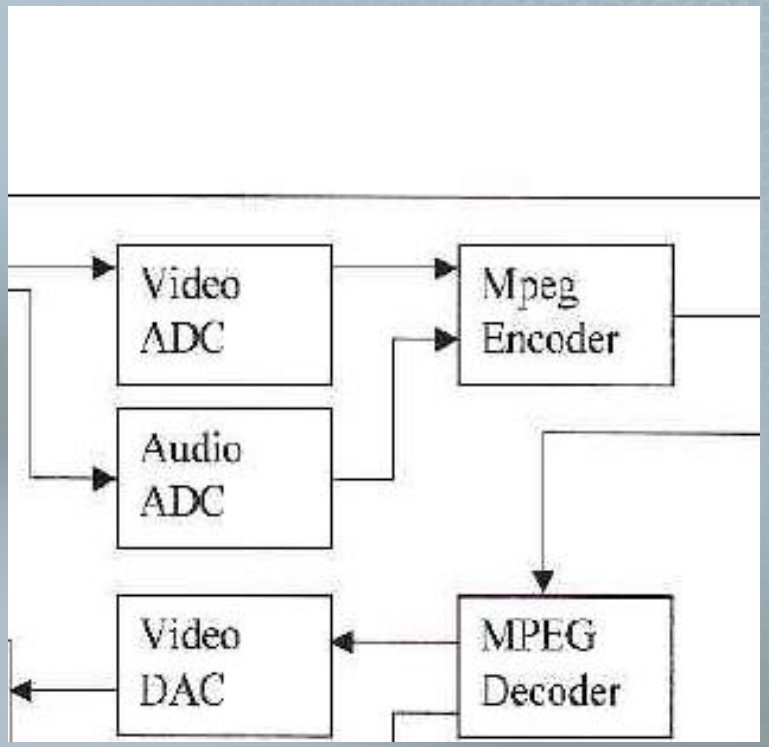
I²C Bus

- I2C Bus = Inter-IC bus
- 2 wire bi-directional bus
- Originally designed for TV
- Supported by most video IC's

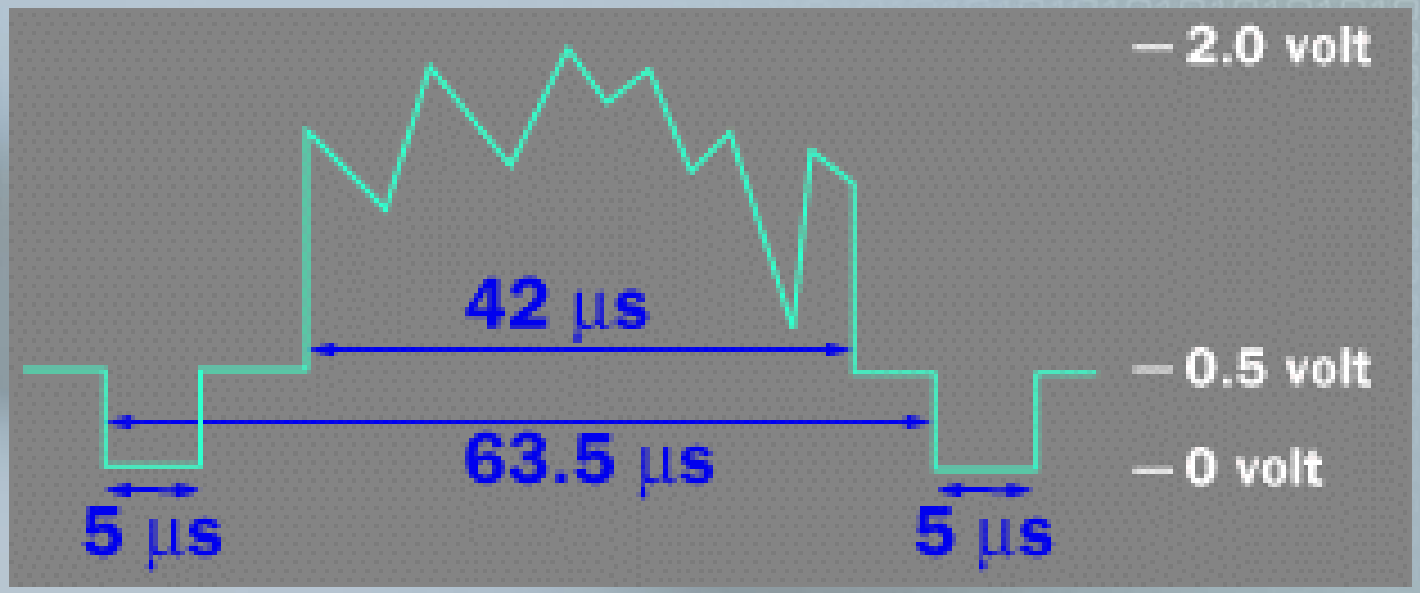


Video Signal Decoding

- Tuner (what channel?) 6MHz apart
- Audio/Video filter
- Audio and Video ADC
- Chrominance and Luminance
- Horizontal and Vertical Sync
- Noise Filter
- Video Amplifier



Analog Video Signal

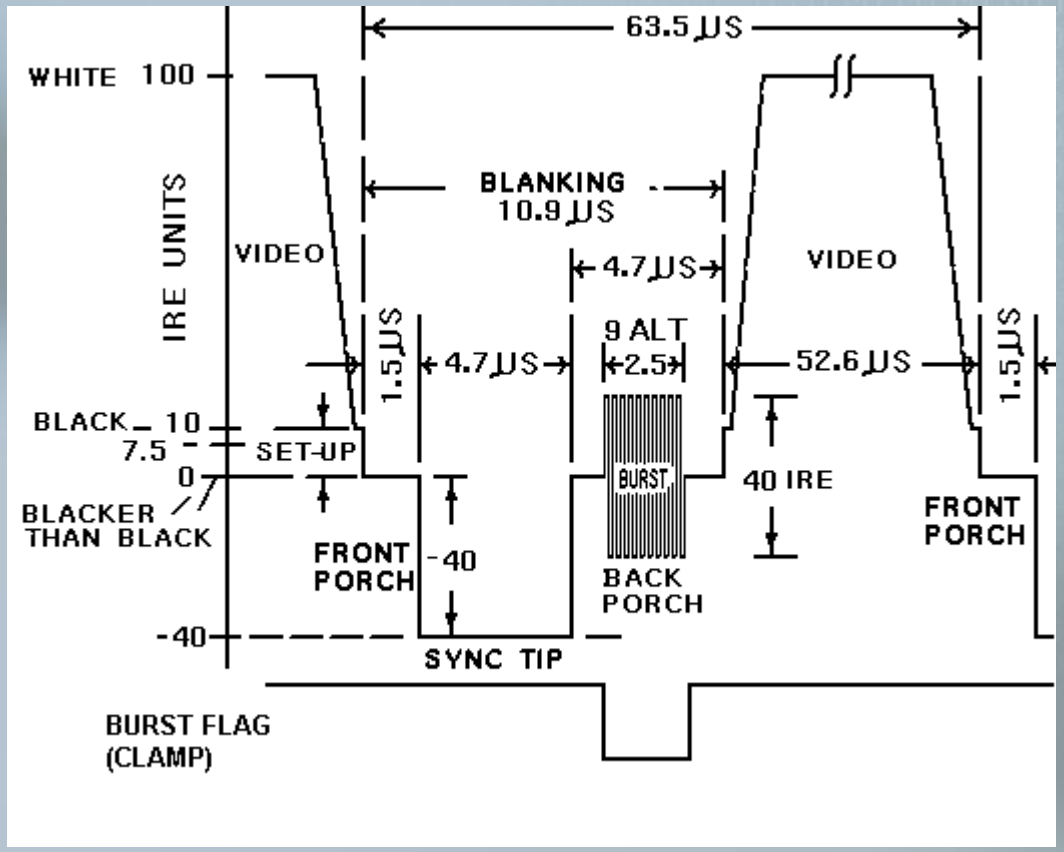


30 Frames per second (60 Hz)

525 lines per frame (15 kHz)

Pixel Frequency = 12.5 MHz

Video Signal Detail Diagram



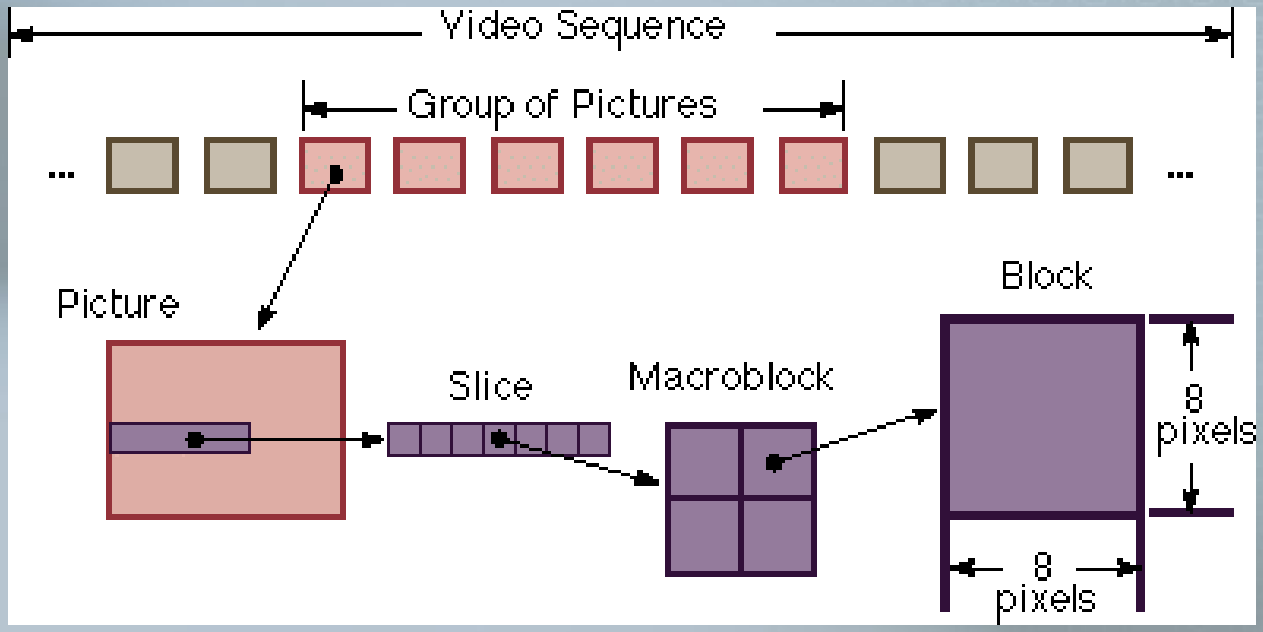
Video Compression

- JPEG
- H.261
- MPEG

What is MPEG

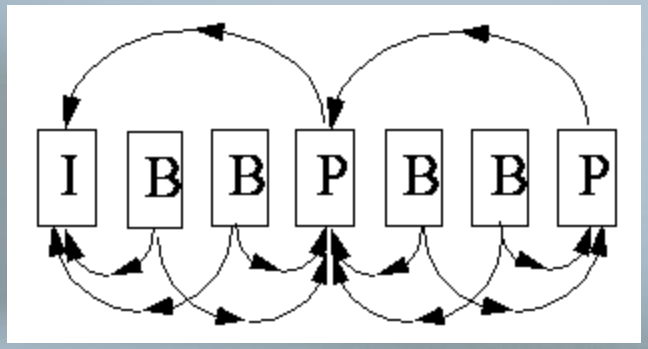
- Moving Pictures Experts Group
- Supports JPEG and H.261 through downward compatibility
- Supports higher Chrominance resolution and pixel resolution (720x480 is standard used for TV signals)
- Supports interlaced and noninterlaced modes
- Uses Bidirectional prediction in “Group Of Pictures” to encode difference frames.

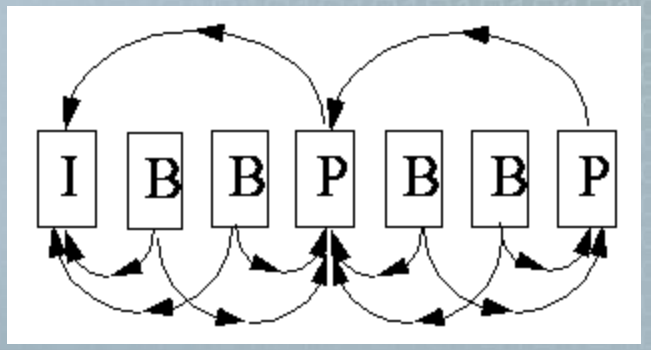
MPEG Bitstream



Source: http://www.doc.ic.ac.uk/~nd/surprise_96/journal/vol4/sab/report.html

Bidirectional Coding

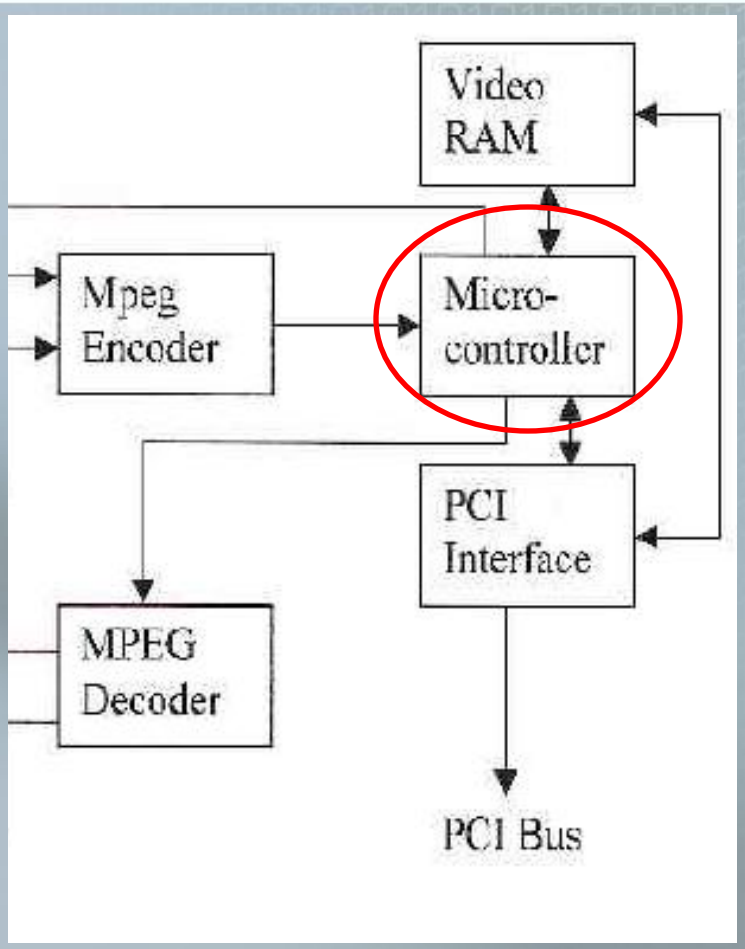




- I = Intra – Anchor picture
- P = Forward predicted
- B = Bidirectionally predicted

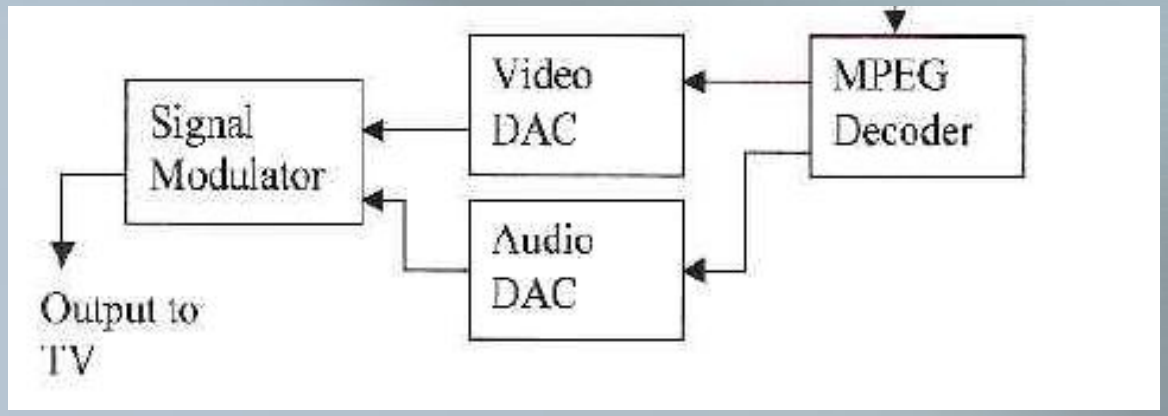
PCI Microcontroller

- There will be a small, simple microcontroller to manage input and output data buffering. This allows major processing to be offloaded to the host computer.
- Will be using the PIC16F870 μC from Microchip

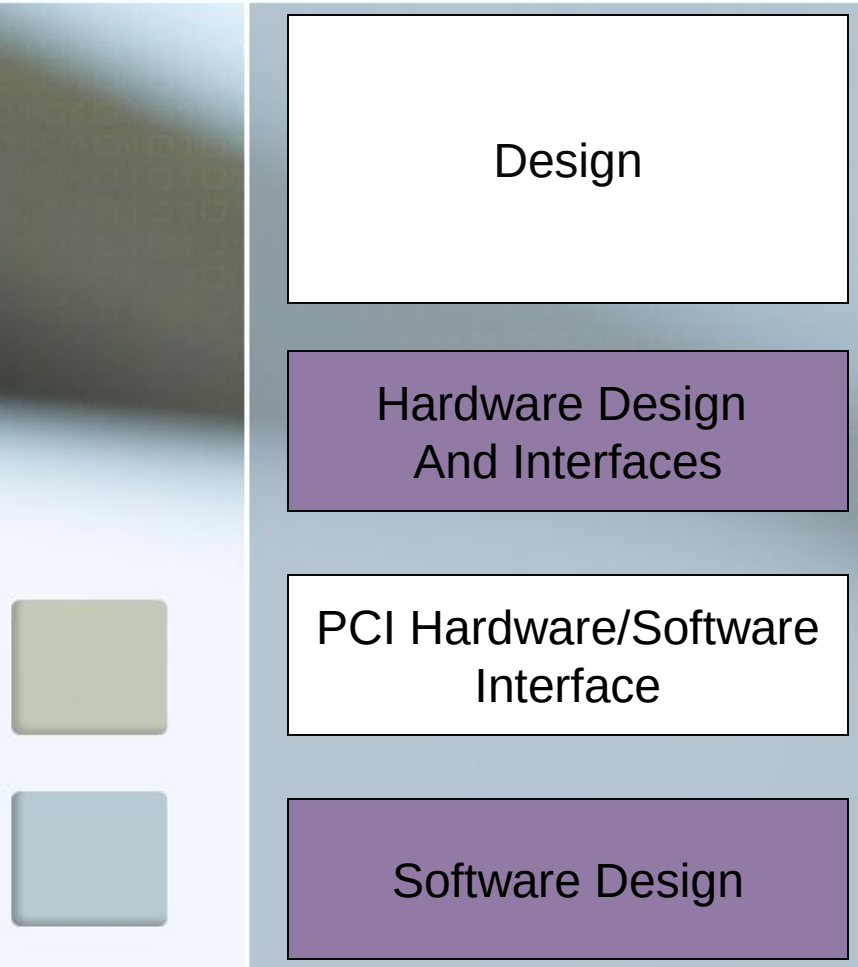


Data Re-encoding

- Data will be passed back to the card, encoded into an analog signal, mixed and then sent as a composite NTSC signal to the Television.



Project Design



PCI Hardware/Software Interface

- PCI-Host Communication
- PCI Interface Chip

Digital Interface

Two major digital paths

1. Sending digitized A/V signal to the host program for storage and manipulation.
2. Retrieving A/V for decompression and display from the host program.

PCI-Host Communication

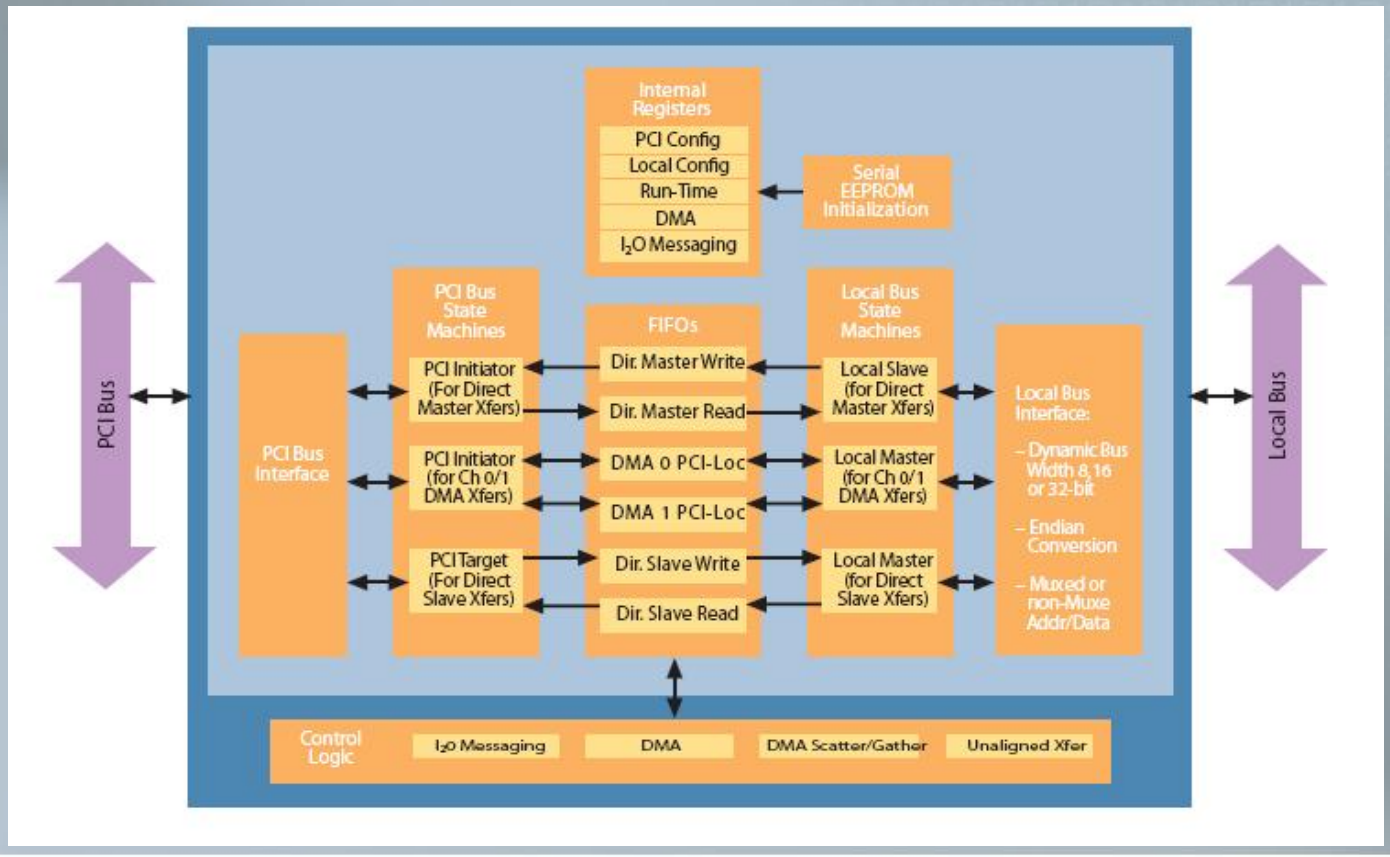
- The device will periodically (about once every frame) unload its digitized signal-buffer onto the PCI bus through standard interrupt methods.
- This data will be received on the host computer by a software application that stores the A/V data to disk.
- This will necessitate building a kernel module for Windows XP.

PCI Interface Chip

- PLX PCI 9054
  - PCI 2.2, 3.3V-Signalling Interface Chip
  - 33 MHz, 132 MB/s

Basic Operation: Card to Host

- The PLX I/O Accelerator chip serves as a target for the PCI Card microcontroller. It takes care of the PCI bus timing and ISR handling.



Basic Operation: Host to Card

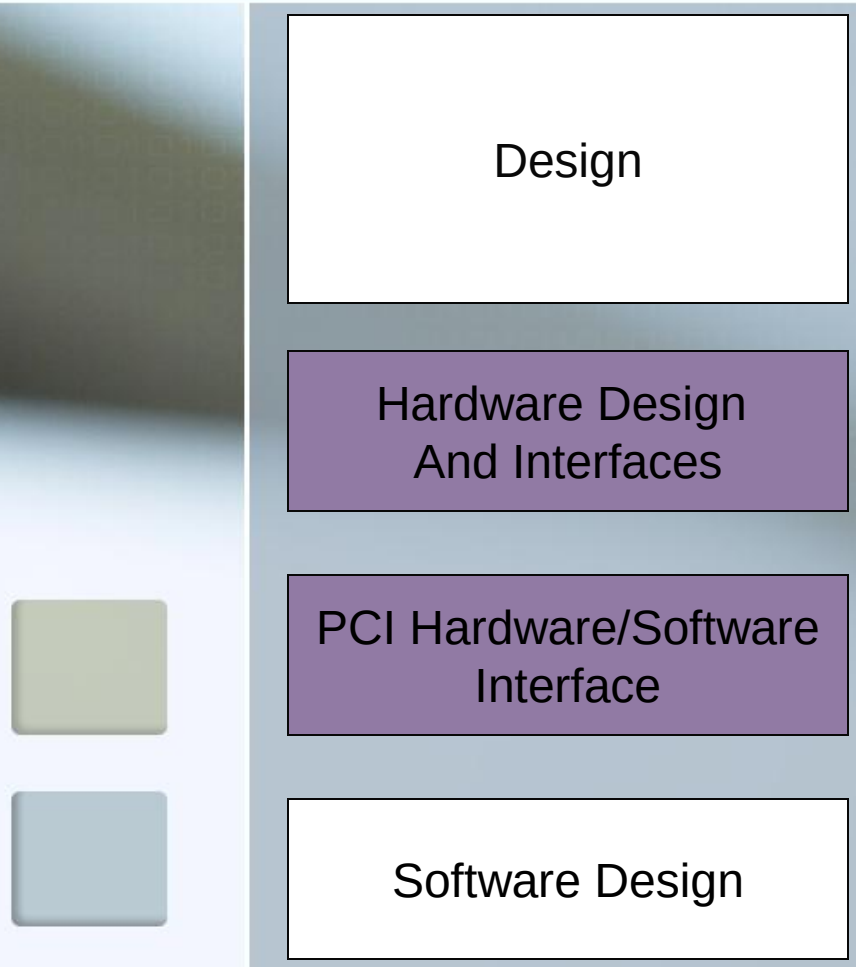
The host application will be in charge of which data is sent back to the PCI Card. Therefore, it will send I/O Request Packets (via the kernel module) to the device containing data to output.

The PLX chip has two DMA engines, allowing the host PC software to "write directly" to the memory on PCI add-on card. This data will then be retrieved by the microcontroller and sent to the television.

The Mediator (Device Driver)

- Most difficult part on the digital side.
- The driver is conceptually like a DLL. Uses DriverEntry and DriverObject to register functionality with the OS.
- Driver processes requests (IRP) sent from applications in user mode. Also registers the ISR with OS to manage incoming data from the PCI card.

Project Design



Software Design

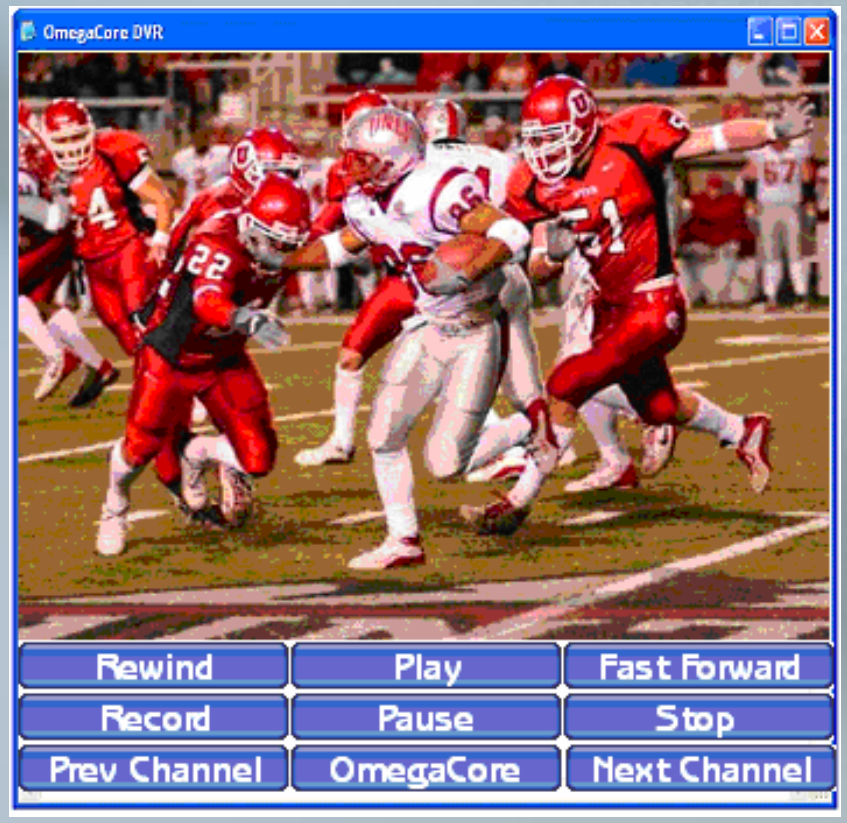
- Host application
- Graphical User Interface
- Microcontroller Software

Host Application

- The host software will act as an agent
- Its job is to constantly record A/V data to disk and send requested data back to the PCI card
- This software will provide the video control user experience
- Ideally there would be a GUI to allow for advanced broadcast manipulation

Host PC API and UI

- Data Flow Control
- User Interface



μC Software

- Receives Data from MPEG Encoder and stores it into VRAM
- Selects address in VRAM of the Data to be sent on the PCI Bus
- Receives Data from the PCI Bus via VRAM and sends it to the MPEG Decoder
- Changes the frequency of the tuner if so required

The End

http://www.cs.utah.edu/~ebowden/omegacore/